VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JNANA SANGAMA", Belagavi-590018

A Mini Project Report on
"Food Ordering Application"

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR

THE AWARD OF DEGREE OF

BACHELOR OF ENGINEERING IN
INFORMATION SCIENCE AND ENGINEERING

SUBMITTED BY

PRADEEP GOWDA H (1JB20IS044)

Under the Guidance of

Dr. PAVITRA BAI S
Associate Professor,
Dept. of ISE, SJBIT
Bengaluru-60

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

SJB INSTITUTE OF TECHNOLOGY

BGS HEALTH AND EDUCATION CITY, KENGERI, BENGALURU-560060, KARNATAKA, INDIA.

2022-2023

|| Jai Sri Gurudev ||
Sri Adichunchanagiri Shikshana Trust ®

SJB INSTITUTE OF TECHNOLOGY

BGS Health & Education City, Kengeri, Bengaluru – 560 060

Department of Information Science & Engineering

CERTIFICATE

Certified that the Mini-project work entitled "**FOOD ORDERING APPLICATION",** is bonafide work carried out by **PRADEEP GOWDA H (1JB20IS044)**, a bonafide student of **SJB Institute of Technology,** in partial fulfilment for 6th semester in **INFORMATION SCIENCE AND ENGINEERING** of the **VISVESVARAYA TECHNOLOGICAL UNIVERSITY**, **BELAGAVI** during the academic 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. The mini project report has been approved as it satisfies the academic requirements in respect of Mini Project prescribed for the said degree.

Dr. Pavitra Bai S **Associate Professor** **Dept. of ISE, SJBIT**	**Dr. Shashidhara H R** **Professor & Head** **Dept. of ISE, SJBIT**

EXTERNAL VIVA

Name of the Examiners	**Signature with Date**
1. _____	_____
2. _____	_____

ACKNOWLEDGEMENT

I would like to express my profound thanks to His Divine Soul **Padmabhushan Sri Sri Sri Dr. Balagangadharanatha Maha Swamiji** and His Holiness **Jagadguru Sri Sri Sri Dr. Nirmalanandanatha Maha Swamiji** for providing me an opportunity to pursue my academics in this esteemed institution.

I would also like to express my profound thanks to **Revered Sri Sri Dr. Prakashnath Swamiji**, Managing Director, SJB Institute of Technology, for his continuous support in providing amenities to carry out this mini project in this admired institution.

I express my gratitude to **Dr. K. V. Mahendra Prashanth**, Principal, SJB Institute of Technology, for providing excellent facilities and academic ambience, which have helped me in satisfactory completion of mini project work.

I extend my sincere thanks to **Dr. Shashidhara H.R**, Head of the Department, Information Science and Engineering, for providing an invaluable support throughout the period of mini project work.

I express my truthful thanks to **Dr. Pavitra Bai S,** Associate professor, Mini Project Coordination and guidance**,** who has been a source of inspiration to me and has extended his/her support throughout the mini project duration.

Finally, I take this opportunity to extend my earnest gratitude and respect to my parents, Teaching & Non-teaching staffs of the department, the library staff and all my friends, for their continuous support and encouragement.

PRADEEP GOWDA H (1JB20IS044)

ABSTRACT

This abstract outlines a proposal for the development of an innovative mobile food ordering application that aims to provide users with a seamless and convenient food ordering experience. The proposed application seeks to simplify the process of ordering food by offering a user-friendly interface and a range of advanced features.The application will enable users to browse through an extensive selection of restaurants, explore diverse menus, customize their orders, and effortlessly place them. Real-time order tracking will allow users to monitor the progress of their orders, ensuring transparency and minimizing uncertainties. Secure payment options will be implemented to guarantee the safety of users' financial information.To ensure widespread accessibility, the application will be developed for Android platforms, catering to a wide range of users. Cross-platform compatibility will enable users to access the application seamlessly from their preferred devices, be it smartphones or tablets.The development process will emphasize, user feedback at each stage to optimize the application's functionality and user interface.By offering a user-friendly interface this proposed food ordering application aims to revolutionize the way users order food. The application's emphasis on convenience, efficiency, and personalization has the potential to significantly enhance the overall food ordering experience for users, enabling them to enjoy their favorite meals with ease.

TABLE OF CONTENTS

CONTENTS

Chapters Organization: **Page :**

Chapter 1

INTRODUCTION

1.1 Overview

With the advent of new mobile technologies, the mobile application industry is advancing rapidly. Consisting of several operating systems like Symbian OS, iOS, blackberry, etc., Android OS is recognized as the most widely used, popular and user- friendly mobile platform. This open-source Linux kernel-based operating system offers high flexibility due to its customization properties making it a dominant mobile operating system. Android applications are programmed in java language. Google android SDK delivers a special software stack that provides developers an easy platform to develop android applications. Moreover, developers can make use of existing java IDEs which provides flexibility to the developers. Java libraries are predominant in the process of third- party application development. Cross-platform approaches make sure that developers do not have to develop platform-dependent applications. With the help of these approaches, an application can be deployed to several platforms without the need for changes in coding.

1.2 Problem Statement

Existing food ordering applications suffer from cluttered and confusing user interfaces (UI), resulting in a subpar user experience. Users struggle to find relevant information, navigate menus, and place orders efficiently. Inconsistent design elements and poor organization of content further contribute to user frustration. The lack of a responsive UI also limits accessibility across devices. Additionally, the absence of visually enticing food images and descriptions hampers users' ability to make informed choices. Therefore, there is a need for a food ordering application that prioritizes a clean and intuitive UI, streamlined navigation, and visually appealing content to enhance user satisfaction and usability.

1.3 Mobile Application Development Need & Importance

In the past few years mobile app development has become a booming industry.

Currently, it is estimated that there are 2.3 million mobile app developers who are devoted to keeping up with the industry demand.

In fact, according to Apple, in 2013 1.25 million apps were registered in the Apple app store and accounted for 50 billion downloads and $5 billion paid to developers.

With these types of industry numbers, it soon becomes clear that mobile app development is a key factor for business success.

With the growing number of people accessing the Internet via smartphones and tablets, mobile app development has the unique ability to access a large number of potential consumers.

Not only have the sales of smartphone and tablets increased, but the amount of mobile apps installed has also grown exponentially. The Pew Research Internet Project indicates that approximately 50 percent of all smartphone users have mobile apps installed; of this percentage, two-thirds of the individuals are regular mobile app users. These statistics show that mobile apps have a unique opportunity to engage with an entirely new type of customer, one whom is constantly connected to the Internet and the global commerce space. In essence, a mobile app allows you to have millions of new customers at your fingertips. All that is left for you to do, is to develop an effective app and reap the benefits of your labors.

There are multiple benefits to creating and distributing a mobile app. Below are a few of the top benefits for businesses across a wide-variety of industries.

- **Reinforce your Brand**
  Mobile apps offer the unique opportunity for brand reinforcement through a new channel. Through mobile apps, customers are encouraged to download the free branded version, where they can customize preferences to fit their specific needs.
- **Increase your Accessibility**
  Smartphone and tablet users are constantly on the go; this means that they don't always have time to sign into a mobile website. And these mobile websites are designed for readability and navigation, NOT for process management. Mobile apps allow users to have easy, functional access to information, products, services and processes that they need in real-time and are optimized for hands on interaction.
- **Increase Sell-Through**
  Recent analysis suggests that mobile app users spend more time on a company's mobile app, then they spend on the company's mobile website.

- **Reduce On-premise costs**
  Most of the services that you provide at your business premises can be provided through android mobile applications. This would put you in a position where you do not need to pay workers to do that particular job.
- **Scope for Innovation**
  With every year, Android brings up innovative ideas and trends that symbolize the future. The devices and technologies used by users to interact with business changes pertaining to users' behaviors and needs.

As we continue to evolve into a mobile-centric society, it comes as no surprise that mobile apps are at the center of the developmental push. Developing a mobile app can go a long way towards propelling your company into the hands of new customers and future business success.

1.4 Android Studio

Android Studio is the official Integrated Development Environment (IDE) for Android app development, based on IntelliJ IDEA. On top of IntelliJ's powerful code editor and developer tools, Android Studio offers even more features that enhance your productivity when building Android apps, such as:

1. A flexible Gradle-based build system.
2. A fast and feature-rich emulator.
3. A unified environment where you can develop for all Android devices.
4. Apply Changes to push code and resource changes to your running app without restarting your app.
5. Code templates and GitHub integration to help you build common app features and import sample code.
6. Extensive testing tools and frameworks.
7. Lint tools to catch performance, usability, version compatibility, and other problems.
8. C++ and NDK support

1.4.1 Project Structure

Each project in Android Studio contains one or more modules with source code files and resource files. Types of modules include:

- Android app modules.
- Library modules.
- Google App Engine modules.

Each app module contains the following folders:

- Manifests - Contains the AndroidManifest.xml file.
- Java - Contains the Java source code files, including JUnit test code.
- Res – Contains all non-code resources, such as XML Layouts, UI Strings and bitmap images.

1.4.2 The User Interface

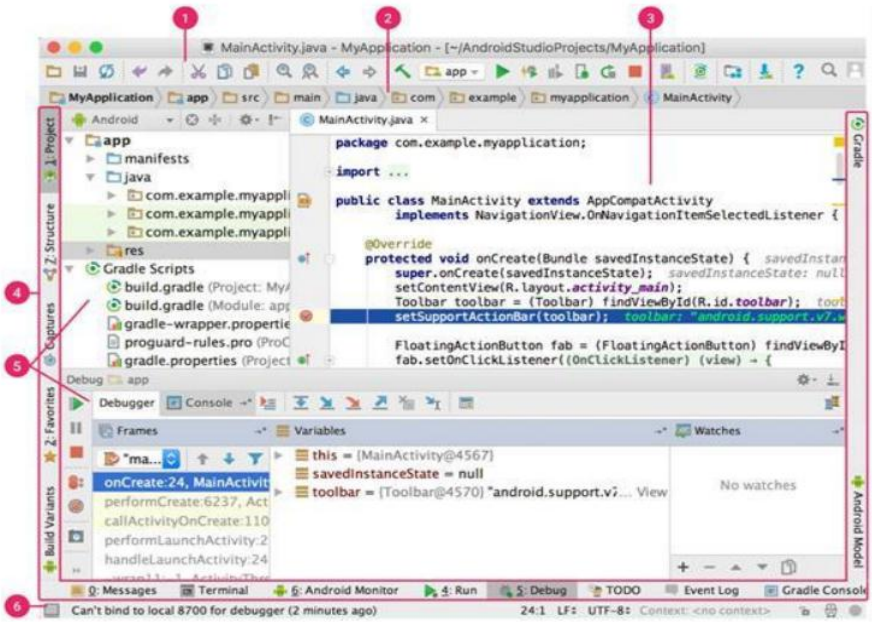


Figure 1.4.2. The Android Studio main window.

- 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. For example, when viewing a layout file, the editor displays the Layout Editor.
- The tool window bar runs around the outside of 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. You can expand them and collapse them.

• The status bar displays the status of your project and the IDE itself, as well as any warnings or messages.

1.4.4 Gradle build System

Android Studio uses Gradle as the foundation of the build system, with more Android specific capabilities provided by the Android plugin for Gradle. This build system runs as an integrated tool from the Android Studio menu, and independently from the command line. You can use the features of the build system to do the following:

- Customize, configure, and extend the build process.
- Create multiple APKs for your app, with different features using the same projects and modules.
- Reuse code and resources across source sets.

By employing the flexibility of Gradle, you can achieve all of this without modifying your app's core source files. Android Studio build files are named build. Gradle. They are plain text files that use Groovy syntax to configure the build with elements provided by the Android plugin for Gradle. Each project has one top-level build file for the entire project and separate module-level build files for each module. When you import an existing project, Android Studio automatically generates the necessary build files.

Chapter 2

SYSTEM REQUIREMENTS

2.1 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 10
- Java Development kit
- Android Studio

2.2 Hardware Requirements

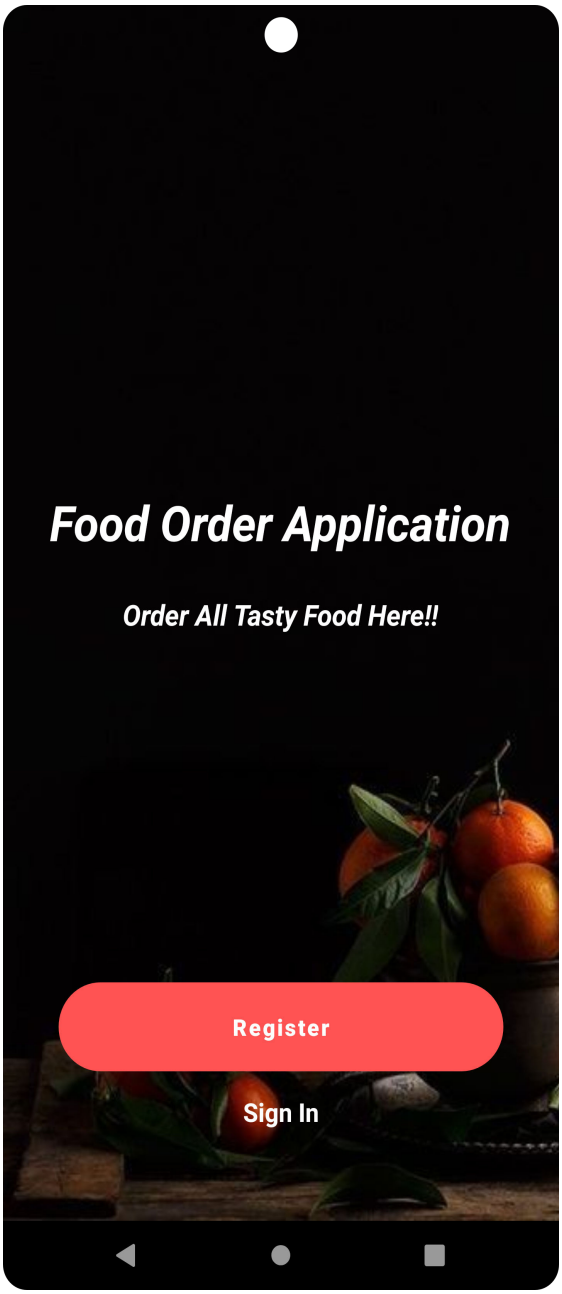
The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware.

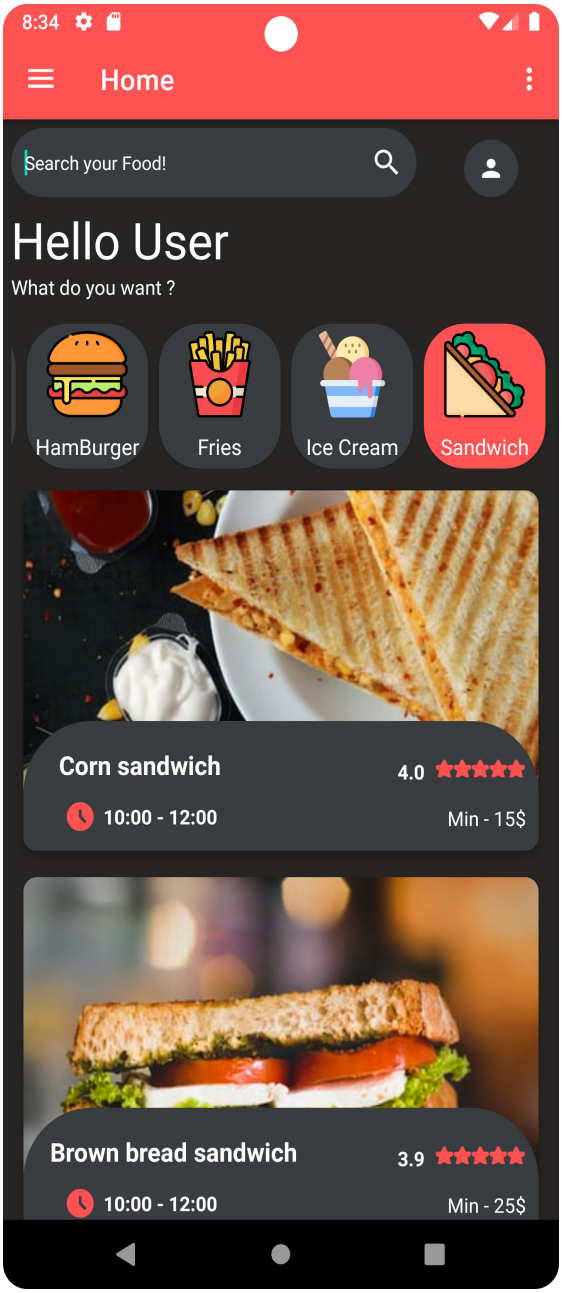
- CPU : Intel or AMD processor.
- Cores : Dual-Core (Quad-Core recommended).
- RAM : minimum 4GB (>4GB recommended).
- Graphics : Intel Integrated Graphics or AMD Equivalent.
- Display Resolution : 1366x768 (1920x1080 recommended).

Chapter 3

SYSTEM DESIGN

3.1 XML DESIGN





3.2 XML Code

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.drawerlayout.widget.DrawerLayout 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:id="@+id/drawer_layout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:fitsSystemWindows="true"
    tools:openDrawer="start">

    <include
        android:id="@+id/app_bar_main"
        layout="@layout/app_bar_main"
        tools:ignore="true"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

    <com.google.android.material.navigation.NavigationView
        android:id="@+id/nav_view"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_gravity="start"
        android:fitsSystemWindows="true"
        app:headerLayout="@layout/nav_header_main"
        app:menu="@menu/activity_main_drawer"
        android:background="@color/gray"
        app:itemIconTint="@color/white"
        app:itemTextColor="@color/white">
        <Button
            android:layout_margin="5dp"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Log Out"
            android:layout_gravity="bottom"
            android:textAllCaps="false"
            android:textStyle="bold"
            android:padding="15dp"
            android:onClick="login"/>
    </com.google.android.material.navigation.NavigationView>
</androidx.drawerlayout.widget.DrawerLayout>
```

3.3 HOME PAGE XML CODE:

```
<?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"
    app:layout_behavior="@string/appbar_scrolling_view_behavior"
    tools:showIn="@layout/app_bar_main">

    <fragment
        android:id="@+id/nav_host_fragment_content_main"
        android:name="androidx.navigation.fragment.NavHostFragment"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        app:defaultNavHost="true"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:navGraph="@navigation/mobile_navigation" />

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

3.3 JAVA CODE

Main Activity.java

```
package com.example.myapplication;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.view.Menu;

import com.example.myapplication.activities.LoginActivity;
import com.example.myapplication.activities.MainActivity2;import
com.example.myapplication.activities.RegistrationActivity;import
com.google.android.material.snackbar.Snackbar;
import com.google.android.material.navigation.NavigationView;

import androidx.navigation.NavController;

import androidx.navigation.Navigation;

import androidx.navigation.ui.AppBarConfiguration;

import androidx.navigation.ui.NavigationUI;
```

```
import com.example.myapplication.databinding.ActivityMainBinding;
public class MainActivity extends AppCompatActivity {

    private AppBarConfiguration mAppBarConfiguration;
    private ActivityMainBinding binding;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        binding = ActivityMainBinding.inflate(getLayoutInflater());
        setContentView(binding.getRoot());

        setSupportActionBar(binding.appBarMain.toolbar);
        DrawerLayout drawer = binding.drawerLayout;
        NavigationView navigationView = binding.navView;
        // Passing each menu ID as a set of Ids because each
        // menu should be considered as top level destinations.
        mAppBarConfiguration = new AppBarConfiguration.Builder(
                R.id.nav_home, R.id.nav_daily_meal, R.id.nav_favourite,R.id.nav_my_cart)
                .setOpenableLayout(drawer)
                .build();
        NavController navController = Navigation.findNavController(this,
R.id.nav_host_fragment_content_main);
        NavigationUI.setupActionBarWithNavController(this, navController, mAppBarConfiguration)
;
        NavigationUI.setupWithNavController(navigationView, navController);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }

    @Override
    public boolean onSupportNavigateUp() {
        NavController navController = Navigation.findNavController(this,
R.id.nav_host_fragment_content_main);
        return NavigationUI.navigateUp(navController, mAppBarConfiguration)
                || super.onSupportNavigateUp();
    }

    public void login(View view) {
        startActivity(new Intent(getApplicationContext(), MainActivity2.class));
        startActivity(new Intent(getApplicationContext(), MainActivity2.class));

    }
```

- **database.java**

```
package com.example.myapplication;

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;

import androidx.annotation.Nullable;

import java.util.ArrayList;

public class Database extends SQLiteOpenHelper {
    public Database(@Nullable Context context, @Nullable String name, @Nullable
SQLiteDatabase.CursorFactory factory, int version) {
        super(context, name, factory, version);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        String qry1 = "create table users(username text,email text,password text)";
        db.execSQL(qry1);

        String qry2="create table cart(username text,product text,price float,otype text)";
        db.execSQL(qry2);
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {

    }
    public void register(String username,String email,String password){
        ContentValues cv = new ContentValues();
        cv.put("username",username);
        cv.put("email",email);
        cv.put("password",password);
        SQLiteDatabase db = getWritableDatabase();
        db.insert("users",null,cv);
        db.close();
    }
    public int login(String username,String password){
        int result=0;
        String str[]=new String[2];
        str[0]=username;
        str[1]=password;
        SQLiteDatabase db=getReadableDatabase();
        Cursor c=db.rawQuery("select * from users where username=? and password=?",str);
        if(c.moveToFirst()){
            result=1;
        }
        return result;
    }
```

```
public void addCart(String username,String product,float price,String otype){
        ContentValues cv = new ContentValues();
        cv.put("username",username);
        cv.put("product",product);
        cv.put("price",price);
        cv.put("otype",otype);
        SQLiteDatabase db = getWritableDatabase();
        db.insert("cart",null,cv);
        db.close();
    }

    public int checkCart(String username,String product){
        int result=0;
        String str[]=new String[2];
        str[0]=username;
        str[1]=product;
        SQLiteDatabase db=getReadableDatabase();
        Cursor c=db.rawQuery("select * from cart where username=? and product=?",str);
        if(c.moveToFirst()){
            result=1;
        }
        db.close();
        return result;
    }

    public void removeCart(String username,String otype){
        String str[]=new String[2];
        str[0]=username;
        str[1]=otype;
        SQLiteDatabase db=getWritableDatabase();
        db.delete("cart","username=? and otype=?",str);
        db.close();
    }

    public ArrayList getCartData(String username,String otype){
        ArrayList<String> arr=new ArrayList<>();
        SQLiteDatabase db=getReadableDatabase();
        String str[]=new String[2];
        str[0]=username;
        str[1]=otype;
        Cursor c=db.rawQuery("select * from cart where username=? and otype=?",str);
        if(c.moveToFirst()){
            do{
                String product=c.getString(1);
                String price=c.getString(2);
                arr.add(product+"$"+price);
            }while(c.moveToNext());
        }
        db.close();
        return arr;
    }
}
```

- **Login and register xml code**

```
<?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=".activities.LoginActivity">

    <ImageView
        android:id="@+id/imageView4"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:foreground="@drawable/foreground_activity"
        android:scaleType="centerCrop"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.0"
        app:srcCompat="@drawable/registerimage" />

    <TextView
        android:id="@+id/textView4"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Food Order App"
        android:textColor="#ffffff"
        android:textSize="40sp"
        android:textStyle="bold|italic"
        app:flow_verticalBias=".2"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="@+id/imageView4"
        app:layout_constraintHorizontal_bias="0.498"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.209" />

    <TextView
        android:id="@+id/textView5"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/sign_in"
        android:textColor="#FBFBFB"
        android:textSize="19sp"
        android:textStyle="bold"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@id/textView4"
        android:layout_marginTop="30dp"
        android:textAlignment="center"/>
```

```
<EditText
        android:id="@+id/editText2"
        android:layout_width="0dp"
        android:layout_height="48dp"
        android:layout_marginTop="30dp"
        android:backgroundTint="#CCCCCC"
        android:drawableLeft="@drawable/ic_launcher_foreground"
        android:drawablePadding="13dp"
        android:hint="@string/user_name"
        android:textColor="@color/white"
        android:textColorHint="#FBFBFB"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@id/textView5"
        app:layout_constraintWidth_percent=".8" />

    <EditText
        android:id="@+id/editText3"
        android:layout_width="0dp"
        android:layout_height="48dp"
        android:layout_marginTop="88dp"
        android:backgroundTint="#CCCCCC"
        android:drawableLeft="@drawable/ic_launcher_key"
        android:drawablePadding="13dp"
        android:hint="@string/password"
        android:textColor="@color/white"
        android:textColorHint="#FBFBFB"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@id/textView5"
        app:layout_constraintWidth_percent=".8"
        app:layout_goneMarginTop="10dp" />

    <Button
        android:onClick="mainActivity"
        android:textColor="@color/white"
        android:id="@+id/button"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:background="@drawable/btn_bg_design"
        android:padding="14dp"
        android:text="Sign In"
        android:textAllCaps="false"
        android:textSize="16sp"
        android:textStyle="bold"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/editText3"
        app:layout_constraintVertical_bias="0.6"
        app:layout_constraintWidth_percent=".8" />

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

- **DetailedDailyMealActivity.java**

```
package com.example.myapplication.activities;

import android.os.Bundle;
import android.widget.ImageView;

import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import com.example.myapplication.R;
import com.example.myapplication.adapters.DetailedDailyAdapter;
import com.example.myapplication.models.DetailedDailyModel;

import java.util.ArrayList;
import java.util.List;

public class DetailedDailyMealActivity extends AppCompatActivity {

    RecyclerView recyclerView;
    List<DetailedDailyModel> detailedDailyModelList;
    DetailedDailyAdapter dailyAdapter;
    ImageView imageView;


    @Override
    protected void onCreate(Bundle savedInstance) {
        super.onCreate(savedInstance);
        setContentView(R.layout.activity_detailed_daily_meal);

        String type=getIntent().getStringExtra("type");

        recyclerView=findViewById(R.id.detailed_rec);
        imageView=findViewById(R.id.cart_img);

        recyclerView.setLayoutManager(new LinearLayoutManager(this));
        detailedDailyModelList = new ArrayList<>();

        dailyAdapter = new DetailedDailyAdapter(detailedDailyModelList);
        recyclerView.setAdapter(dailyAdapter);

        if(type!=null && type.equalsIgnoreCase("breakfast")){
            detailedDailyModelList.add(new DetailedDailyModel(R.drawable.fav1,"Oat Breakfast","
Healthy","4.1","18","10am to 1pm"));
            detailedDailyModelList.add(new DetailedDailyModel(R.drawable.fav2,"House favourite
burger","Glutton as hell","4.5","40","10am to 1pm"));
            detailedDailyModelList.add(new DetailedDailyModel(R.drawable.fav3,"Noodles","Healthy
carb","4.8","40","10am to 1pm"));
            dailyAdapter.notifyDataSetChanged();
        }
```

```
  if(type!=null && type.equalsIgnoreCase("lunch")){
            imageView.setImageResource(R.drawable.lunch);
            detailedDailyModelList.add(new DetailedDailyModel(R.drawable.lunch1,"Rice and
beans","Stomach filler","4.4","39","2pm to 5pm"));
            detailedDailyModelList.add(new DetailedDailyModel(R.drawable.lunch2,"Complete
south indian","El Spicy","4.0","39","2pm to 5pm"));
            detailedDailyModelList.add(new DetailedDailyModel(R.drawable.lunch3,"Salmon dish","
Classique","4.0","45","2pm to 5pm"));
            dailyAdapter.notifyDataSetChanged();
        }
        if(type!=null && type.equalsIgnoreCase("dinner")){
            imageView.setImageResource(R.drawable.dinner);
            detailedDailyModelList.add(new DetailedDailyModel(R.drawable.dinner1,"Chicken dish"
,"Simple","4.4","70","7pm to 10pm"));
            detailedDailyModelList.add(new DetailedDailyModel(R.drawable.dinner2,"Rice pudding"
,"Low fat","4.1","50","7pm to 10pm"));
            detailedDailyModelList.add(new DetailedDailyModel(R.drawable.dinner3,"Ham and
potato","Filler","4.0","70","7pm to 10pm"));
            detailedDailyModelList.add(new DetailedDailyModel(R.drawable.dinner4,"Tomato soup
and bread","Light on spices","3.9","40","7pm to 10pm"));
            dailyAdapter.notifyDataSetChanged();
        }
        if(type!=null && type.equalsIgnoreCase("sweets")){
            imageView.setImageResource(R.drawable.sweets);
            detailedDailyModelList.add(new DetailedDailyModel(R.drawable.s2,"Doughnut","Fresh
and homemade","4.4","40","10am to 9pm"));
            detailedDailyModelList.add(new DetailedDailyModel(R.drawable.s3,"Candy ice cream","
Freshly brewed","4.8","10","10am to 9pm"));
            detailedDailyModelList.add(new DetailedDailyModel(R.drawable.s4,"Brownie","Freshly
baked","3.5","20","10am to 9pm"));
            dailyAdapter.notifyDataSetChanged();
        }
        if(type!=null && type.equalsIgnoreCase("coffee")){
            imageView.setImageResource(R.drawable.coffee);
            detailedDailyModelList.add(new DetailedDailyModel(R.drawable.coffee_cappuccino,"
Cappuccino","Classy","4.9","19","10am to 9pm"));
            detailedDailyModelList.add(new DetailedDailyModel(R.drawable.coffee_iced,"Iced
Coffee","Chiller","4.1","29","10am to 9pm"));
            detailedDailyModelList.add(new DetailedDailyModel(R.drawable.coffee_dalgona,"
Dalgona Coffee","Fan favourite","5.0","39","10am to 9pm"));
            dailyAdapter.notifyDataSetChanged();
        }
    }
}
```

- **DetailedDailyMealActivity.java**

```
package com.example.myapplication.activities;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import com.example.myapplication.Database;
import com.example.myapplication.MainActivity;
import com.example.myapplication.R;

public class RegistrationActivity extends AppCompatActivity {

    EditText edUsername,edEmail,edPassword,edConfirm;
    Button button;
    TextView tv;

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

        edUsername=findViewById(R.id.editText1);
        edEmail=findViewById(R.id.editText2);
        edPassword=findViewById(R.id.editText3);
        edConfirm=findViewById(R.id.editText4);
        button=findViewById(R.id.button);
        tv=findViewById(R.id.sign_in_txt);

        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                String username=edUsername.getText().toString();
                String email=edEmail.getText().toString();
                String password=edPassword.getText().toString();
                String confirm=edConfirm.getText().toString();
                Database db=new Database(getApplicationContext(),"foodAppUsers",null,1);
                if(username.length()==0 || email.length()==0 || password.length()==0 ||
confirm.length()==0){
                    Toast.makeText(getApplicationContext(), "Fill All Details",
Toast.LENGTH_SHORT).show();
                }
```

```
  else {
                if(password.compareTo(confirm)==0){
                    if (isValid(password)){
                        db.register(username,email,password);
                        Toast.makeText(getApplicationContext(), "User Registered",
Toast.LENGTH_SHORT).show();
                        startActivity(new Intent(RegistrationActivity.this,LoginActivity.class));
   }

else{
                        Toast.makeText(getApplicationContext(), "Password must contain at least 8
characters of letters with at least one number and one special character ",
Toast.LENGTH_SHORT).show();
                    }
                }else {
                    Toast.makeText(getApplicationContext(), "Password And Confirm Password Has
To Match", Toast.LENGTH_SHORT).show();
                }
            }
        }
    });
  }
  public static boolean isValid(String passwordHere) {
    int f1 = 0, f2 = 0, f3 = 0;
    if (passwordHere.length() < 8) {
        return false;
    } else {
        for (int p = 0; p < passwordHere.length(); p++) {
            if (Character.isLetter(passwordHere.charAt(p))) {
                f1 = 1;
            }
        }
        for (int r = 0; r < passwordHere.length(); r++) {
            if (Character.isDigit(passwordHere.charAt(r))) {
                f2 = 1;
            }
        }
        for (int s = 0; s < passwordHere.length(); s++) {
            char c=passwordHere.charAt(s);
            if (c>=33&&c<=46||c==64) {
                f3 = 1;
            }
        }
        if(f1==1&&f2==1&&f3==1)
            return true;
        return false;
    }
  }
  public void mainActivity(View view) {
    startActivity(new Intent(RegistrationActivity.this, MainActivity.class));
   }
  }
```

- **LoginActivity.java**

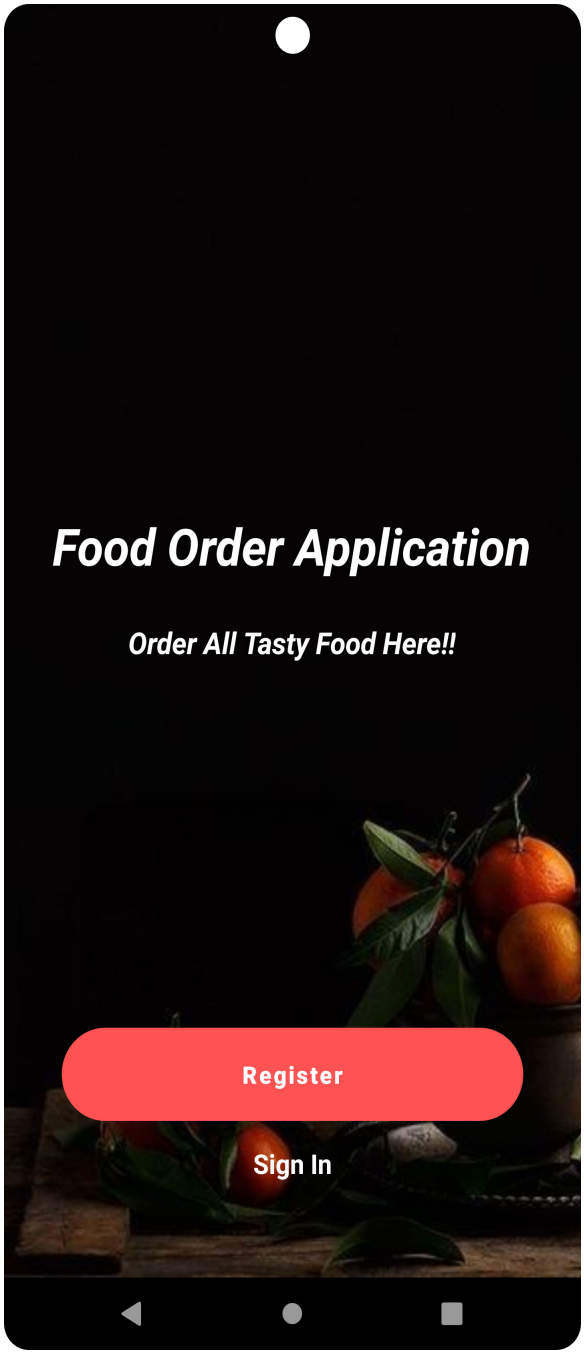
```
package com.example.myapplication.activities;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import com.example.myapplication.Database;
import com.example.myapplication.MainActivity;
import com.example.myapplication.R;

public class LoginActivity extends AppCompatActivity {

    EditText edUsername,edPassword;
    Button button;
    TextView tv;

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

        edUsername=findViewById(R.id.editText2);
        edPassword=findViewById(R.id.editText3);
        button=findViewById(R.id.button);
        tv=findViewById(R.id.register_txt);

        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                String username=edUsername.getText().toString();
                String password=edPassword.getText().toString();
                Database db=new Database(getApplicationContext(),"foodAppUsers",null,1);
                if(username.length()==0||password.length()==0){
                    Toast.makeText(LoginActivity.this, "Please fill all the details",Toast.
LENGTH_SHORT).show();
                }else{
                    if(db.login(username,password)==1){
                        Toast.makeText(getApplicationContext(), "Signing In", Toast.LENGTH_SHORT
).show();
                        SharedPreferences sharedPreferences=getSharedPreferences("shared_prefs",
Context.MODE_PRIVATE);
```

```
SharedPreferences.Editor editor=sharedPreferences.edit();
                    editor.putString("username",username);
                    //to save data with key and value
                    editor.apply();
                    startActivity(new Intent(LoginActivity.this,MainActivity.class));
                }
            else {
                    Toast.makeText(getApplicationContext(), "Invalid Credentials",
Toast.LENGTH_SHORT).show();
                }
             }
          }
       });

       tv.setOnClickListener(new View.OnClickListener() {
          @Override
          public void onClick(View view) {
             startActivity(new Intent(LoginActivity.this,RegistrationActivity.class));
          }
       });
    }

    public void register(View view) {
       startActivity(new Intent(LoginActivity.this, RegistrationActivity.class));
    }

    public void mainActivity(View view) {
       startActivity(new Intent(LoginActivity.this, MainActivity.class));

    }
}
```

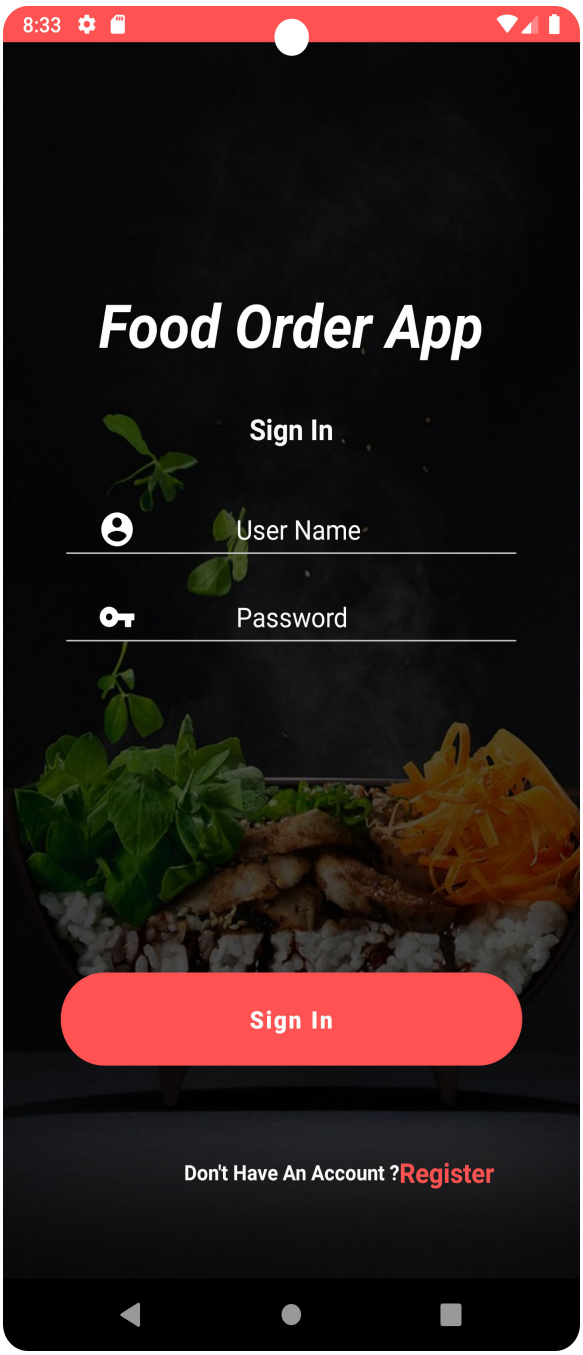
Chapter 4

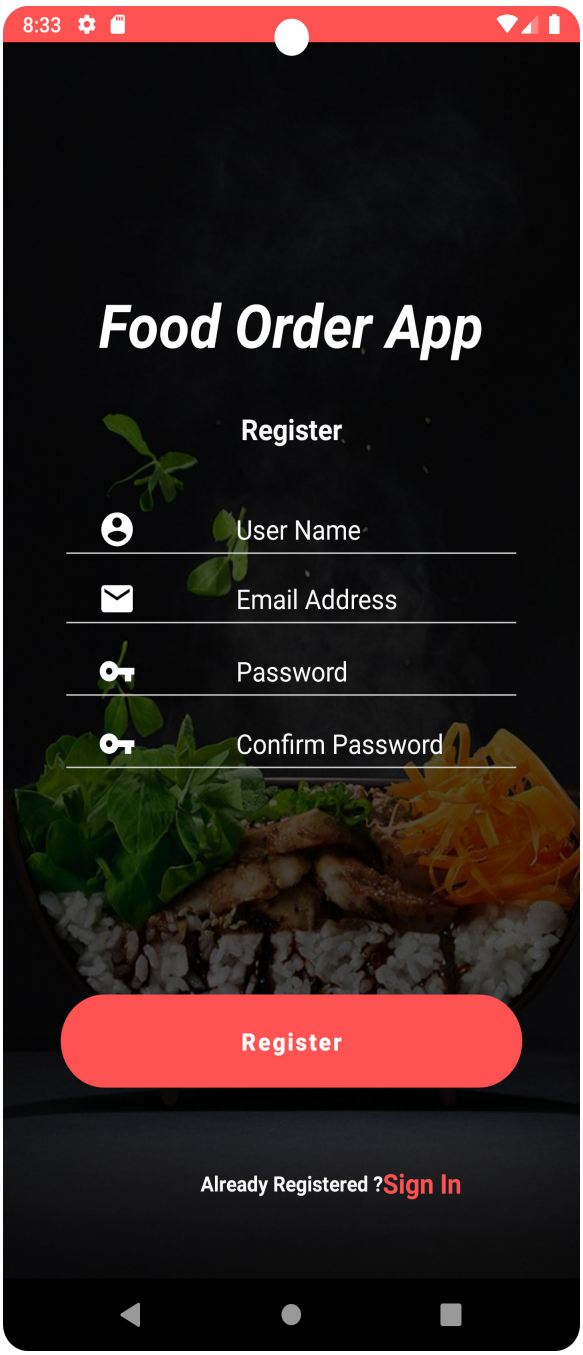
RESULT

- **START :**

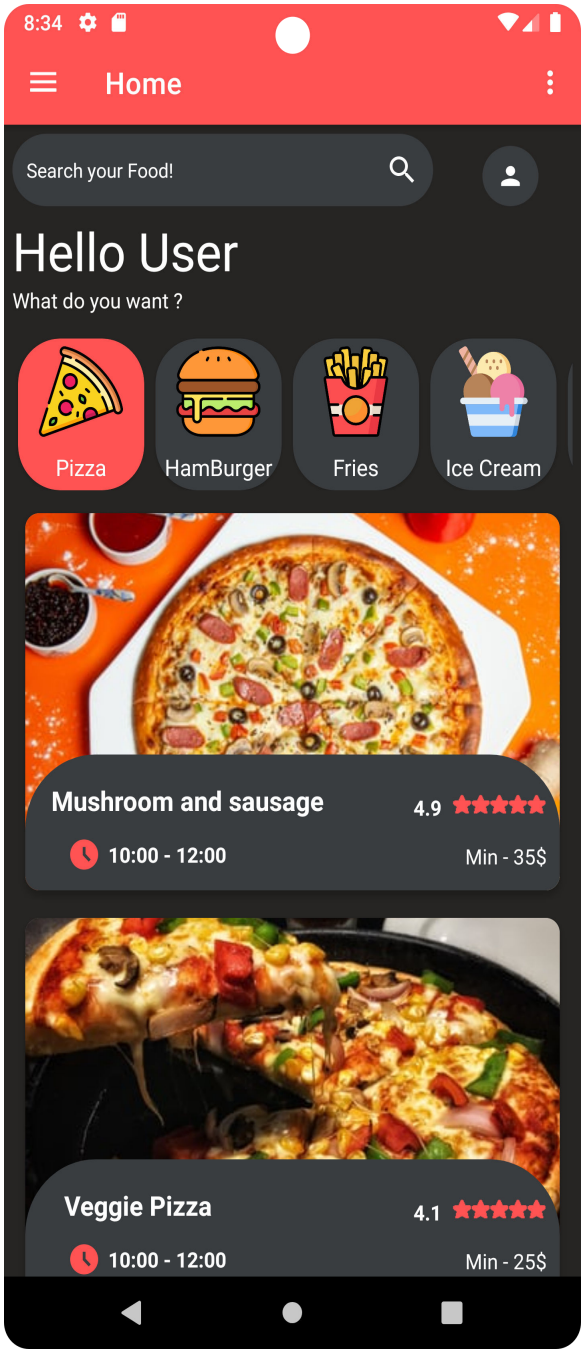


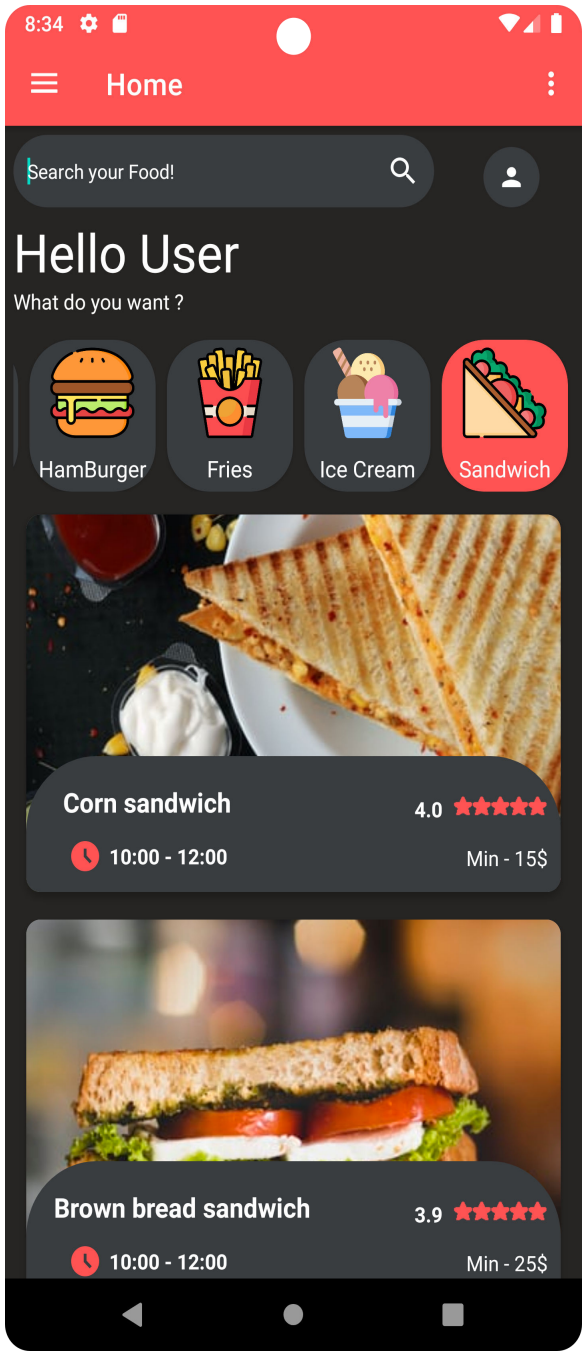
- **LOGIN AND REGISTER PAGE:**



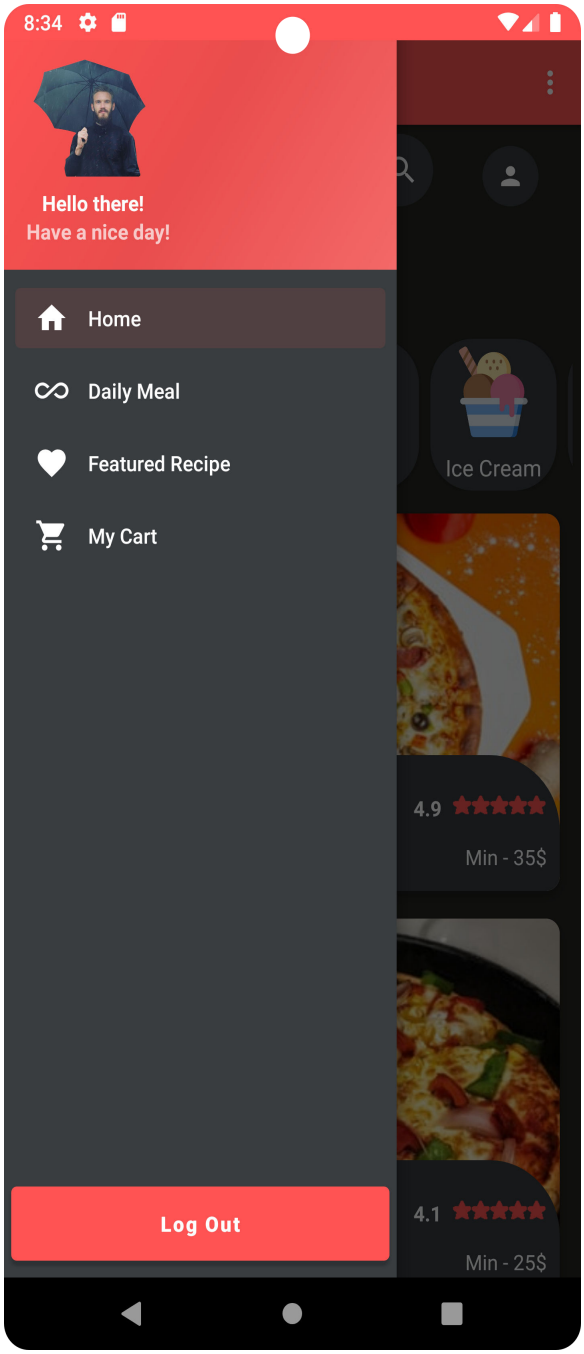


- **HOME PAGE:**

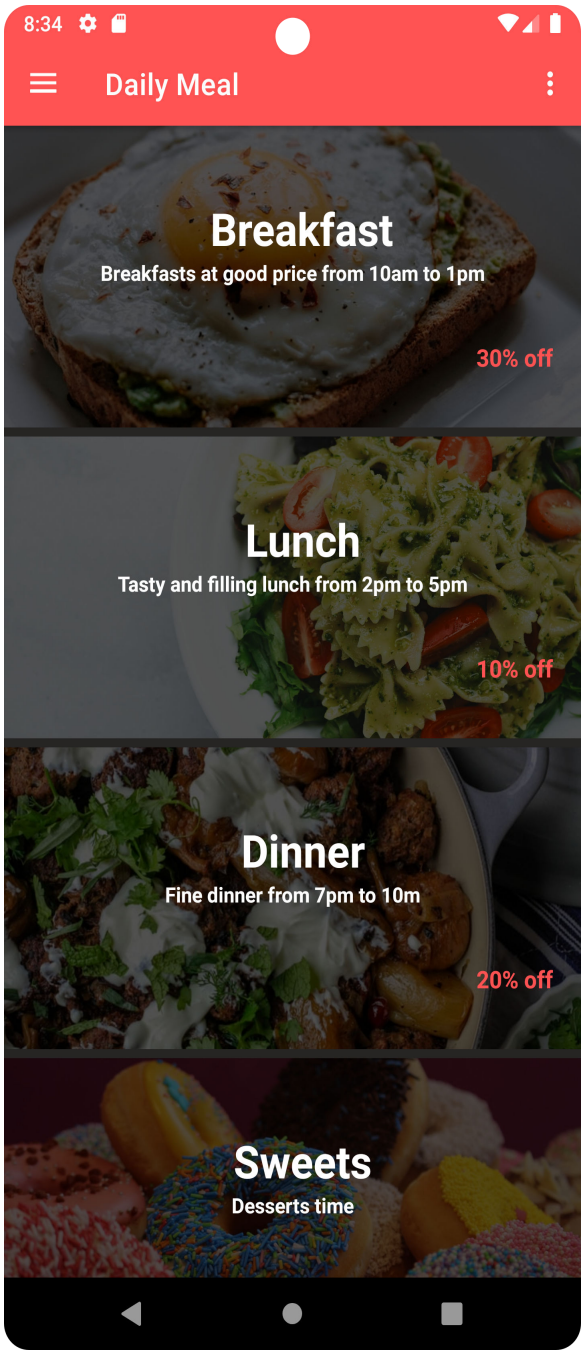

8:34
Home
Search your Food!
Hello User
What do you want ?
Pizza
HamBurger
Fries
Ice Cream
Mushroom and sausage
4.9
10:00 - 12:00
Min - 35$
Veggie Pizza
4.1
10:00 - 12:00
Min - 25$

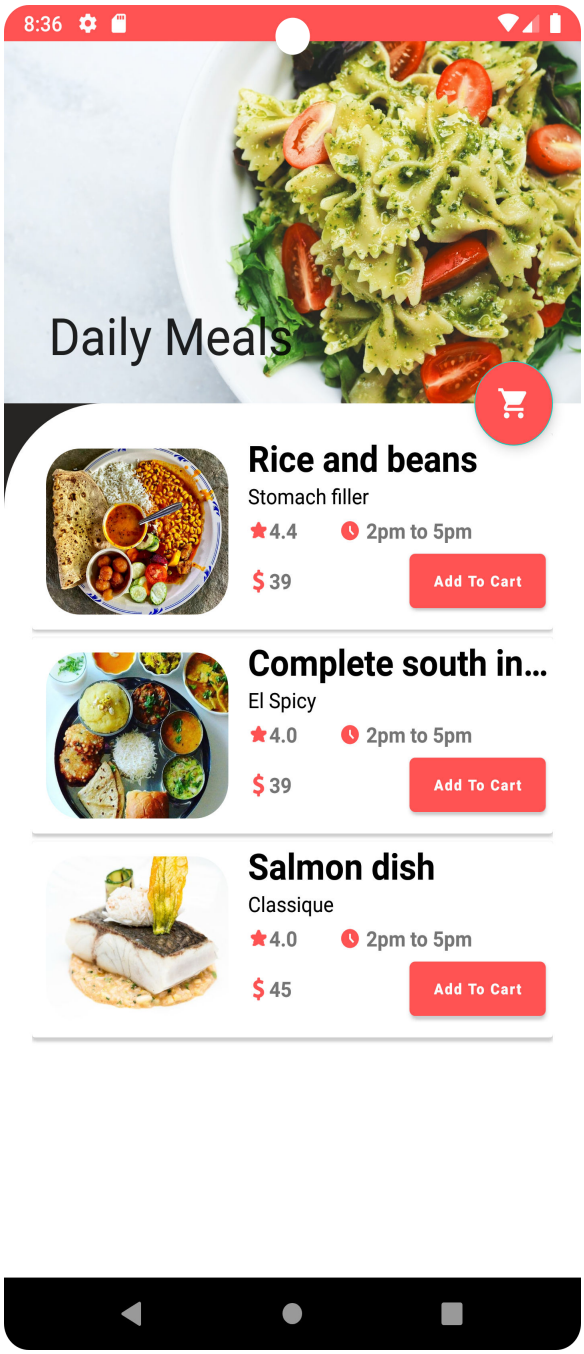

8:34
Home
Search your Food!
Hello User
What do you want ?
HamBurger
Fries
Ice Cream
Sandwich
Corn sandwich
4.0
10:00 - 12:00
Min - 15$
Brown bread sandwich
3.9
10:00 - 12:00
Min - 25$

- **SIDE DRAWER:**

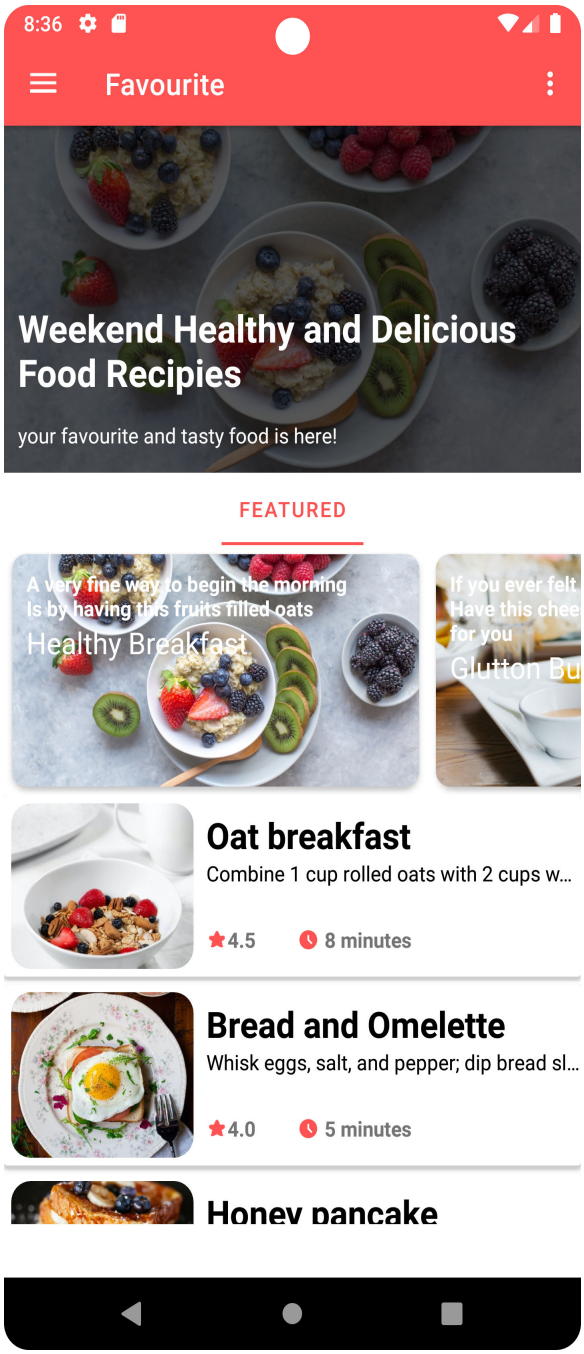

8:34
Hello there!
Have a nice day!
Home
Daily Meal
Featured Recipe
My Cart
Log Out
Ice Cream
4.9
Min - 35$
4.1
Min - 25$

- **DAILY MEALS PAGE:**

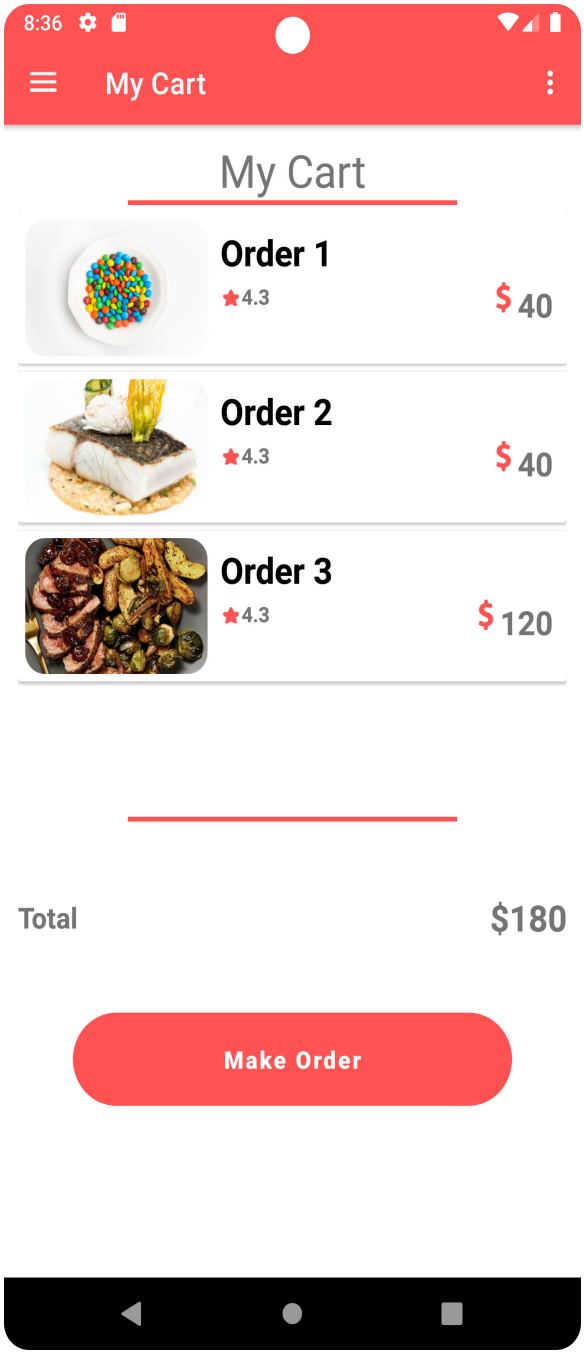




- **FEATURED PAGE:**



- **CART PAGE:**


8:36
My Cart
My Cart
Order 1
4.3
$ 40
Order 2
4.3
$ 40
Order 3
4.3
$ 120
Total
$180
Make Order

Chapter 5

CONCLUSION, FUTURE ENHANCEMENT AND REFERENCES

Conclusion

Android as a full, open and free mobile device platform, with its powerful function and good user experience rapidly developed into the most popular mobile operating system.

In conclusion, the development of a food ordering application presents a significant opportunity to revolutionize the food delivery industry by providing a convenient and efficient platform for users to order their favorite meals. The proposed application aims to address the current challenges faced by customers and restaurant owners, offering seamless ordering experiences and streamlined operations.

Future Enhancement

In future developments, several enhancements can further improve the food ordering application. Integration of artificial intelligence and machine learning algorithms can enhance personalized recommendations, ensuring more accurate suggestions based on user preferences and behaviors. Implementing advanced geolocation features can enable targeted promotions and improve delivery tracking accuracy. Integration with social media platforms can facilitate social sharing, user reviews, and recommendations, fostering engagement and community-building. Gamification elements, such as loyalty programs and rewards, can enhance user retention and encourage repeat orders. Additionally, incorporating voice recognition and natural language processing capabilities can enable voice-based ordering, making the application more accessible and user-friendly. By embracing these future enhancements, the food ordering application can continue to evolve, providing an even more exceptional and personalized experience for users and contributing to the growth and success of the food delivery industry.

References

- https://www.geeksforgeeks.org/
- https://www.javatpoint.com/
- https://stackoverflow.com/
- https://youtube.com/