

# VISVESVARAYA TECHNOLOGICAL UNIVERSITY

JNANASANGAMA, BELAGAVI – 590018



## Mobile Application Development Mini Project Report on

### CAFETERIA APPLICATION

*Submitted in partial fulfilment for the award of degree of*

## **Bachelor of Engineering In Computer Science and Engineering.**

Submitted by

**Param S - 1BG19CS070**

**Prajwal R - 1BG19CS071**

**Rohan B - 1BG19CS084**

Under the Guidance of

**Prof. Kavyashree K R**

Department of CSE, BNMIT



*Vidyaya Amrutam Ashnute*

## *B.N.M. Institute of Technology*

**An Autonomous Institution under VTU**

Approved by AICTE, Accredited as grade A Institution by NAAC. All eligible branches – CSE, ECE, EEE, ISE & Mech. Engg. are Accredited by NBA for academic years 2018-19 to 2021-22 & valid up to 30.06.2022

URL: [www.bnmit.org](http://www.bnmit.org)

**Department of Computer Science and Engineering**

**2021 – 2022**

# *B.N.M. Institute of Technology*

An Autonomous Institute Under VTU

## DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



*Vidyaya Amrutam Ashnatho*

## CERTIFICATE

Certified that the Mini Project entitled Chat Application by **Mr. Param S USN 1BG19CS070, Mr. PRAJWAL R USN 1BG19CS071, Mr. ROHAN B USN 1BG19CS084**, bona fide students of VI Semester B.E., **B.N.M Institute of Technology** in partial fulfillment for the Bachelor of Engineering in **COMPUTER SCIENCE AND ENGINEERING** of the **Visvesvaraya Technological University, Belagavi** during the year 2021-22. It is certified that all corrections / suggestions indicated for Internal Assessment have been incorporated in the report. The Project report has been approved as it satisfies the academic requirements in respect of Mobile Application Development with Mini Project Laboratory prescribed for the said Degree.

**Prof. Kavyashree K R**  
**Assistant Professor**  
**Department of CSE**  
**BNMIT, Bengaluru**

**Dr. Chayadevi M L**  
**Professor and HOD**  
**Department of CSE**  
**BNMIT, Bengaluru**

Name & Signature

1. Examiner 1:
2. Examiner 2:

# **ABSTRACT**

Smartphones have revolutionized the human lifestyle of today. This is because Mobile Applications are designed specifically to help smartphone users by connecting to the Internet. Most of us always want updates fast, so Mobile App can help users to provide information quickly. The technological advancement of this era has made it easier to work and save time and effort. The Cafeteria Service System is design especially for Mobile Application for students to see the food menu presented in the college cafeteria and order food via smartphones. Students do not have to worry about finding or taking a long time to choose what food they want to eat. With the availability of online food orders, students no longer need to go out for food and wait in queues. In conclusion, this Mobile App can save time and effort at the same time. The outcome of this project is to help users to use this Mobile App wherever they are and order food. Once the order is ready the user will get an email with the Total amount. In conclusion, with the ease of technology available, it has greatly facilitated and utilized our work.

# ACKNOWLEDGEMENT

The success and outcome of this project required a lot of guidance and assistance from many people, and I am extremely privileged to have got this all along the completion of my project.

I would like to thank **Shri. Narayan Rao R Maanay**, Secretary, BNMIT, Bengaluru for providing the excellent environment and infrastructure in the college.

I would like to sincerely thank **Prof. T J Rama Murthy**, Director, BNMIT, Bengaluru for having extended his constant support and encouragement during this project.

I would like to sincerely thank **Dr. S Y Kulkarni**, Additional Director, BNMIT, Bengaluru for having extended his constant support and encouragement during this project

I would like to express my gratitude to **Prof. Eishwar N Maanay**, Dean, BNMIT, Bengaluru for his relentless support and encouragement.

I would like to thank **Dr. Krishnamurthy G N**, Principal, BNMIT, Bengaluru for his constant encouragement

I would like to thank, **Dr. Chayadevi M L**, Professor & Head of the Department of Computer Science and Engineering for the encouragement and motivation she provides.

I would also like to thank **Mrs. Kavyashree K. R**, Assistant Professor, Department of Computer Science and Engineering for providing me with her valuable insight and guidance wherever required throughout the course of the project and its successful completion.

**PARAM S (1BG19CS070)**

**PRAJWAL R (1BG19CS071)**

**ROHAN B (1BG19CS084)**

# TABLE OF CONTENTS

<b>CONTENTS</b>	<b>Page No</b>
<b>ABSTRACT</b>	<b>I</b>
<b>ACKNOWLEDGMENT</b>	<b>II</b>
<b>1. INTRODUCTION</b> 1.1. Overview 1.2. Problem Statement 1.3. Mobile Application Need and Importance 1.4. Android Studio 1.5. Firebase	 1 1 2 3 5
<b>2. LITERARY SURVEY</b> 2.1. Related Work	 6
<b>3. SYSTEM REQUIREMENTS</b> 3.1. Software Requirements 3.2. Hardware Requirements	 8 8
<b>4. SYSTEM DESIGN</b> 4.1. Connecting to the firebase 4.2. XML Code	 10 10
<b>5. IMPLEMENTATION</b> 5.1. Description 5.2. Java Code	 34 35
<b>6. RESULT</b>	50
<b>7. CONCLUSION &amp; FUTURE SCOPE</b>	56
<b>BIBLIOGRAPHY</b>	57

## **List of Figures**

<b>Figure No.</b>	<b>Figure Name</b>	<b>Page No.</b>
Fig 1.1	Android Studio main window	4
Fig 6.1	SignIn Page	50
Fig 6.2	SignUp Page	51
Fig 6.3	Landing Page	52
Fig 6.4	Coffee Order Page	53
Fig 6.5	Order Summary Page	54
Fig 6.6	Firebase Realtime Database	55

## Chapter-1

# INTRODUCTION

### 1.1 Overview

With the advent of the digital within the cafeteria industry, every cafeteria has picked up the pace to match the changing scenarios. Be it augmenting the interface for consumers, relationship with facilitators for delivery and procurement, or better interacting software with third party APIs, every step taken added convenience and embraced technology.

When entering a business in the online food delivery category, the best way forward is to understand the trends that mobilize the industry. Awareness into how an app adds to the popularity convinces owners to develop a mobile app for their cafeterias.

Some noteworthy facts that add to the argument are:

- Adding digital channels for ordering and subsequently upgrading to delivery, the share prices of Dominos have risen by 40% since 2012. Today, it is reportedly getting 60% of all its orders through these channels. It also claims to have seen a significant rise in same-store sales within a quarter, due to addition of channels like online ordering and introducing a dedicated app.
- Starbucks has been able to add specific personalization features for its customers through the mobile app. Extending exclusive loyalty programs, deals and offers, it has exemplified user experience, and has been able to capitalize on the investments made on technical feature enhancements.

The transformation within the food value-chain has gained momentum with technologies like big data and artificial intelligence coming into main sight.

A customized food or cafeteria app development will exploit technology even more exhaustively.

Cloud, blockchain, AI, big data, and 5G services will all contribute to better customer understanding, experience, service, and safety in the online food delivery.

## **1.2 Problem Statement**

The main objective of the Cafeteria Application is to make a computerized system, an efficient and user-friendly desktop application for the fast transaction and services to the students and staffs of BNMIT. It manages all the information about Cafeteria, User Profile and current order summary. The project is totally built at user level and thus all users are guaranteed the access. The purpose of the project is to build an application program to reduce the manual work for managing the cafeteria.

## **1.3 Mobile Application Development Need & Importance**

Every business mind understands the language of RoI, the return on investment made. Developing a cafeteria app that helps fulfil a major purpose of seamless access demands a decent investment, and the benefits drawn from this development can contribute to growth of the cafeteria business.

Creating a cafeteria app can prove beneficial to both, a new cafeteria business owner who is yet to launch into the market, as well as those who are already running their cafeteria and are looking to add to their marketing and expansion efforts.

Here we discuss for you some most prominent benefits that promote the idea of developing an adept app for your cafeteria.

### **1. Brand Value Addition**

A mobile application that allows consumers to access the cafeteria from any place, order their favourite food without being restricted to a device is a definite value addition to the brand. Convenient connectivity goes a long way in creating impressions in the consumer minds.

### **2. Better Customer Experience**

A well-designed application that transitions smoothly from locating cafeterias, making food item selections, ordering dishes, adds appeal and builds a refined experience for the end user.

All features integrated via well written APIs that help the application run uninterrupted for all operations enhances the UX.



## 1.4 Android Studio

In recent times, Android became the world's most popular operating system for various reasons. As an Android programmer, I want to share what the Android Studio is? Android Studio is an IDE for Google Android Development launched on 16th May 2013, during Google's I/O 2013 event. Android Studio contains all the Android tools to design, test, debug, and profile your application. The Android Studio uses Gradle to manage your project, a Build Automation Tool.

For developing your first app, you need to download Android Studio for your preferred platform (Windows®, Mac OS X, or Linux) from the Android developers' site. Android Studio can develop and test your application on either a real device or an emulator.

Android Studio can be installed on Windows operating systems, OSX and Linux and is recommended by Google itself that the hardware must have at least 4 GB of memory and 1GB of free hard disk space, but we recommend that you have more memory because it was noted that Android Studio is still a little slow. You must have Java installed on the machine via the JDK (Java Development Kit), not the JRE, as it is usually installed, once to develop on Android is necessary for all Java development classes to be present on the machine.

**Android Studio has many exciting features that can help you to develop your Android application like:**

- Powerful code editor with smart editing and code re-factoring.
- Emulator to show your code output in various resolutions, including Nexus 4, Nexus 7, Nexus 10, and many other android phones.
- Gradle based build support.
- Maven Support.
- Template-based wizards.
- Dracula Theme Environment to enjoy your coding experience.

You can experience all the awesome features by using Android Studio in-hand.

## The user interface

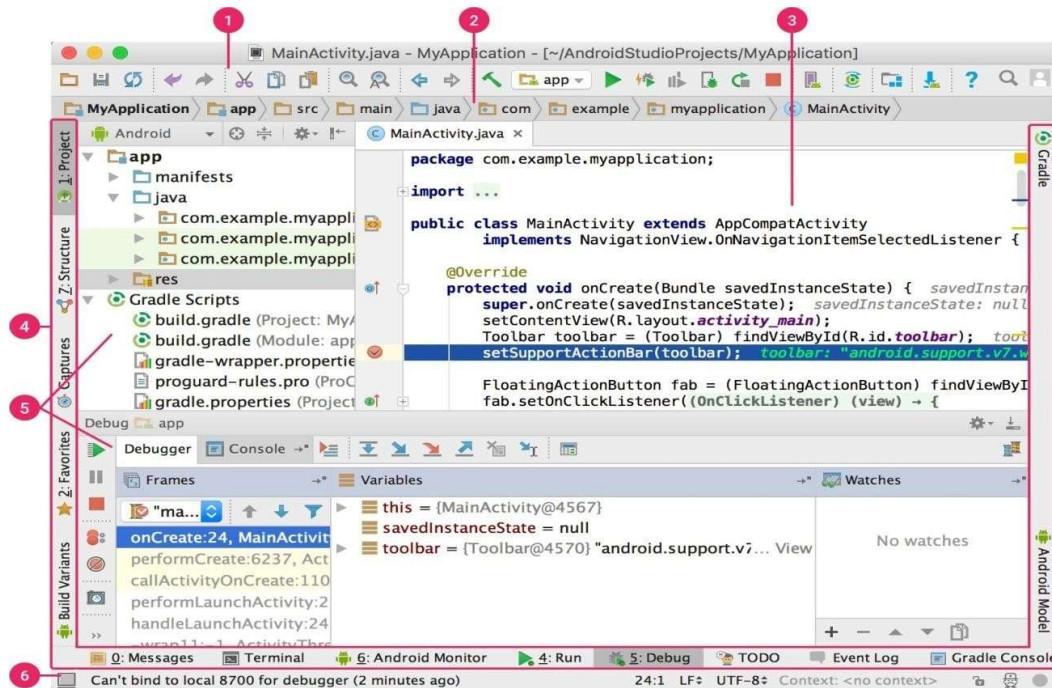


Fig 1.1 The Android Studio main window.

- **The toolbar** lets you carry out a wide range of actions, including running your app and launching Android tools.
- **The navigation bar** helps you navigate through your project and open files for editing. It provides a more compact view of the structure visible in the Project window.
- **The editor window** is where you create and modify code. Depending on the current file type, the editor can change.
- **The tool window bar** runs around the outside of the IDE window and contains the buttons that allow you to expand or collapse individual tool windows.
- **The tool windows** give you access to specific tasks like project management, search, version control, and more.
- **The status bar** displays the status of your project and the IDE itself, as well as any warnings or messages.

## 1.5 Firebase

Google Firebase is a Google-backed application development software. Firebase provides tools for tracking analytics, reporting, and fixing app crashes, creating marketing and product experiment.

Firebase offers several services, including:

- Analytics – Google Analytics for Firebase offers free, unlimited reporting on as many as 500 separate events. Analytics presents data about user behavior in iOS and Android apps, enabling better decision-making about improving performance and app marketing.
- Authentication – Firebase Authentication makes it easy for developers to build secure authentication systems and enhances the sign-in and onboarding experience for users. This feature offers a complete identity solution, supporting email and password accounts, phone auth, as well as Google, Facebook, GitHub, Twitter login and more.
- Cloud messaging – Firebase Cloud Messaging (FCM) is a cross-platform messaging tool that lets companies reliably receive and deliver messages on iOS, Android and the web at no cost.
- Realtime database – the Firebase Realtime Database is a cloud-hosted NoSQL database that enables data to be stored and synced between users in real time. The data is synced across all clients in real time and is still available when an app goes offline.
- Crashlytics – Firebase Crashlytics is a real-time crash reporter that helps developers track, prioritize and fix stability issues that reduce the quality of their apps. With crashlytics, developers spend less time organizing and troubleshooting crashes and more time building features for their apps.
- Performance – Firebase Performance Monitoring service gives developers insight into the performance characteristics of their iOS and Android apps to help them determine where and when the performance of their apps can be improved.

## Chapter-2

### LITERATURE SURVEY

#### 2.1 Related Work

- **Types of mobile apps:**

There are three main types of mobile apps including native apps, web-based mobile apps and hybrid apps. Hybrid mobile apps combine elements of native and web-based apps. Native apps include Android, Windows Phone, and iOS. Hybrid apps are designed for platforms including Xamarin, Angular Mobile Sencha Touch, React Native, Ionic and others. On the other hand, web-based apps are responsive versions of websites designed to work on any mobile device.

- **Native Mobile Apps:**

Native apps are developed for a certain mobile device operating system like Windows Phone or Android. Therefore, they are native for a certain device or platform. Apps built for Android, Windows Phone, Blackberry, Symbian cannot be used on any other platform except on their own. Therefore, a mobile app designed for Android can only be used on an Android device. Main advantages of native apps are good user experience and high performance. In addition, an access to broad range of APIs puts no limits on app usage. Native mobile apps are accessible from app stores of their kind and have that very clear tendency of reaching target customers. Some disadvantages of native mobile apps include higher costs in comparison to other types of mobile apps. Creating a native mobile app duplicates costs since maintenance and separate supports for different apps are required that result in greater product price.

- **Hybrid Mobile Apps:**

Hybrid mobile apps are specifically built using different multi-platform web technologies like JavaScript and HTML5. Hybrid apps are website applications created in a native wrapper that means they use elements of both native and web-based apps. Hybrid apps also possess common cons and pros of both web mobile and native mobile applications. Hybrid multi-platform mobile apps are relatively easy to develop which is a clear advantage. Since code base for hybrid apps ensures that low-cost maintenance alongside smooth updates. On the other hand, hybrid application lack in speed, performance and overall optimization compared to native mobile apps. There are also specific design issues thanks to apps inability to look in the same way on different platforms.

- **Web-Based Apps:**

Web-based applications behave in very similar fashion to those native mobile apps. Web apps use a certain browser to run and they are commonly written in CSS, JavaScript or HTML5. Web apps redirect users to URL and further offer install options by creating a bookmark on their browser. The greatest advantage of web apps is that they require a minimum of device memory. Users can access web apps from any device that is connected to the Internet. All personal databases are saved on a certain server, so the use of web applications with poor internet connection commonly results in very bad user experience. Another drawback on web apps is access to not so many APIs, with exception of geolocation and several others. In addition, a performance of web-based apps is inextricably linked to network connection and browser work. Therefore, only around fourteen percent of time users spend on web-based apps as only some of device APIs can be used like geolocation.

- **Categories of Apps:**

Virtual reality (VR) is an experience taking place within a computer-generated reality of immersive environments can be like or completely different from the real world. Applications of virtual reality can include entertainment (i.e., gaming) and educational purposes (i.e., medical, or military training). Other, distinct types of VR style technology include augmented reality and mixed reality. Currently standard virtual reality systems use either virtual reality headsets or multi-projected environments to generate realistic images, sounds and other sensations that simulate a user's physical presence in a virtual environment. A person using virtual reality equipment can look around the artificial world, move around in it, and interact with virtual features or items. The effect is commonly created by VR headsets consisting of a head-mounted display with a small screen in front of the eyes but can also be created through specially designed rooms with multiple large screens. Virtual reality typically incorporates auditory and video feedback but may also allow other types of sensory and force feedback through haptic technology.

## Chapter-3

# SYSTEM REQUIREMENTS

### 3.1 Software Requirements

Software requirements deal with defining software resource requirements and prerequisites that need to be installed on a computer to provide optimal functioning of an application.

The following are the software requirements for the application:

- Operating System: Windows 11
- Development Environment: Android Studio, Firebase
- API: Java Development Kit (JDK)
- Core Languages: Java, XML for Front End

### 3.1 Hardware Requirements

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware.

- CPU: Intel processor with support for Intel VT-x (Intel 64)
- Cores: Dual-Core (Quad-Core recommended)
- RAM: minimum 4GB (>4GB recommended)
- Graphics: Intel Integrated Graphics or AMD Equivalent
- Secondary Storage: 250GB hard disk space plus at least 1GB for Android SDK, Emulator System images, and caches.
- Display Resolution: 1366x768 (1920x1080 recommended)

## Chapter-4

### SYSTEM DESIGN

#### 4.1 App Design

The proposed system is to create a cafeteria application for the users to order and receive the same easily. Every new user is required to register on the application, after which they are taken to a menu from which they can order. The user can select a food item, the quantity and place the order. The order summary is recorded, stored in the firebase, and is sent to the user to his mail.

##### Features:

1. User profile
2. Cafeteria information/menu customization screens
3. Item selection
4. Cafeteria button/link

#### 4.2 Connecting to the Firebase

- Add Firebase using the Firebase Assistant present in Android Studio.
- The Firebase Assistant registers the app with a Firebase project and adds the necessary Firebase files, plugins, and dependencies to your Android project — all from within Android Studio.
- Open the Android project in Android Studio.
- Open the Firebase Assistant: Tools > Firebase.
- In the Assistant pane, Click Connect to Firebase to connect the Android project with Firebase.
- Sync the app to ensure that all dependencies have the necessary versions.

## 4.2.1 Adding Libraries

The following libraries are added to the build.gradle (app level) file and the project is synched:

```
implementation 'com.google.firebase:firebase-core:16.0.8'  
implementation 'com.google.firebase:firebase-auth:16.2.1'  
implementation 'com.google.android.gms:play-services-auth:16.0.1'  
implementation 'com.google.firebase:firebase-database:16.1.0'  
implementation 'com.firebaseui:firebase-ui:0.6.2'
```

## 4.3 XML Code

### 1. activity\_main.xml

```
<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout  
xmlns:android="http://schemas.android.com/apk/res/android"  
xmlns:app="http://schemas.android.com/apk/res-auto"  
xmlns:tools="http://schemas.android.com/tools"  
android:layout_width="match_parent"  
android:layout_height="match_parent"  
tools:context=".MainActivity">  
  
    <ImageView  
        android:id="@+id/imageView3"  
        android:layout_width="412dp"  
        android:layout_height="232dp"  
        app:layout_constraintBottom_toBottomOf="parent"  
        app:layout_constraintEnd_toEndOf="parent"  
        app:layout_constraintHorizontal_bias="1.0"  
        app:layout_constraintStart_toStartOf="parent"  
        app:layout_constraintTop_toTopOf="parent"  
        app:layout_constraintVertical_bias="0.0"  
        app:srcCompat="@drawable/_192988" />  
  
    <ImageButton  
        android:id="@+id/coffeeButton"  
        style="?attr/selectableItemBackgroundBorderless"
```



```
android:layout_width="100dp"
android:layout_height="100dp"
android:contentDescription="Coffee"
android:scaleType="fitCenter"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.086"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.57"
app:srcCompat="@drawable/_5973" />
```

<ImageButton

```
android:id="@+id/idliButton"
style="?attr/selectableItemBackgroundBorderless"
android:layout_width="100dp"
android:layout_height="100dp"
android:contentDescription="Idli Vada"
android:scaleType="fitCenter"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.498"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.57"
app:srcCompat="@drawable/idli_vada_logo" />
```

<ImageButton

```
android:id="@+id/dosaButton"
style="?attr/selectableItemBackgroundBorderless"
android:layout_width="100dp"
android:layout_height="100dp"
android:contentDescription="Dosa"
android:scaleType="fitCenter"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.909"
app:layout_constraintStart_toStartOf="parent"
```

```
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.57"
app:srcCompat="@drawable/dosa_logo" />
```

<ImageButton

```
android:id="@+id/samosaButton"
style="?attr/selectableItemBackgroundBorderless"
android:layout_width="100dp"
android:layout_height="100dp"
android:contentDescription="Samosa"
android:scaleType="fitCenter"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.498"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.77"
app:srcCompat="@drawable/samosa_logo" />
```

<ImageButton

```
android:id="@+id/breadButton"
style="?attr/selectableItemBackgroundBorderless"
android:layout_width="100dp"
android:layout_height="100dp"
android:contentDescription="Bread Pakoda"
android:scaleType="fitCenter"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.909"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.77"
app:srcCompat="@drawable/bread_pakoda_logo" />
```

<ImageButton

```
android:id="@+id/cakeButton"
style="?attr/selectableItemBackgroundBorderless"
android:layout_width="100dp"
```

```
android:layout_height="100dp"
android:contentDescription="Cake"
android:scaleType="fitCenter"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.498"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.974"
app:srcCompat="@drawable/cake_logo" />
```

```
<ImageButton
    android:id="@+id/paniButton"
    style="?attr/selectableItemBackgroundBorderless"
    android:layout_width="100dp"
    android:layout_height="100dp"
    android:contentDescription="Pani Puri"
    android:scaleType="fitCenter"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.086"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.77"
    app:srcCompat="@drawable/panipuri_logo" />
```

```
<TextView
    android:id="@+id/textView"
    android:layout_width="308dp"
    android:layout_height="33dp"
    android:text="Welcome to Cafeteria .... !! "
    android:textColor="#000000"
    android:textSize="24sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.495"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
```

```
        app:layout_constraintVertical_bias="0.332" />

<TextView
    android:id="@+id/textView2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="What would you like to Choose??"
    android:textColor="#000000"
    android:textSize="16sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.497"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.461" />

<EditText
    android:id="@+id/unameDisp"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:ems="10"
    android:inputType="text"
    android:minHeight="48dp"
    android:background="@null"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.497"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.4"
    tools:ignore="SpeakableTextPresentCheck" />

</androidx.constraintlayout.widget.ConstraintLayout>

theme_1000"/>
</LinearLayout>
```

## 2. activity\_sign\_up.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".SignUpActivity">
```

```
<TextView
    android:id="@+id/banner"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:fontFamily="cursive"
    android:text="Cafeteria"
    android:textAlignment="center"
    android:textColor="#FF0000"
    android:textSize="48sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.075" />
```

```
<EditText
    android:id="@+id/FNAME"
    android:layout_width="229dp"
    android:layout_height="59dp"
    android:hint="Username"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.294" />
```

```
<EditText
    android:id="@+id/USN"
    android:layout_width="229dp"
    android:layout_height="59dp"
    android:hint="USN"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.452" />
```

```
<EditText
    android:id="@+id/EMAIL"
    android:layout_width="229dp"
    android:layout_height="59dp"
    android:hint="Email"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.615" />
```

```
<Button
    android:id="@+id/SIGNUP"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="SIGN UP"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.918" />
```

```
<EditText
    android:id="@+id/PASSWD"
    android:layout_width="227dp"
    android:layout_height="53dp"
    android:layout_marginTop="24dp"
```

```
android:ems="10"
android:hint="Password"
android:inputType="textPassword"
app:layout_constraintBottom_toTopOf="@+id/SIGNUP"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.494"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/EMAIL"
app:layout_constraintVertical_bias="0.282" />
```

```
</androidx.constraintlayout.widget.ConstraintLayout>
```

### 3. activity\_sign\_in.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".SignInActivity">

<TextView
    android:id="@+id/Cafe"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:fontFamily="cursive"
    android:text="Cafeteria"
    android:textAlignment="center"
    android:textColor="#FF0000"
    android:textSize="48sp"
    android:textStyle="bold|italic"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.497"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.073" />
```

```
<EditText
    android:id="@+id/emailSi"
    android:layout_width="346dp"
    android:layout_height="64dp"
    android:hint="Email"
    android:textSize="24sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.492"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.322" />
```

```
<EditText
    android:id="@+id/passwdSi"
    android:layout_width="349dp"
    android:layout_height="56dp"
    android:ems="10"
    android:hint="Password"
    android:inputType="textPassword"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.499" />
```

```
<Button
    android:id="@+id/signin"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="36dp"
    android:text="SIGN IN"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.498"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/passwdSi"
    app:layout_constraintVertical_bias="0.0" />
```



```
<Button
    android:id="@+id/Register"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="36dp"
    android:text="SIGN UP"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.498"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/passwdSi"
    app:layout_constraintVertical_bias="0.731" />
```

```
<TextView
    android:id="@+id/OR"
    android:layout_width="70dp"
    android:layout_height="39dp"
    android:layout_marginTop="27dp"
    android:text="OR"
    android:textAlignment="center"
    android:textSize="24sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toTopOf="@+id/Register"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.498"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/signin"
    app:layout_constraintVertical_bias="0.0" />
```

```
</androidx.constraintlayout.widget.ConstraintLayout>
```

**4. activity\_coffee.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".CoffeeActivity">

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="256dp"
        android:layout_height="258dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.496"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.141"
        app:srcCompat="@drawable/_5973" />

    <TextView
        android:id="@+id/textView3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Coffee"
        android:textColor="#000000"
        android:textSize="24sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.499" />

    <Button
        android:id="@+id/coffeeAdd"
        android:layout_width="wrap_content"
```

```
android:layout_height="wrap_content"
android:text="+"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.268"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.635" />
```

<Button

```
android:id="@+id/coffeeSub"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="-"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.753"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.635" />
```

<EditText

```
android:id="@+id/coffeeNos"
android:layout_width="171dp"
android:layout_height="60dp"
android:ems="10"
android:hint="Nos."
android:inputType="textPersonName"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.758" />
```

<Button

```
android:id="@+id/coffeeOrder"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
```

```
        android:text="Order"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.498"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.91" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

### 5. activity\_order\_summary.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".OrderSummaryActivity">

    <EditText
        android:id="@+id/breadOrderCost"
        android:layout_width="89dp"
        android:layout_height="48dp"
        android:ems="10"
        android:hint="0"
        android:inputType="textPersonName"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.922"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.736" />

    <EditText
        android:id="@+id/cakeOrderCost"
        android:layout_width="89dp"
        android:layout_height="45dp"
        android:ems="10"
        android:hint="0"
```

```
android:inputType="textPersonName"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.954"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.855" />
```

<EditText

```
android:id="@+id/paniOrderCost"
android:layout_width="85dp"
android:layout_height="48dp"
android:ems="10"
android:hint="0"
android:inputType="textPersonName"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.907"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.487" />
```

<EditText

```
android:id="@+id/dosaOrderCost"
android:layout_width="94dp"
android:layout_height="48dp"
android:ems="10"
android:hint="0"
android:inputType="textPersonName"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.933"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.367" />
```

<EditText

```
android:id="@+id/idliOrderCost"
```

```
android:layout_width="90dp"
android:layout_height="48dp"
android:ems="10"
android:hint="0"
android:inputType="textPersonName"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.922"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.251" />
```

<EditText

```
android:id="@+id/coffeeOrderCost"
android:layout_width="92dp"
android:layout_height="38dp"
android:ems="10"
android:hint="0"
android:inputType="textPersonName"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.927"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.152"
tools:ignore="TouchTargetSizeCheck" />
```

<EditText

```
android:id="@+id/samosaOrderCost"
android:layout_width="87dp"
android:layout_height="48dp"
android:ems="10"
android:hint="0"
android:inputType="textPersonName"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.913"
app:layout_constraintStart_toStartOf="parent"
```

```
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.605" />
```

```
<TextView
    android:id="@+id/OrderSummary"
    android:layout_width="182dp"
    android:layout_height="43dp"
    android:text="Order Summary"
    android:textColor="#000000"
    android:textSize="24sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.122"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.046" />
```

```
<EditText
    android:id="@+id/orderTotal"
    android:layout_width="132dp"
    android:layout_height="48dp"
    android:ems="10"
    android:hint="0"
    android:inputType="textPersonName"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.781"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.948" />
```

```
<EditText
    android:id="@+id/coffeeNosOrder"
    android:layout_width="92dp"
    android:layout_height="48dp"
    android:ems="10"
    android:hint="0"
    android:inputType="textPersonName"
```

```
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.589"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.141" />
```

<ImageView

```
android:id="@+id/imageView9"
android:layout_width="50dp"
android:layout_height="50dp"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.077"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.145"
app:srcCompat="@drawable/_5973" />
```

<ImageView

```
android:id="@+id/imageView11"
android:layout_width="50dp"
android:layout_height="50dp"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.077"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.372"
app:srcCompat="@drawable/dosa_logo" />
```

<TextView

```
android:id="@+id/textView13"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Idli Vada 20"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
```



```
app:layout_constraintHorizontal_bias="0.28"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.257" />
```

```
<TextView
    android:id="@+id/textView15"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Pani Puri 30"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.281"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.494" />
```

```
<TextView
    android:id="@+id/textView17"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Cake 20"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.261"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.851" />
```

```
<TextView
    android:id="@+id/textView19"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Total Cost"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.27"
    app:layout_constraintStart_toStartOf="parent"
```

```
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.949" />
```

```
<TextView
    android:id="@+id/textView11"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Coffee 10"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.269"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.164" />
```

```
<TextView
    android:id="@+id/textView12"
    android:layout_width="78dp"
    android:layout_height="39dp"
    android:text="Bread Pakoda 20"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.288"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.738" />
```

```
<EditText
    android:id="@+id/breadNosOrder"
    android:layout_width="89dp"
    android:layout_height="48dp"
    android:ems="10"
    android:hint="0"
    android:inputType="textPersonName"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.586"
    app:layout_constraintStart_toStartOf="parent"
```

```
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.738" />
```

```
<ImageView
    android:id="@+id/imageView15"
    android:layout_width="50dp"
    android:layout_height="50dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.077"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.863"
    app:srcCompat="@drawable/cake_logo" />
```

```
<TextView
    android:id="@+id/textView14"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Dosa 40"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.261"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.386" />
```

```
<ImageView
    android:id="@+id/imageView10"
    android:layout_width="50dp"
    android:layout_height="50dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.077"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.255"
    app:srcCompat="@drawable/idli_vada_logo" />
```

```
<EditText
    android:id="@+id/cakeNosOrder"
    android:layout_width="104dp"
    android:layout_height="48dp"
    android:ems="10"
    android:hint="0"
    android:inputType="textPersonName"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.599"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.856" />
```

```
<ImageView
    android:id="@+id/imageView12"
    android:layout_width="50dp"
    android:layout_height="50dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.077"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.493"
    app:srcCompat="@drawable/panipuri_logo" />
```

```
<ImageView
    android:id="@+id/imageView13"
    android:layout_width="50dp"
    android:layout_height="50dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.077"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.616"
    app:srcCompat="@drawable/samosa_logo" />
```

```
<EditText
    android:id="@+id/samosaNosOrder"
    android:layout_width="87dp"
    android:layout_height="48dp"
    android:ems="10"
    android:hint="0"
    android:inputType="textPersonName"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.58"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.606" />
```

```
<TextView
    android:id="@+id/textView16"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Samosa 20"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.28"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.613" />
```

```
<EditText
    android:id="@+id/paniNosOrder"
    android:layout_width="85dp"
    android:layout_height="48dp"
    android:ems="10"
    android:hint="0"
    android:inputType="textPersonName"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.576"
    app:layout_constraintStart_toStartOf="parent"
```

```
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.489" />
```

```
<EditText
    android:id="@+id/dosaNosOrder"
    android:layout_width="94dp"
    android:layout_height="48dp"
    android:ems="10"
    android:hint="0"
    android:inputType="textPersonName"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.593"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.369" />
```

```
<ImageView
    android:id="@+id/imageView14"
    android:layout_width="50dp"
    android:layout_height="50dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.077"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.743"
    app:srcCompat="@drawable/bread_pakoda_logo" />
```

```
<EditText
    android:id="@+id/idliNosOrder"
    android:layout_width="90dp"
    android:layout_height="48dp"
    android:ems="10"
    android:hint="0"
    android:inputType="textPersonName"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
```

```
app:layout_constraintHorizontal_bias="0.585"  
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toTopOf="parent"  
app:layout_constraintVertical_bias="0.251" />
```

```
<Button
```

```
    android:id="@+id/placeOrder"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Place Order"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="0.905"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent"  
    app:layout_constraintVertical_bias="0.046" />
```

```
</androidx.constraintlayout.widget.ConstraintLayout>
```

## Chapter 5

### IMPLEMENTATION

#### 5.1 Description

- **Void onCreate(Bundle)**

The entire lifetime of an activity happens between the first call to onCreate(Bundle) and onDestroy(). An activity will do all setup of "global" state in onCreate(), and release all remaining resources in onDestroy().

- **public static interface View.OnClickListener**

Interface definition for a callback to be invoked when a view is clicked.

- **public class Intent**

An Intent provides a facility for performing late runtime binding between the code in different applications. Its most significant use is in the launching of activities, where it can be thought of as the glue between activities.

- **public void onDataChange ( )**

The onDataChange() method is used to read a static snapshot of the contents at a given path, as they existed at the time of the event. This method is triggered once when the listener is attached and again every time the data, including children, changes. The event callback is passed a snapshot containing all data at that location, including child data. If there is no data, the snapshot will return false when you call exists() and null when you call getValue() on it.

- **public void updateChildren ( )**

Update specific fields, to simultaneously write to specific children of a node without overwriting other child nodes, use the updateChildren() method.

When calling updateChildren(), you can update lower-level child values by specifying a path for the key.

- **public Bundle ( )**

Constructs a new, empty Bundle. Android Bundle is used to pass data between activities. The values that are to be passed are mapped to String keys which are later used in the next activity to retrieve the values.

- **public void onComplete(@NonNull Task<AuthResult> task){}**

It is a listener which when the task is completed gets Triggered.



## 5.2 Java Code

### 1. SignUpActivity.java

```
package com.example.cafeteria;

import static android.content.ContentValues.TAG;

import androidx.appcompat.app.AppCompatActivity;
import androidx.annotation.NonNull;
import android.os.Bundle;
import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.util.Patterns;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;

import android.widget.TextView;
import android.widget.Toast;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.FirebaseApp;
import com.google.firebase.FirebaseOptions;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.firestore.DocumentReference;
import com.google.firebase.firestore.FirebaseFirestore;

import java.io.FileInputStream;
```

```
public class SignUpActivity extends AppCompatActivity implements View.OnClickListener {  
    private FirebaseAuth mAuth;  
    private TextView cafe;  
    private EditText Fname,Usn,Email,Password;  
  
    FirebaseFirestore database = FirebaseFirestore.getInstance();  
  
    private Button signup;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_sign_up);  
        mAuth = FirebaseAuth.getInstance();  
        cafe=(TextView) findViewById(R.id.banner);  
        cafe.setOnClickListener(this);  
  
        Fname = (EditText) findViewById(R.id.FNAME);  
        Usn = (EditText) findViewById(R.id.USN);  
        Email = (EditText) findViewById(R.id.EMAIL);  
        Password = (EditText) findViewById(R.id.PASSWD);  
        signup = findViewById(R.id.SIGNUP);  
        signup.setOnClickListener(this);  
    }  
  
    @Override  
    public void onClick(View v) {  
        switch(v.getId()){  
            case R.id.banner:  
                startActivity(new Intent(this,SignInActivity.class));  
                break;  
            case R.id.SIGNUP:  
                RegisterUser();  
        }  
    }  
  
    private void RegisterUser() {
```

```
String email = Email.getText().toString().trim();
String name = Fname.getText().toString().trim();
String usn = Usn.getText().toString().trim();
String passwd = Password.getText().toString().trim();
```

```
Users user = new Users(email,name,usn);
```

```
if (email.isEmpty()) {
    Email.setError("Email is Required");
    Email.requestFocus();
    return;
}
if (!Patterns.EMAIL_ADDRESS.matcher(email).matches()) {
    Email.setError("Email is Invalid");
    Email.requestFocus();
    return;
}
```

```
if (name.isEmpty()) {
    Fname.setError("Name is Required");
    Fname.requestFocus();
    return;
}
if (usn.isEmpty()) {
    Usn.setError("Usn is Required");
    Usn.requestFocus();
    return;
}
if (passwd.isEmpty()) {
    Password.setError("Password is Required");
    Password.requestFocus();
    return;
}
if (passwd.length() < 6) {
    Password.setError("Password is too short");
    Password.requestFocus();
    return;
}
```

```
    }

    database.collection("Users").document(name).set(user);

    mAuth.createUserWithEmailAndPassword(email, passwd).addOnCompleteListener(new
    OnCompleteListener<AuthResult>() {
        @Override
        public void onComplete(@NonNull Task<AuthResult> task) {
            if (task.isSuccessful()) {
                Toast.makeText(SignUpActivity.this, "User Registered Successfully",
                Toast.LENGTH_LONG).show();
                Intent signIn = new Intent(SignUpActivity.this, SignInActivity.class);
                startActivity(signIn);

            } else {
                Toast.makeText(SignUpActivity.this, "Failed to Register! Try Again",
                Toast.LENGTH_LONG).show();

            }
        }
    });
}
```

### 2.Users.java

```
package com.example.cafeteria;

public class Users {
    String email;
    String name;
    String usn;

    public Users();
    public Users(String email, String name,String usn)
    {
        this.email = email;
        this.name = name;
    }
}
```

```
        this.usn = usn;
    }

    public String getEmail() {
        return email;
    }

    public void setEmail(String email) {
        this.email = email;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getUsn() {
        return usn;
    }

    public void setUsn(String courseDuration) {
        this.usn = usn;
    }
}
```

### 3.SignInActivity.java

```
package com.example.cafeteria;

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

import android.content.Intent;
import android.os.Bundle;
import android.util.Patterns;
import android.view.View;
import android.widget.Button;
```

```
import android.widget.EditText;
import android.widget.ProgressBar;
import android.widget.Toast;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.firestore.DocumentReference;
import com.google.firebase.firestore.FirebaseFirestore;

public class SignInActivity extends AppCompatActivity implements View.OnClickListener {
    private Button register;
    private EditText email,Password;
    private Button SignIn;

    FirebaseFirestore database = FirebaseFirestore.getInstance();
    private FirebaseAuth mAuth;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_sign_in);
        register= findViewById(R.id.Register);
        register.setOnClickListener(this);
        SignIn = findViewById(R.id.signin);
        SignIn.setOnClickListener(this);

        email = findViewById(R.id.emailSi);
        Password = findViewById(R.id.passwdSi);
        mAuth = FirebaseAuth.getInstance();
    }

    @Override
    public void onClick(View v) {
        switch (v.getId()){
            case R.id.Register:
```

```
        startActivity(new Intent(SignInActivity.this,SignUpActivity.class));
        break;
    case R.id.signin:
        userLogin();
        break;

    }

    if(v.equals(SignIn)){
        userLogin();
    }

}

private void userLogin() {
    String Email = email.getText().toString().trim();
    String passwd = Password.getText().toString().trim();

    if(Email.isEmpty()){
        email.setError("Email is required !");
        email.requestFocus();
        return;
    }

    if(!Patterns.EMAIL_ADDRESS.matcher(Email).matches()){
        email.setError("Enter A Valid Email!!!");
        email.requestFocus();
        return;
    }

    if(passwd.isEmpty()){
        Password.setError("Password is Required!");
        Password.requestFocus();
        return;
    }

    if(passwd.length()<6){
        Password.setError("Min password length is 6 characters");
        Password.requestFocus();
        return;
    }
}
```

```
mAuth.signInWithEmailAndPassword(Email,passwd).addOnCompleteListener(new
OnCompleteListener<AuthResult>() {
    @Override
    public void onComplete(@NonNull Task<AuthResult> task) {
        if(task.isSuccessful()){
            String currentUser = Email;
            Bundle username = new Bundle();
            username.putString("key1",currentUser);
            Toast.makeText(getApplicationContext(),"Welcome
User",Toast.LENGTH_LONG).show();
            startActivity(new
Intent(SignInActivity.this,MainActivity.class).putExtras(username));

        }
        else{
            Toast.makeText(SignInActivity.this,"Failed to Login! Check
Credentials",Toast.LENGTH_LONG).show();
        }

    }
});

}
}
```

### 4. MainActivity.java

```
package com.example.cafeteria;

import androidx.appcompat.app.AppCompatActivity;
import androidx.constraintlayout.widget.ImageFilterButton;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.ImageButton;

import com.google.firebase.database.DatabaseReference;
```



```
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.firestore.FirebaseFirestore;

public class MainActivity extends AppCompatActivity {

    ImageButton dosa,idli,coffee,cake,samosa,bread,pani;
    EditText unameDisp;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        FirebaseFirestore database = FirebaseFirestore.getInstance();

        Bundle username = getIntent().getExtras();

        String Uname = username.getString("key1","default");

        dosa = (ImageButton) findViewById(R.id.dosaButton);
        idli = (ImageButton) findViewById(R.id.idliButton);
        coffee = (ImageButton) findViewById(R.id.coffeeButton);
        cake = (ImageButton) findViewById(R.id.cakeButton);
        samosa = (ImageButton) findViewById(R.id.samosaButton);
        bread = (ImageButton) findViewById(R.id.breadButton);
        pani = (ImageButton) findViewById(R.id.paniButton);

        unameDisp = findViewById(R.id.unameDisp);

        unameDisp.setText(Uname);

        dosa.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent dosaAct = new Intent(MainActivity.this,DosaActivity.class);
                dosaAct.putExtras(username);
                startActivity(dosaAct);
            }
        })
    }
}
```

```
});
```

```
idli.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        Intent idliAct = new Intent(MainActivity.this,IdliVadaActivity.class);  
        idliAct.putExtras(username);  
        startActivity(idliAct);  
    }  
});
```

```
coffee.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        Intent coffeeAct = new Intent(MainActivity.this,CoffeeActivity.class);  
        coffeeAct.putExtras(username);  
        startActivity(coffeeAct);  
    }  
});
```

```
cake.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        Intent cakeAct = new Intent(MainActivity.this,CakeActivity.class);  
        cakeAct.putExtras(username);  
        startActivity(cakeAct);  
    }  
});
```

```
samosa.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        Intent samosaAct = new Intent(MainActivity.this,SamosaActivity.class);  
        samosaAct.putExtras(username);  
        startActivity(samosaAct);  
    }  
});
```

```
bread.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent breadAct = new Intent(MainActivity.this,BreadPakodaActivity.class);
        breadAct.putExtras(username);
        startActivity(breadAct);
    }
});

pani.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent paniAct = new Intent(MainActivity.this,PaniPuriActivity.class);
        paniAct.putExtras(username);
        startActivity(paniAct);
    }
});

}
}
```

### 5. CoffeeActivity.java

```
package com.example.cafeteria;

import static android.content.ContentValues.TAG;

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

import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;

import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
```

```
import com.google.firebase.firestore.DocumentReference;
import com.google.firebase.firestore.FirebaseFirestore;
import com.google.firebase.firestore.SetOptions;

import java.util.HashMap;
import java.util.Map;

public class CoffeeActivity extends AppCompatActivity {

    EditText coffeeNos;
    Button coffeeAdd,coffeeSub,coffeeOrder;
    FirebaseFirestore database = FirebaseFirestore.getInstance();

    int nos;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_coffee);

        coffeeNos = (EditText) findViewById(R.id.coffeeNos);
        coffeeAdd = (Button) findViewById(R.id.coffeeAdd);
        coffeeSub = (Button) findViewById(R.id.coffeeSub);
        coffeeOrder = (Button) findViewById(R.id.coffeeOrder);

        Bundle username = getIntent().getExtras();

        String Uname = username.getString("key1","default");

        coffeeAdd.setOnClickListener(new View.OnClickListener()
        {
            @Override
            public void onClick(View view)
            {
                nos+=1;
                coffeeNos.setText(String.valueOf(nos));
            }
        });
    }
}
```

```
coffeeSub.setOnClickListener(new View.OnClickListener()
{
    @Override
    public void onClick(View view)
    {
        nos-=1;
        coffeeNos.setText(String.valueOf(nos));
    }
});

coffeeOrder.setOnClickListener(new View.OnClickListener()
{
    @Override
    public void onClick(View view)
    {
        coffeeStore(nos, Uname);
        Intent coffeeOrder = new Intent(CoffeeActivity.this, MainActivity.class);
        coffeeOrder.putExtras(username);
        startActivity(coffeeOrder);
    }
});

}

public void coffeeStore(int nos, String uname)
{
    Map<String, Object> user = new HashMap<>();
    user.put("Coffee Nos", nos);

    database.collection("Orders")
        .document(uname).set(user, SetOptions.merge());

}
}
```

## 6. OrderSummaryActivity.java

```
package com.example.cafeteria;
```

```
import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.widget.Button;
import android.widget.EditText;

import com.google.firebase.firestore.DocumentReference;
import com.google.firebase.firestore.FirebaseFirestore;

import java.util.Map;

public class OrderSummaryActivity extends AppCompatActivity {

    Button placeOrder;
    EditText coffeeNosOrder, idliNosOrder, dosaNosOrder, paniNosOrder, samosaNosOrder,
    breadNosOrder, cakeNosOrder;
    EditText coffeeOrderCost, idliOrderCost, dosaOrderCost, paniOrderCost, samosaOrderCost,
    breadOrderCost, cakeOrderCost;
    EditText orderTotal;

    FirebaseFirestore database = FirebaseFirestore.getInstance();

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_order_summary);

        placeOrder = findViewById(R.id.placeOrder);

        coffeeNosOrder = findViewById(R.id.coffeeNosOrder);
        idliNosOrder = findViewById(R.id.idliNosOrder);
        dosaNosOrder = findViewById(R.id.dosaNosOrder);
        paniNosOrder = findViewById(R.id.paniNosOrder);
        samosaNosOrder = findViewById(R.id.samosaNosOrder);
        breadNosOrder = findViewById(R.id.breadNosOrder);
        cakeNosOrder = findViewById(R.id.cakeNosOrder);

        coffeeOrderCost = findViewById(R.id.coffeeOrderCost);
```

```
        idliOrderCost = findViewById(R.id.idliOrderCost);
        dosaOrderCost = findViewById(R.id.dosaOrderCost);
        paniOrderCost = findViewById(R.id.paniOrderCost);
        samosaOrderCost = findViewById(R.id.samosaOrderCost);
        breadOrderCost = findViewById(R.id.breadOrderCost);
        cakeOrderCost = findViewById(R.id.cakeOrderCost);

        orderTotal = findViewById(R.id.orderTotal);

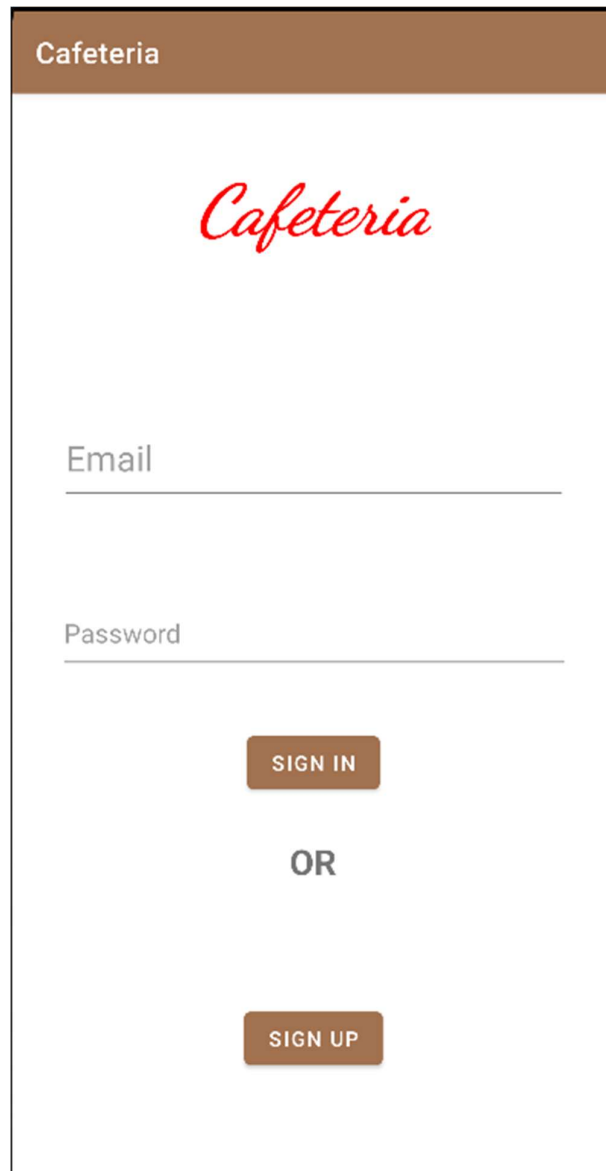
    }
}
```

## Chapter 6

### RESULTS

- **SignIn page:**

Every user's landing page is the SignIn page, where the user has to enter their Email and Password. If the user is new, he/she can Signup by clicking on the SignUp Button.



The image shows a mobile application interface for a 'Cafeteria'. At the top, there is a brown header bar with the word 'Cafeteria' in white. Below the header, the word 'Cafeteria' is written in a large, red, cursive font. Underneath the logo, there are two input fields: 'Email' and 'Password', each with a horizontal line for text entry. Below the 'Password' field, there is a brown button with the text 'SIGN IN' in white. Below this button, the word 'OR' is centered. At the bottom, there is another brown button with the text 'SIGN UP' in white.

**Fig 6.1 SignIn page**



- **SignUp Page:**

In this page the user can register to the App by providing details like Full name, USN, Email and Password.



Cafeteria

*Cafeteria*

Username

USN

Email

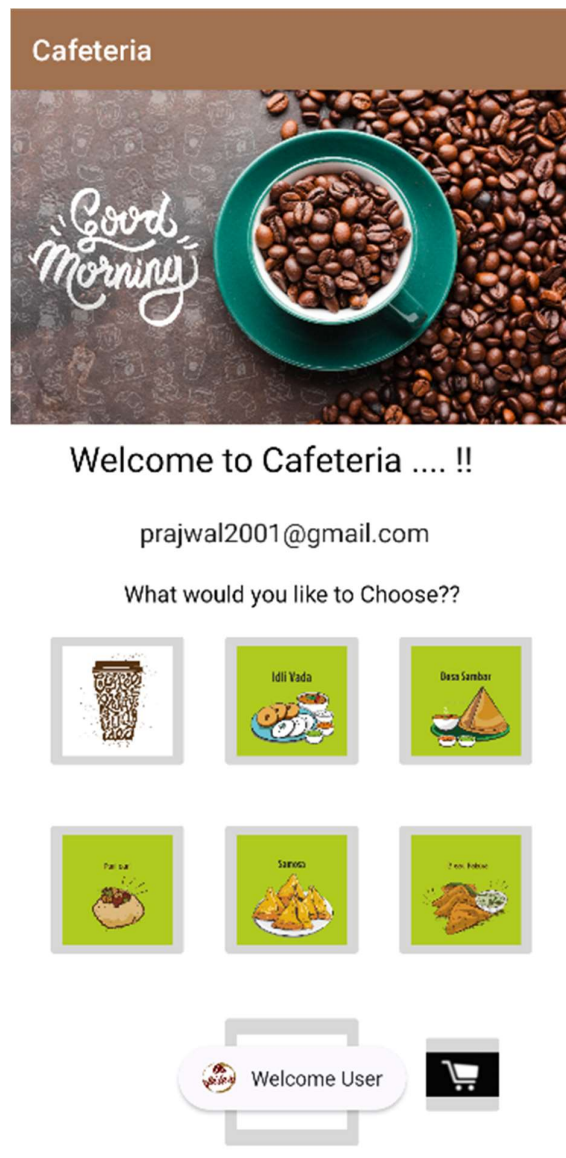
Password

SIGN UP

**Fig 6.2 SignUp Page**

- **Landing Page:**

After Logging into the App the user is greeted by this page where he/she can click on any item and start ordering. He/she can also view their order summary by clicking on the floating action button.



**Fig 6.3 Landing Page**

- **Coffee Order Page:**

When the user clicks on the Coffee Image button on the Landing Page, he/she is taken to this page where they can choose the quantity of Coffee by using two buttons to increase or decrease the quantity.

Cafeteria



Coffee

+

-

Nos.

ORDER

**Fig 6.4 Coffee Order Page**








- **Order Summary Page:**

All the information about the current Order, Individual cost of each item and the Total Cost is Displayed in this page. On clicking on Place Order button, the Order is officially taken and sent to the Kitchen for preparation.

Cafeteria

Order Summary

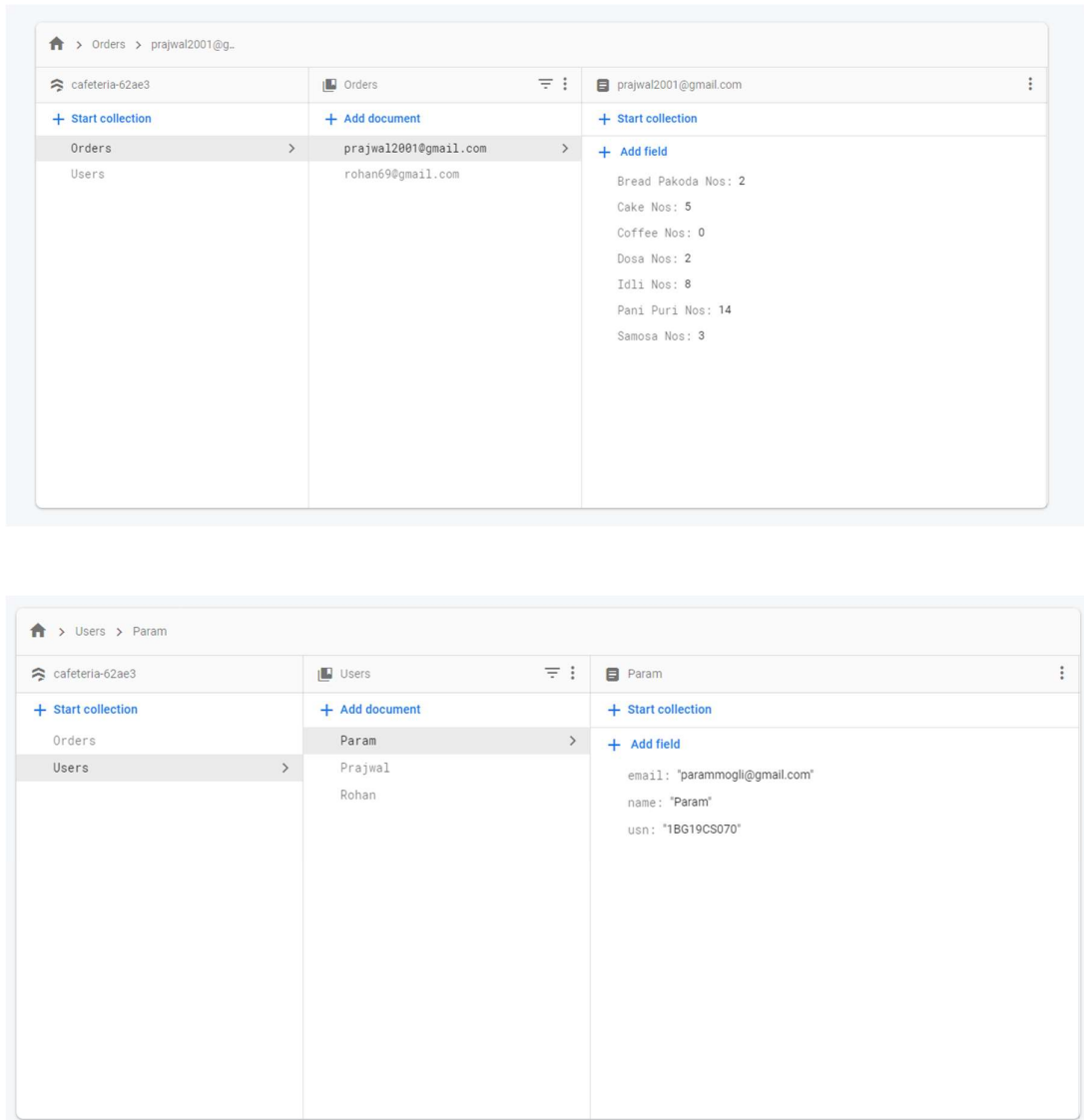
PLACE ORDER

	Coffee 10	<input type="text"/>	<input type="text"/>
	Idli Vada 20	<input type="text"/>	<input type="text"/>
	Dosa 40	<input type="text"/>	<input type="text"/>
	Pani Puri 30	<input type="text"/>	<input type="text"/>
	Samosa 20	<input type="text"/>	<input type="text"/>
	Bread Pakoda 20	<input type="text"/>	<input type="text"/>
	Cake 20	<input type="text"/>	<input type="text"/>
Total Cost		<input type="text"/>	

**Fig 6.5 Order Summary Page**

- **FireBase Entries:**

This is a snapshot of the Real Time Database which shows the users and their orders.



**Fig 6.6 Firebase Realtime Database**

## Chapter 7

### CONCLUSION AND FUTURE ENHANCEMENTS

#### CONCLUSION:

Mobile technologies are a must in the restaurant business. They help engage customers, increase loyalty, and grow profit. Using the application, the end users register online, logs in and selects the food from the menu available to order food online. Once the customer selects the required food item and specify the quantity the chef will be able to see the results on the screen and start processing the food. This application nullifies the need of a waiter or reduces the workload of the waiter. The advantage is that in a crowded cafeteria there will be chances that the cafe is overloaded with orders, and they are unable to meet the requirements of the customer in a satisfactory manner. Therefore, by using this application, the users can directly place the order for food to the chef online.

In conclusion we have developed a Cafeteria Application which mitigates the above issues and provides an interactive interface to the users.

#### FUTURE ENHANCEMENTS:

In future the proposed system would:

- learn each customer's eating habits and routines
- suggest a food item to the customer based on his previous choices or his nutrients requirement
- record the consumption pattern of food items from the menu. This would allow the admin to prepare only demanded items.

With the customer's ever increasing busy schedules, many times he may even forget to place an order on time. Our system would allow the customer to place order automatically, allowing him to not worry about it.

In a work environment with always a time crunch, people don't even think about eating food, let alone consider their nutrition requirements. Our system tries to make the task from placing an order to receiving it comparatively easier and eating the same healthier.

## **BIBLIOGRAPHY**

- [1] Connect to Firebase | Android Developers
- [2] <https://www.developerfusion.com/article/84479/jsp-for-netdevelopers/>
- [3] <https://developer.android.com/>
- [4] <https://docs.oracle.com/javase/tutorial/>
- [6] Android Programming Tutorials by Mark L. Murphy