# Project Design Phase-I
# Project Document Template

| | |
|---|---|
| Date | 8th October, 2023 |
| Team ID | Team-590932 |
| Project Name | **Snack Squad- A customisable snack ordering and delivery app.** |

| **Team member name:** | **Member details:** |
|---|---|
| Prasanna Gudivada | prasanna.21bce7104@vitapstudent.ac.in |
| Tishita Godavarthi | tishita.21bce7110@vitapstudent.ac.in |
| Varshitha Mattupalli | varshitha.21bce7327@vitapstudent.ac.in |

## INTRODUCTION

### Project Overview

Snack Squad is a sample project built using the Android Compose UI toolkit. It demonstrates how to create a simple e-commerce app for snacks using the Compose libraries. It is user friendly and customisable which enables the user to select the item/snack of their choice. This app is simple, easy to use and supports all devices and OS. While opening the app after installing the user can see the start/opening page where they would find a button named “Get started” and then by clicking on it user is redirected to the login page and then the home page where they can see the list of categories of items that are present and can select one from their preference and then ‘add their choice to the cart’. Similarly, they can find all the items they require and add them to the cart. And finally, they can see all the items that they wish to be in cart. The items that are not required can be deleted from the cart. The final order with the total amount can be seen at last before conforming the order. So, they can proceed to purchase.

### Purpose

The food delivery app industry is rapidly growing, with a projected market value of $230.8 billion by 2025. This growth is being driven by several factors, including the increasing popularity of smartphones, the rising demand for convenience, and the growing number of people living in urban areas.

## LITERATURE SURVEY

### Existing problem

There are a number of challenges facing the food delivery app industry, including:

- The need to attract and retain couriers.
- The need to ensure food safety and quality
- They need to provide seamless customer service
- Developing app that easy to use and navigate
- Partner with variety of restaurants to offer wide collection of food varieties to customers
- Development of delivery management system

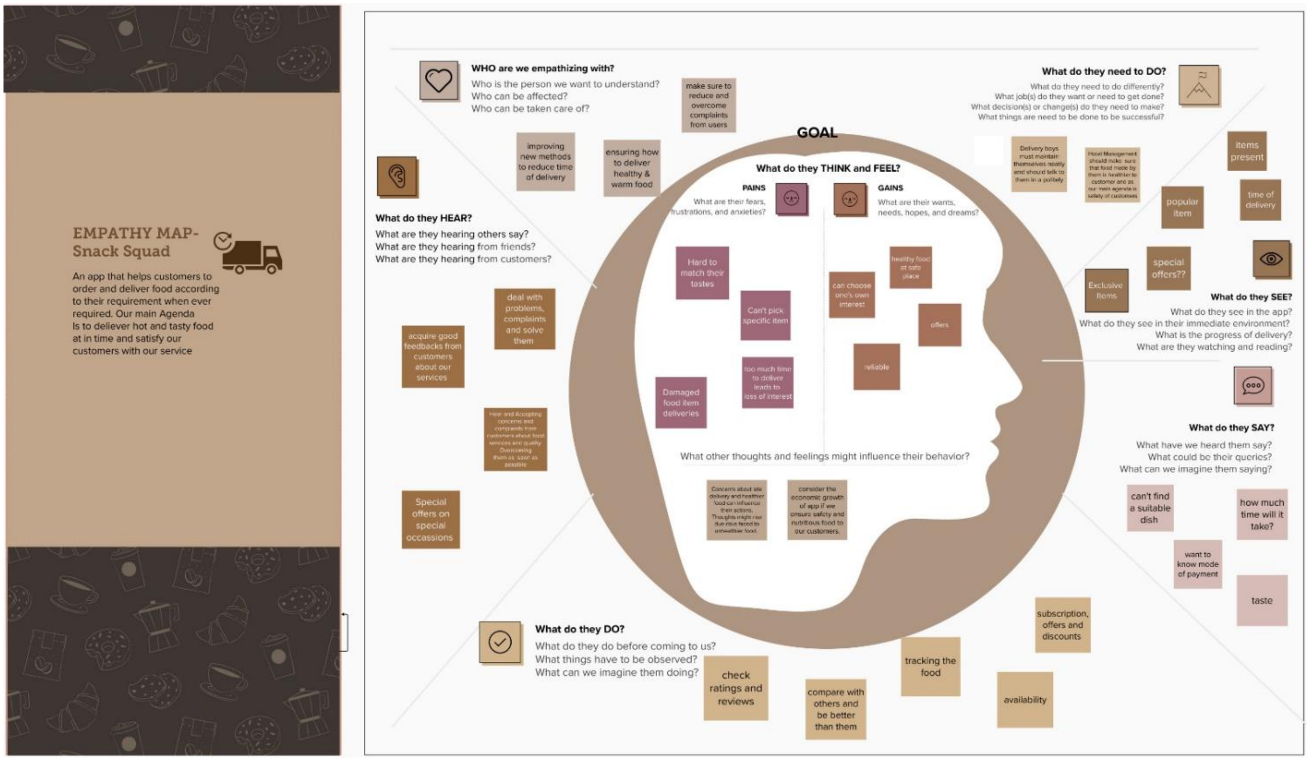
### 2.2 References

https://fareye.com/resources/blogs/on-demand-food-delivery-industry#:~:text=Improper%20food%20handling,and%20safety%20standards%20are%20maintained.

## 3. IDEATION & PROPOSED SOLUTION

### 3.1 Empathy Map Canvas

**Link:**
https://app.mural.co/t/snackapp4253/m/snackapp4253/1696919521115/f9d1a46ee800dfd0b8cf95cc51a7562e9025a69a?sender=udcc8868c2155880f88631938

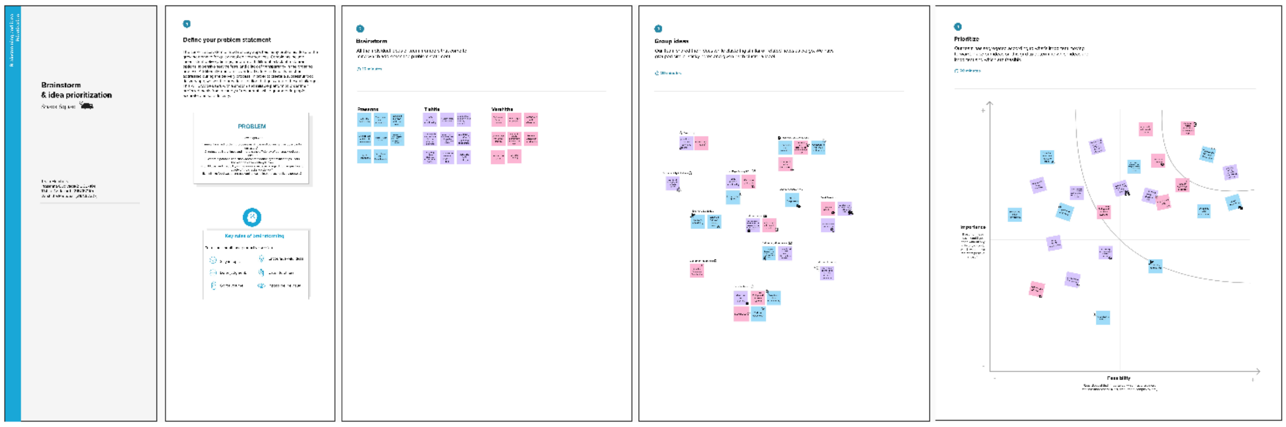


EMPATHY MAP-
Snack Squad
An app that helps customers to order and deliver food according to their requirement when ever required. Our main Agenda Is to deliever hot and tasty food at in time and satisfy our customers with our service
WHO are we empathizing with?
Who is the person we want to understand?
Who can be affected?
Who can be taken care of?
make sure to reduce and overcome complaints from users
improving new methods to reduce time of delivery
ensuring how to deliver healthy & warm food
GOAL
What do they need to DO?
What do they need to do differently?
What job(s) do they want or need to get done?
What decision(s) or change(s) do they need to make?
What things are need to be done to be successful?
Delivery boys must maintain themselves neatly and should talk to them in a politely
Hotel Management should make sure that food made by them is healthier to customer and as our main agenda is safety of customers
items present
popular item
time of delivery
What do they HEAR?
What are they hearing others say?
What are they hearing from friends?
What are they hearing from customers?
deal with problems, complaints and solve them
acquire good feedbacks from customers about our services
Hear and Accepting concerns and complaints from customers about food services and quality. Overcoming them as soon as possible
Special offers on special occassions
What do they THINK and FEEL?
PAINS
What are their fears, frustrations, and anxieties?
GAINS
What are their wants, needs, hopes, and dreams?
Hard to match their tastes
Can't pick specific item
too much time to deliver leads to loss of interest
Damaged food item deliveries
can choose one's own interest
healthy food at safe place
offers
reliable
What other thoughts and feelings might influence their behavior?
Concerns about late delivery and healthier food can influence their actions. Thoughts might rise due risks faced to unhealthier food.
consider the economic growth of app if we ensure safety and nutritious food to our customers.
Exclusive Items
special offers??
What do they SEE?
What do they see in the app?
What do they see in their immediate environment?
What is the progress of delivery?
What are they watching and reading?
What do they SAY?
What have we heard them say?
What could be their queries?
What can we imagine them saying?
can't find a suitable dish
how much time will it take?
want to know mode of payment
taste
What do they DO?
What do they do before coming to us?
What things have to be observed?
What can we imagine them doing?
check ratings and reviews
compare with others and be better than them
tracking the food
subscription, offers and discounts
availability

## 3.2 Ideation & Brainstorming

**Link:**

https://app.mural.co/t/snackapp4253/m/snackapp4253/1696936301111/4e1061d210165ab6cfada87f65017d8abf228b51?sender=udcc8868c2155880f88631938



Brainstorm & idea prioritization
Define your problem statement
PROBLEM
Brainstorm
Group ideas
Prioritize
Importance
Feasibility

## 4. REQUIREMENT ANALYSIS

### 4.1 Functional requirement

User Registration and Authentication

- Users can register for an account using their email address or social media credentials.
- Users should log in to their account using their registered email address and password.

Restaurant Search and Discovery

- Always search for restaurants by name, cuisine type, location, and other relevant criteria.
- The customers must be able to view restaurant details, including menus, ratings, reviews, and delivery times.

Menu Browsing and Order Placement

- Users can view restaurant menus, including item descriptions, prices, and availability.
- Users could customize their orders by selecting items, quantities, and any additional options.
- App users can add items to their cart and review their order before placing it.

Payment Processing and Order Completion

- Users can pay for their orders using a variety of payment methods, including credit cards, debit cards, and online payment gateways.
- Users receive a confirmation email or notification upon successful completion of their order.
- Users can rate and review their orders and restaurants.

### 4.2 Non-Functional requirements

Performance

- The app can handle a large volume of users and orders without experiencing performance issues.
- The app is able to load restaurant menus and order information quickly.

Availability

- The app will be available 24/7.
- The app should able to handle high traffic during peak hours.

Usability

- The app can be easy to use and navigate.

Reliability

- The app is regularly updated to fix bugs and improve performance.

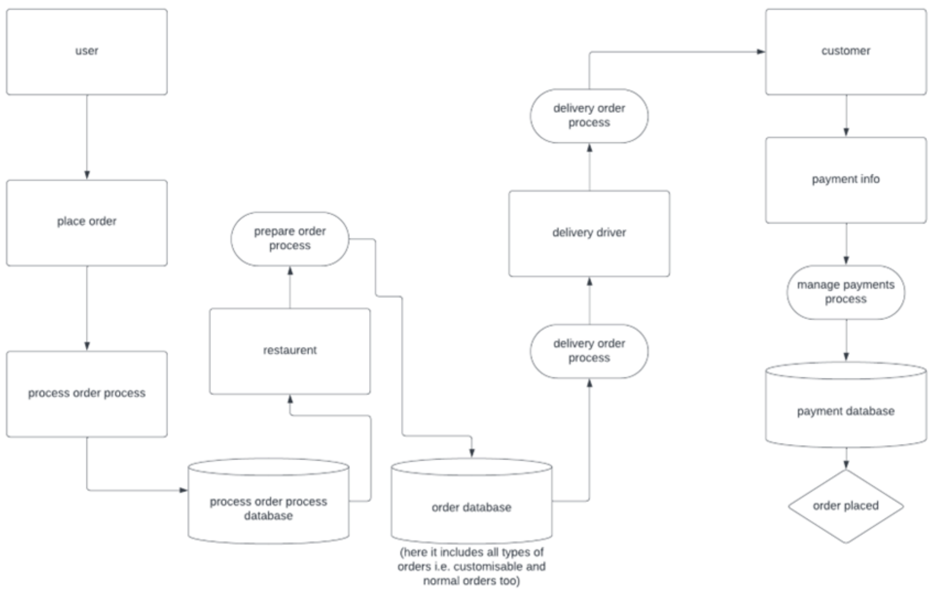
Maintainability

- Snack squad is well-documented and easy to understand.
- The app is modular and easy to maintain.

## 5. PROJECT DESIGN.

### 5.1 Data Flow Diagrams & User Stories

#### Dataflow diagram



## User Stories

| User Type | Functional Requirement (Epic) | User Story Number | User Story / Task | Acceptance criteria | Priority |
|---|---|---|---|---|---|
| Customer | Registration | USN-1 | As a user, I can register for the application by entering my email, password, and confirming my password. | I can access my account / dashboard | High |
| | | USN-2 | As a user, I will receive confirmation email once I have registered for the application | I can receive confirmation email & click confirm | High |
| | | USN-3 | As a user, I can register for the application through Facebook | I can register & access the dashboard with Facebook Login | Low |
| | | USN-4 | As a user, I can register for the application through Gmail | | Medium |
| | Login | USN-5 | As a user, I can log into the application by entering email & password | | High |
| | | | | | |
| Delivery Driver | Driver reg. | Driver-1 | As a new driver, I want to register with my personal and vehicle information. | I am accessed with my account and customer order details. | High |
| | | Driver-2 | As a new driver, I want to undergo a | I am well confirmed with all the info | High |

|  |  |  |  |  |  |
|---|---|---|---|---|---|
|  |  |  | background check and provide necessary documents. | check and documentation. |  |
|  | Availability. | Driver-3 | As a driver, I want to set my availability for delivering orders. | I will be available when the order is received or delivered. | High |
|  |  | Driver-4 | As a driver, I want to receive notifications for new delivery requests. | I am aware with the new order notifications. | Medium |
|  | Pickup and delivery of order. | Driver-5 | As a driver, I want to view the order details and accept or decline delivery requests. | I am available to accept the order to any place and at any time. | Low |
|  |  | Driver-6 | As a driver, I want to navigate to the customer's location for order pickup. | I must be aware of the root map. | Low |
|  |  | Driver-7 | As a driver, I want to confirm the delivery and receive payment. | I am trustful to the customer. | High |
| Administrator | User and driver management | ADM-1 | As an admin, I want to manage user accounts and access. | I can manage all user database. | High |
|  |  | ADM-2 | As an admin, I want to onboard and verify new delivery drivers. | I have knowledge on all the delivery drivers. | Medium |
|  | Content management | ADM-3 | As an admin, I want to | I am fully aware of the food | Medium |

| | | | | | |
|---|---|---|---|---|---|
| | | | manage the snack catalogue, including adding, updating, and removing snacks. | cuisines and the people choice. | |
| | | ADM-4 | As an admin, I want to manage promo codes, discounts, and other marketing content. | I am capable of accounting and sales. | High |
| | Order management | ADM-5 | As an admin, I want to view and manage all orders, including order status updates. | I am available at any time. | High |
| | | ADM-6 | As an admin, I want to handle user inquiries and resolve issues. | I am good problem solver. | Medium |

## 5.2 Solution Architecture

**Architecture**:

There are several components involved in creating the architecture for a snack delivery application, which ensures that it operates smoothly and effectively. You will find a high-level overview of the architecture for snack apps:

- **Client-side Application:**

  Mobile App: The main interface for customers who want to see snacks and order them, or make payments.

  Web App (Optional): A web-based version for customers who prefer ordering snacks through a web browser.

- **Backend Server:**

Web server: handles incoming requests from clients' applications, controls user sessions and serves as an API gateway.

Application logic: Applies core business logic such as user authentication, snack catalogue management, order handling and payment processing.

- **Database:**

User's data: retains user names, personal information, and order history.

The Snack catalogue: contains information about available snacks, their descriptions, and prices as well as the level of stocks at present.

Order data: The information on existing and past orders is stored.

- **Authentication and Authorization:**

To make sure that features of an application can be accessed with confidence, implement authentication methods such as OAuth, JWT or API keys.

- **Order Management:**

Shopping cart: Keep the user's shopping carts, which allow users to add, modify and remove snacks.

Order processing: handling the placing, order confirmation and real-time status updates of orders.

Delivery Scheduling: Implement a feature for customers to select delivery time slots.

- **Delivery Management:**

Delivery driver assignment: assign to incoming orders drivers who are available.

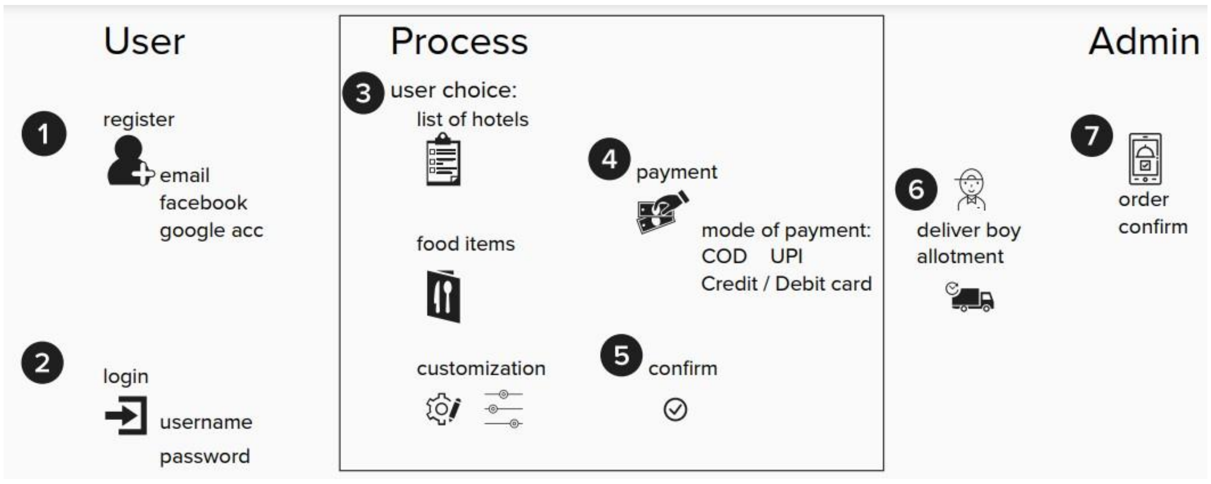
Realtime tracking: keep an eye on drivers' location and let customers know about it.

Optimisation of routes: optimise delivery routes for drivers in order to minimise transport times and costs.

The specific technologies and frameworks that are used will depend on team's expertise and the targeting platform (iOS, Android, web). Additionally, considering the user experience and design to make the app appealing and user-friendly. And continuously improving and updating app based on user feedback and changing market demands.

## 6. PROJECT PLANNING & SCHEDULING

### 6.1 Technical Architecture



Guidelines:

1. **Include All Processes (Application Logic / Technology Block**): Ensure that all essential processes within the application are defined and accounted for, including user interfaces, business logic, and any additional technology blocks. This should cover all user interactions, application functionality, and backend processes.
2. **Provide Infrastructural Demarcation (Local / Cloud):** Clearly specify the infrastructure demarcation, including whether components or services are deployed locally for development and testing or in the cloud for production. Outline the server configurations for each environment.
3. **Indicate External Interfaces (Third-Party APIs, etc.):** Identify and document all external interfaces, such as third-party APIs, that the Snack Squad app interacts with. Describe the purpose and usage of each external interface and ensure that their integration is well-documented.

4. **Indicate Data Storage Components/Services:** Define the data storage components and services used for different types of data. Specify whether you are using databases (e.g., MySQL or NoSQL), cloud-based data storage solutions (e.g., AWS DynamoDB), or other file storage mechanisms (e.g., IBM Block Storage).
5. **Indicate Interface to Machine Learning Models (If Applicable**): If the Snack Squad app utilizes machine learning models, clearly outline the interface between the application and these models. Describe the purpose and functionality of these models, as well as the technology and tools used for integration. By following these guidelines, you can ensure that the Snack Squad app project is well-documented and organized, with clear demarcation of processes, infrastructure, external interfaces, data storage components, and machine learning integration, ultimately leading to a well-structured and efficiently managed project.

**Table-1** : Components & Technologies:

| S.No | Component | Description | Technology |
|---|---|---|---|
| 1. | User Interface | web<br>UI<br>mobile<br>app<br>chat<br>bot | HTML, CSS, JavaScript<br>React native.<br>dialogue flow for chatbot interaction |
| 2. | Application Logic-1 | business<br>logic server<br>logic | kotlin for Android, swift for iOS node.js for web backend |
| 3. | Application Logic-2 | voice recognition | IBM Watson speech to text service |
| 4. | Application Logic-3 | chatbot interaction | IBM Watson Assistant |
| 5. | Database | user<br>profiles<br>snack<br>inventory | MySQL<br>NoSQL |
| 6. | Cloud Database | cloud based storage | AWS DynamoDB |

|  |  |  |  |
|---|---|---|---|
| 7. | File Storage | user content<br>application files | Amazon S3 for cloud storage local file system and cloud storage |
| 8. | External API-1 | weather data | IBM Weather API, etc. |
| 9. | External API-2 | user verification | Aadhar API local file system and cloud storage for secure user identification |
| 10. | Machine Learning Model | snack recommendation | custom machine learning model for personalised recommendations. |
| 11. | Infrastructure (Server / Cloud) | application<br>deployment local<br>server<br>configuration<br>cloud server configuration cloud deployment orchestration | AWS cloud for scalability<br>N/A (development and testing)<br>AWS EC 2 instances for production<br>KUBERNETES for containerization and scaling |

**Table-2:** Application Characteristics:

| S. No | Characteristics | Description | Technology |
|---|---|---|---|
| 1. | Open-Source Frameworks | List the open-source frameworks used | react for the user interface (web and mobile app) node JS for server logic. mongo DB as an NoSQL database |
| 2. | Security Implementations | List all the security / access controls implemented, use of firewalls etc. | e.g., SHA-256, Encryptions, IAM Controls, OWASP etc. |
| 3. | Scalable Architecture | Justify the scalability of architecture (3 – tier, Microservices) | micro services architecture for components scalability Uber needs for container orchestration and auto scaling |
| 4. | Availability | Justify the availability of application (e.g., use of load balancers, distributed servers etc.) | Use of load balancers for even distribution of traffic<br>AWS elastic load balancing for high availability |
| 5. | Performance | Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN's) etc. | Catching mechanisms to reduce database queries.<br>content delivery networks for faster content delivery And reduce latency. |

## 6.2 Sprint Planning & Estimation:

| Sprint | Functional Requirement (Epic) | User Story Number | User Story / Task | Story Points | Priority |
|---|---|---|---|---|---|
| SPRINT 1 | USER AUTHENTICATION | USOO1<br>US002 | As a user, I want to create an account to start using the app.<br>As a user I want to login with my credentials securely. | 3<br>2 | HIGH |
| SPRINT 1 | SNACK CATALOG | US003<br>US004 | As a user I want to view a list of available snacks.<br>As a user I want to filter snacks by category. | 5<br>3 | HIGH<br>MEDIUM |
| SPRINT 2 | SNACK CUSTOMIZATION | US005<br>US006 | as a user I want to create a customised snack box. as a user I want to add and remove snacks from my box or cart. | 8<br>5 | HIGH |
| SPRINT 2 | RECOMMENDATION ENGINE | US007<br>US008 | as a user I want to receive personalised snack suggestions.<br>As a user I want to see the reasoning behind the recommendations. | 8<br>3 | HIGH<br>MEDIUM |
| SPRINT 3 | CART AND CHECKOUT | US009<br>US010 | as a user I want to review and confirm my snack box contents. as a user I want to proceed to payment and complete my purchase. | 5<br>5 | HIGH |

| Sprint | Total Story Points | Duration | Sprint Start Date | Sprint End Date (Planned) | Story Points Completed (as on Planned End Date) | Sprint Release Date (Actual) |
|---|---|---|---|---|---|---|
| Sprint-1 | 13 | 7 Days | 20 Oct 2023 | 26 Oct 2023 | 13 | 26 Oct 2023 |
| Sprint-2 | 16 | 7 Days | 27 Oct 2023 | 02 Nov 2023 | 16 | 02 Nov 2023 |
| Sprint-3 | 15 | 7 Days | 03 Nov 2023 | 09 Nov 2023 | 15 | 09 Nov 2023 |

## 6.3 Sprint Delivery Schedule

For Sprint-3: AV3 = 7 days / 15 story points = 0.4667 days/story point

Now, to find the overall AV, we take the average of the individual sprint average velocities:

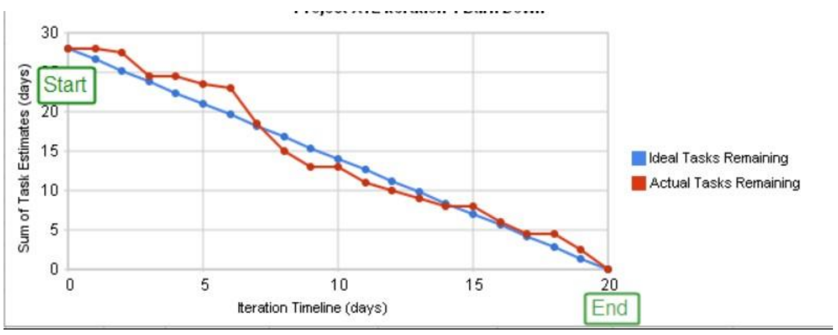
AV = (AV1 + AV2 + AV3) / 3

AV = (0.5385 + 0.4375 + 0.4667) / 3

AV = 0.4809 days/story point (approximately)

So, the overall Average Velocity for the project, based on the updated data, is approximately 0.4809 days/story point. This represents the average time it takes to complete one story point across all three sprints in the Snack Squad app project.

Graph:



## 7. CODING & SOLUTIONING (Explain the features added in the project along with code)

Project Workflow:

- User initially register into Snack Squad App.
- After Registration, login into Snack Squad
- User is directed to main page

- User could see list of items, select and order according to their requirement
- They could customize them according to their taste and requirement.

**Tasks:**

1.Required initial steps

2.Creating a new project.

3.Adding required dependencies.

4.Creating the database classes.

5.Building application UI and connecting to database.

6.Using AndroidManifest.xml

7.Running the application.

**Task 1:**

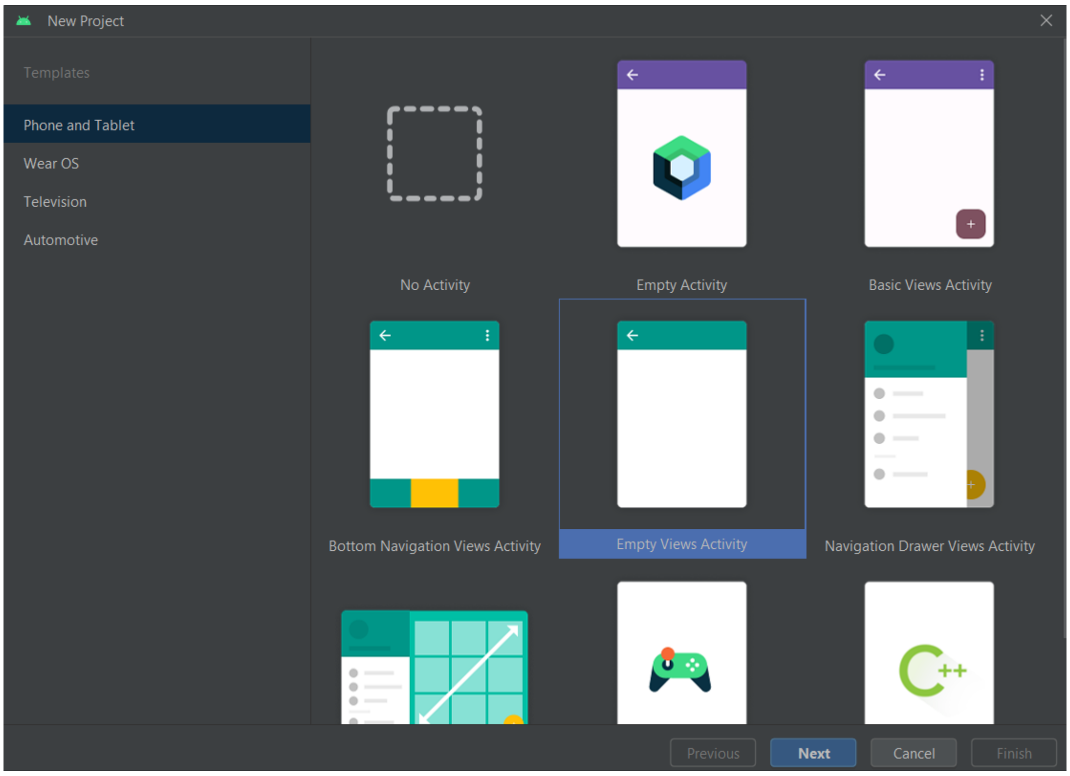
Required initial steps : https://developer.android.com/studio/install

**Task 2:**

Creating a new project.

Step 1: Android studio > File > New > New Project > Empty Views Activity

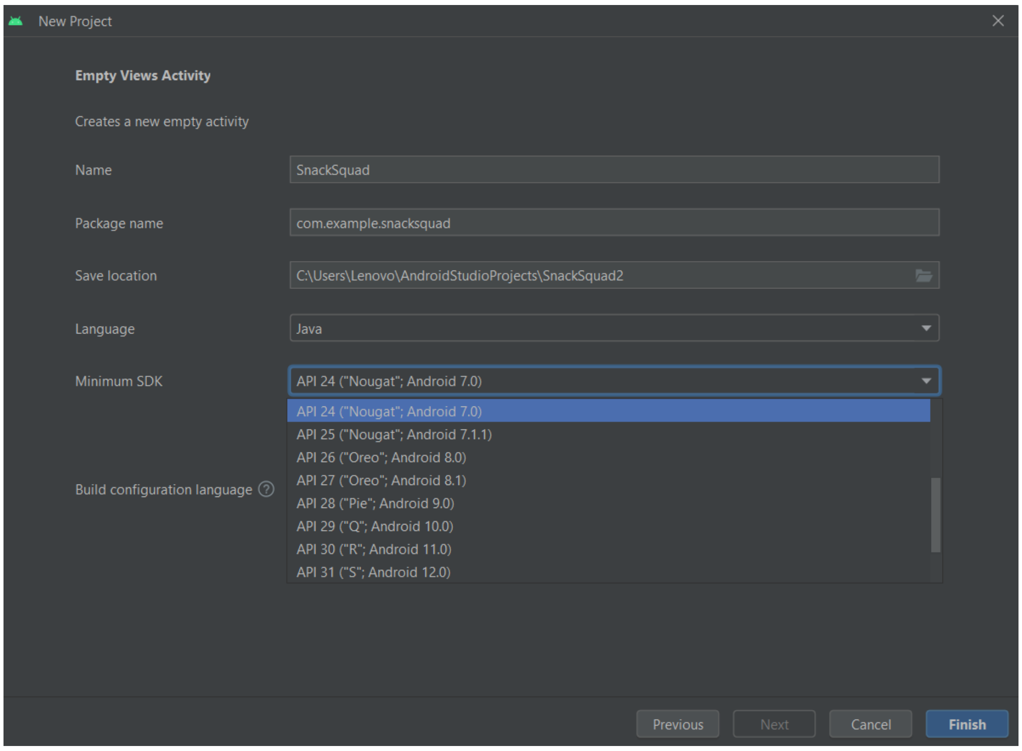
Step 2: Click on Next button.



Step 3: Give name to the new project

Step 4: Give the Minimum SDK value

Step 5: Click Finish



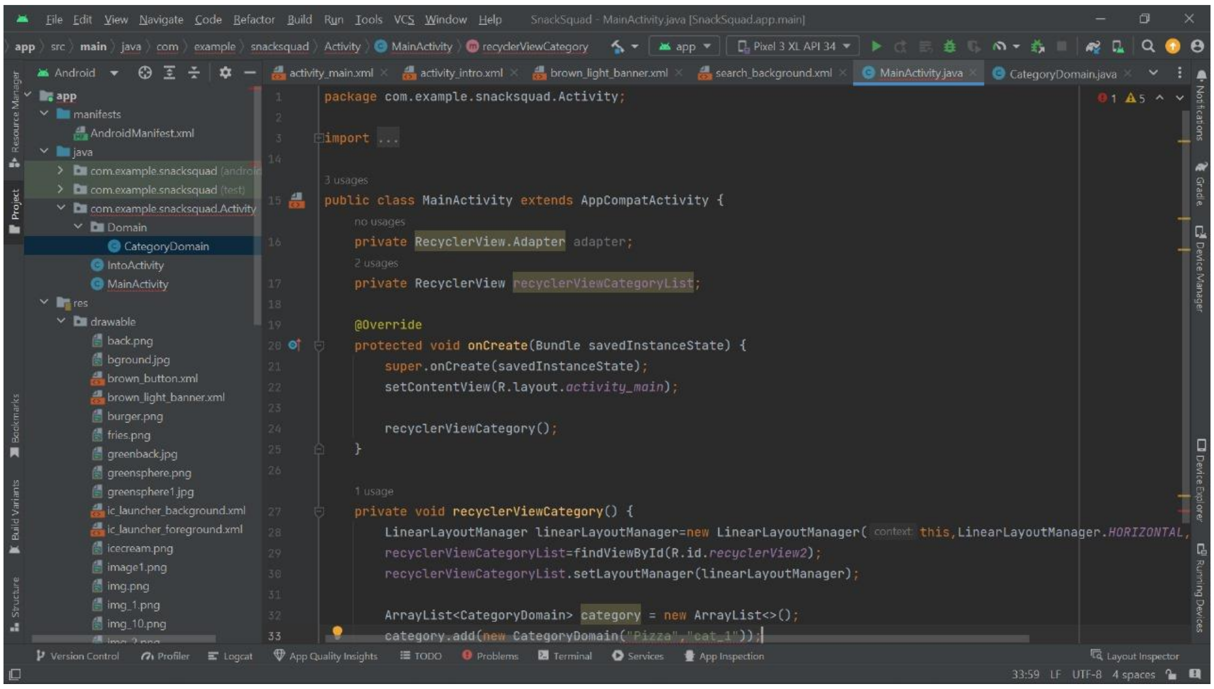
**Empty Views Activity**

Creates a new empty activity

| | |
|---|---|
| Name | SnackSquad |
| Package name | com.example.snacksquad |
| Save location | C:\Users\Lenovo\AndroidStudioProjects\SnackSquad2 |
| Language | Java |
| Minimum SDK | API 24 ("Nougat"; Android 7.0) |
| Build configuration language | |

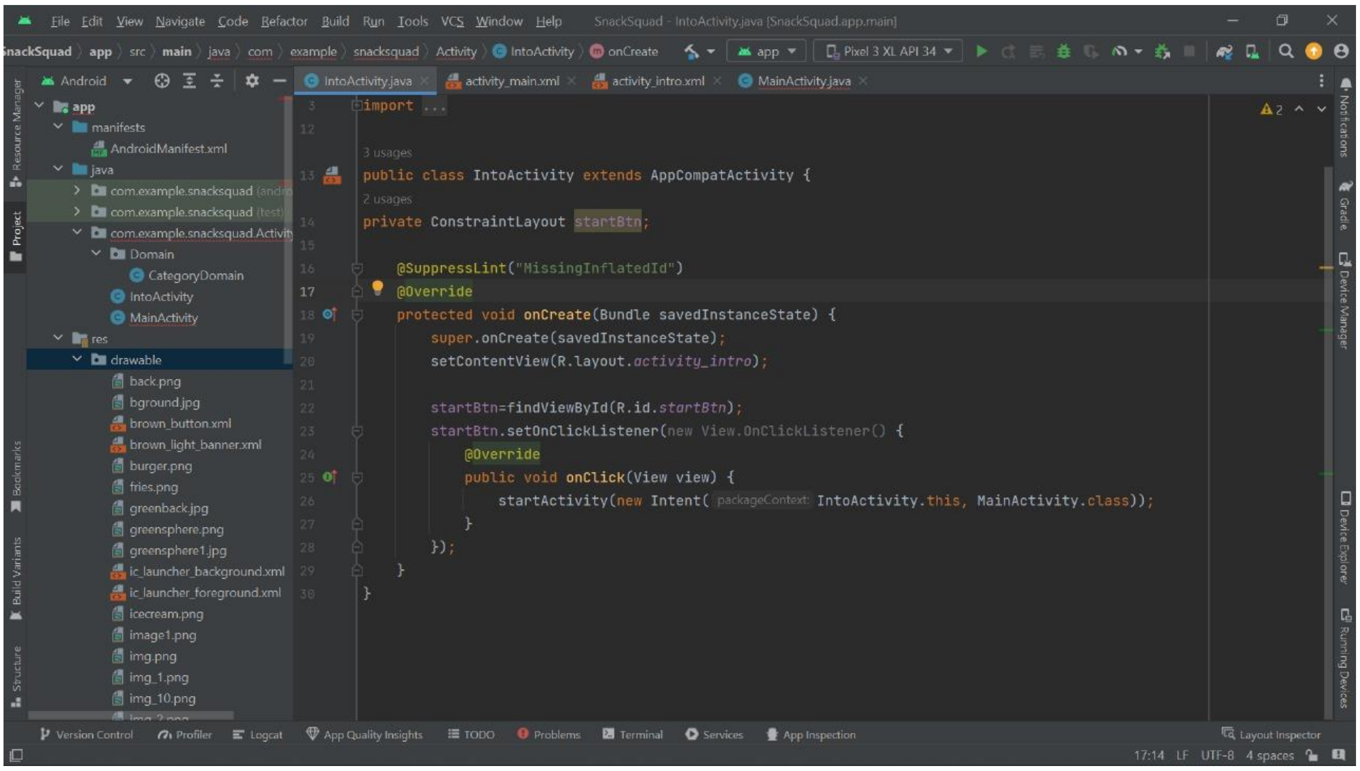
API 24 ("Nougat"; Android 7.0)
API 25 ("Nougat"; Android 7.1.1)
API 26 ("Oreo"; Android 8.0)
API 27 ("Oreo"; Android 8.1)
API 28 ("Pie"; Android 9.0)
API 29 ("Q"; Android 10.0)
API 30 ("R"; Android 11.0)
API 31 ("S"; Android 12.0)

Main activity file



```java
package com.example.snacksquad.Activity;

import ...

3 usages
public class MainActivity extends AppCompatActivity {
    no usages
    private RecyclerView.Adapter adapter;
    2 usages
    private RecyclerView recyclerViewCategoryList;

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

        recyclerViewCategory();
    }

    1 usage
    private void recyclerViewCategory() {
        LinearLayoutManager linearLayoutManager=new LinearLayoutManager( context: this,LinearLayoutManager.HORIZONTAL,
        recyclerViewCategoryList=findViewById(R.id.recyclerView2);
        recyclerViewCategoryList.setLayoutManager(linearLayoutManager);

        ArrayList<CategoryDomain> category = new ArrayList<>();
        category.add(new CategoryDomain("Pizza","cat_1"));
```

IntroActivity File

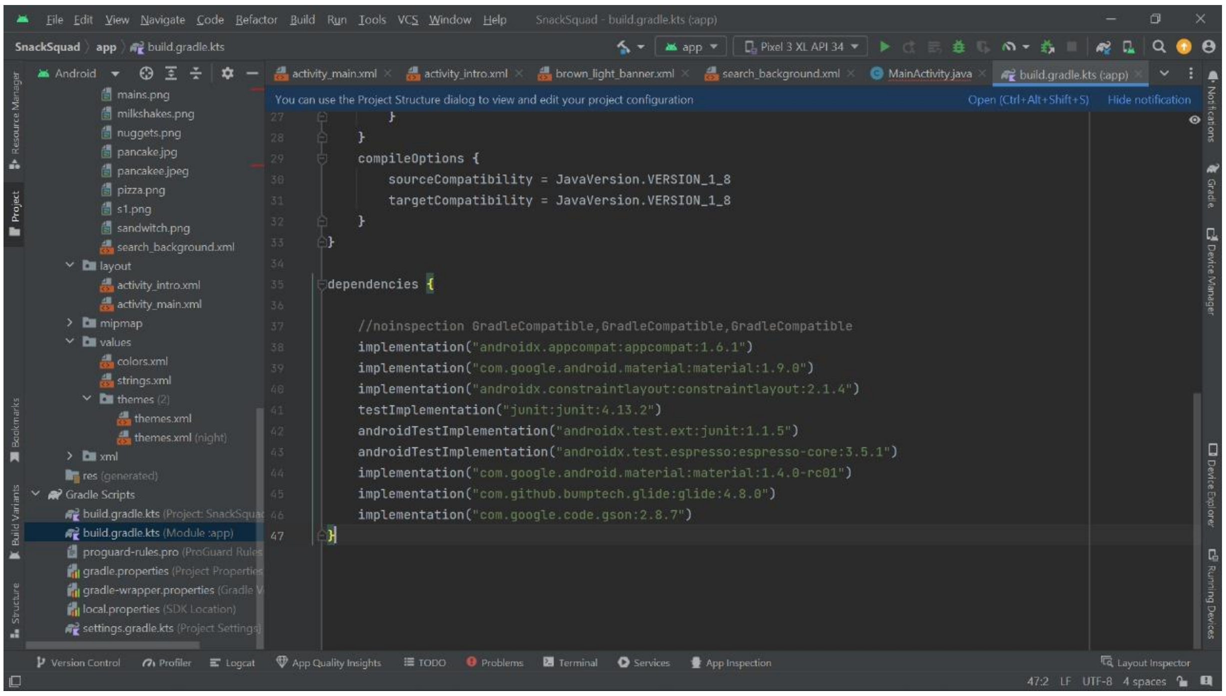


```
import ...

3 usages
public class IntoActivity extends AppCompatActivity {
2 usages
private ConstraintLayout startBtn;

    @SuppressLint("MissingInflatedId")
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_intro);

        startBtn=findViewById(R.id.startBtn);
        startBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startActivity(new Intent( packageContext: IntoActivity.this, MainActivity.class));
            }
        });
    }
}
```

Task 3 :

Adding required dependencies.

Step 1 : Gradle scripts > build.gradle(Module :app)



You can use the Project Structure dialog to view and edit your project configuration

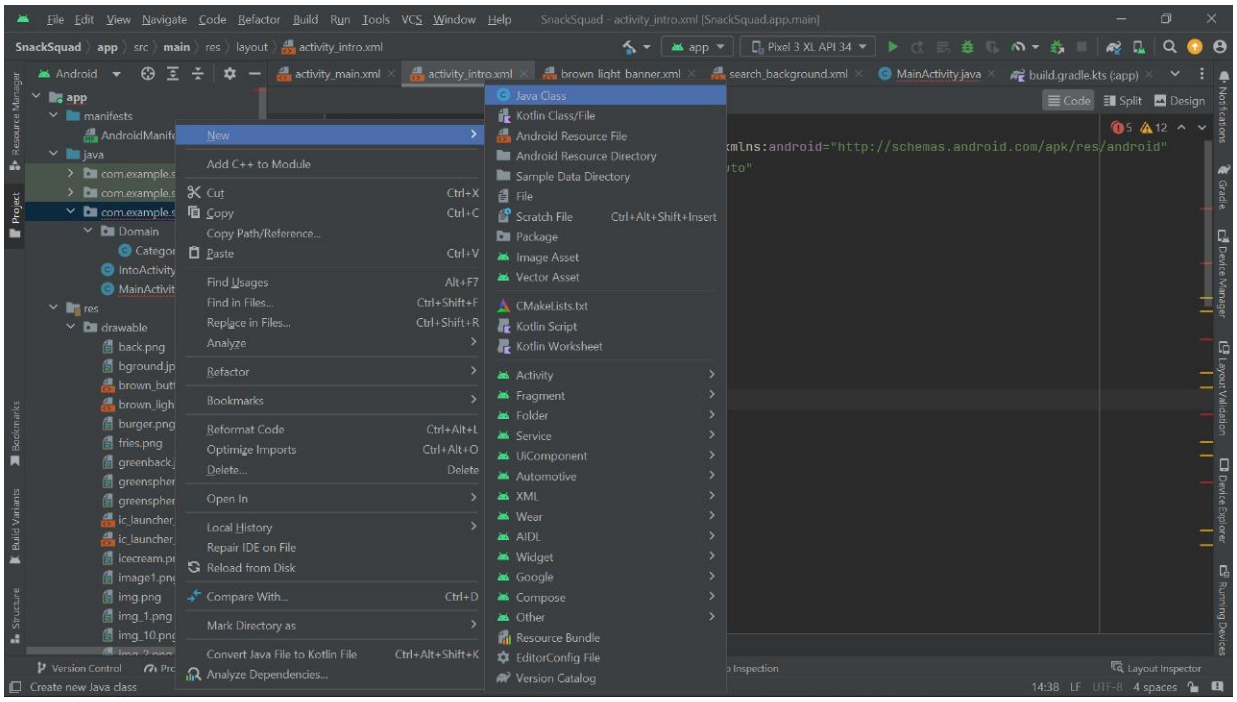
```
        }
    }
    compileOptions {
        sourceCompatibility = JavaVersion.VERSION_1_8
        targetCompatibility = JavaVersion.VERSION_1_8
    }
}

dependencies {

    //noinspection GradleCompatible,GradleCompatible,GradleCompatible
    implementation("androidx.appcompat:appcompat:1.6.1")
    implementation("com.google.android.material:material:1.9.0")
    implementation("androidx.constraintlayout:constraintlayout:2.1.4")
    testImplementation("junit:junit:4.13.2")
    androidTestImplementation("androidx.test.ext:junit:1.1.5")
    androidTestImplementation("androidx.test.espresso:espresso-core:3.5.1")
    implementation("com.google.android.material:material:1.4.0-rc01")
    implementation("com.github.bumptech.glide:glide:4.8.0")
    implementation("com.google.code.gson:gson:2.8.7")
}
```

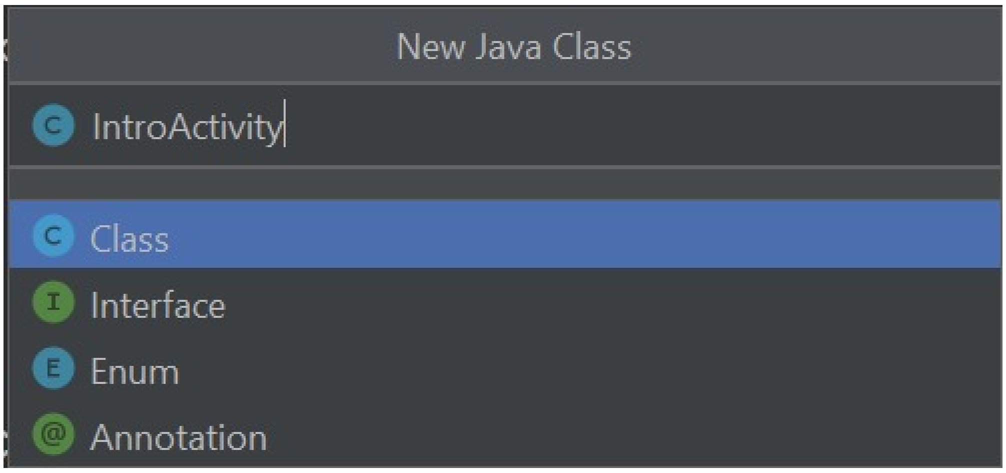
Step 2 : Click on Sync now

Task 4:

User View

Step 1 : Create IntroActivity class




New Java Class

IntroActivity

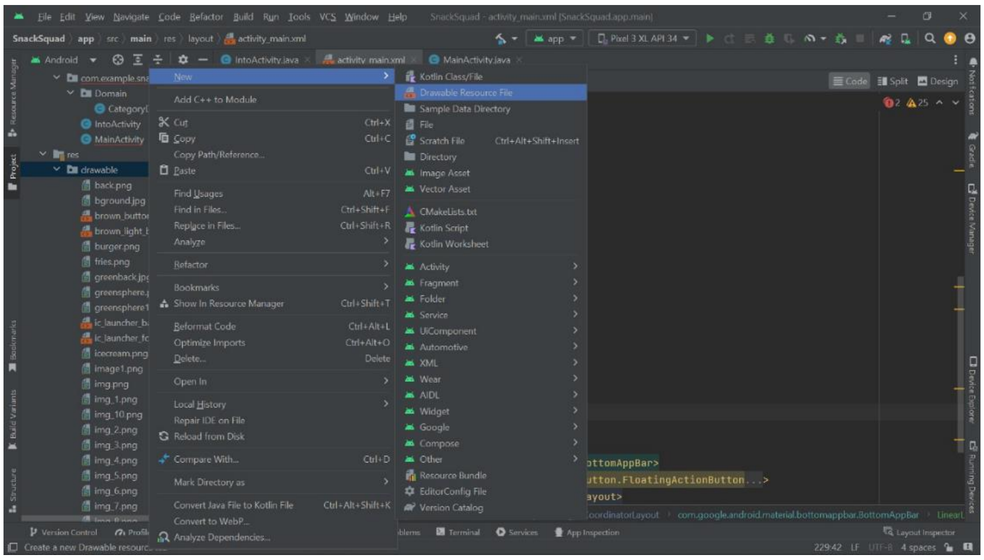
Class

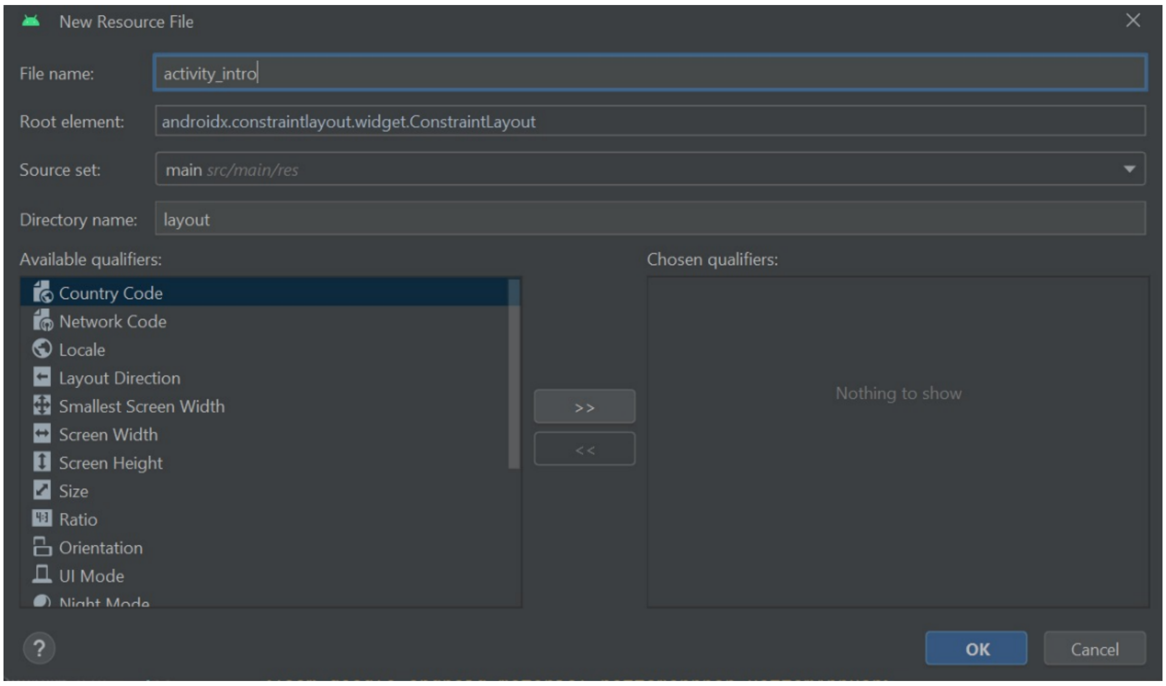
Interface

Enum

Annotation

Step2:Creating layout activity_intro (Similar to activity_main)



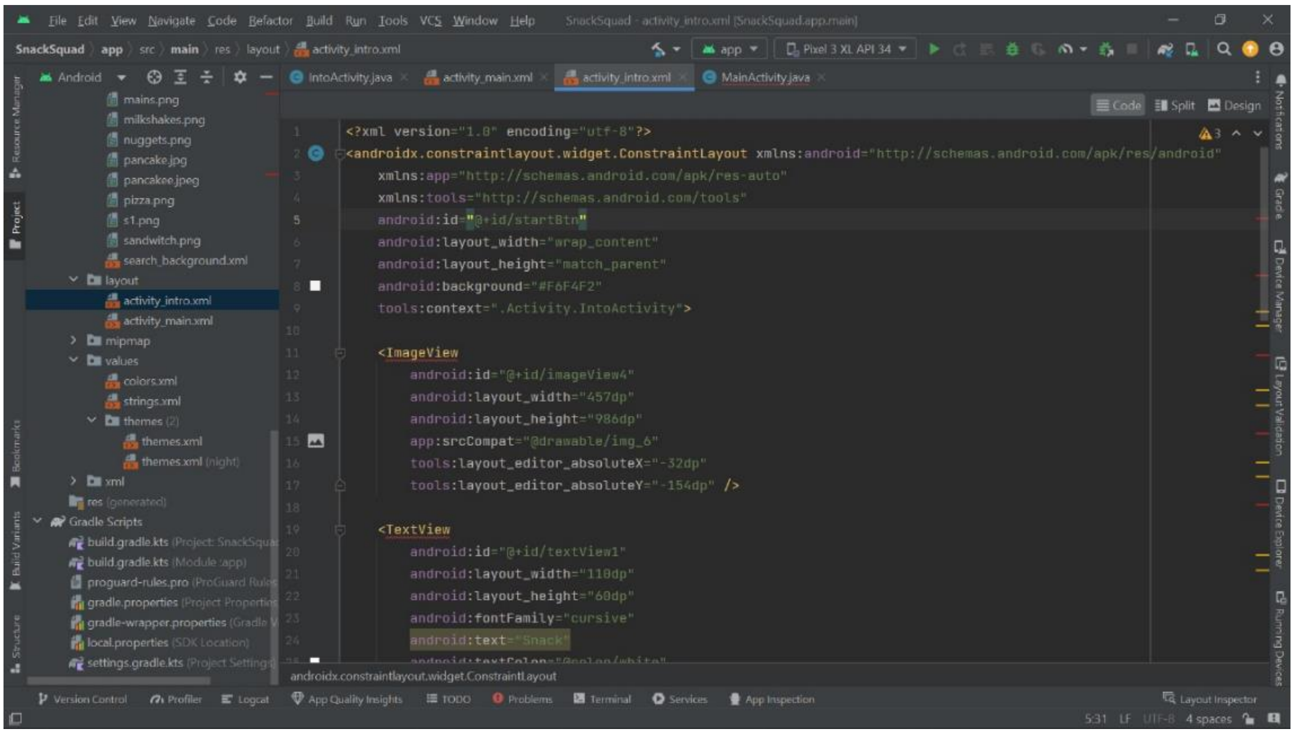

New Resource File

File name: activity_intro

Root element: androidx.constraintlayout.widget.ConstraintLayout

Source set: main src/main/res

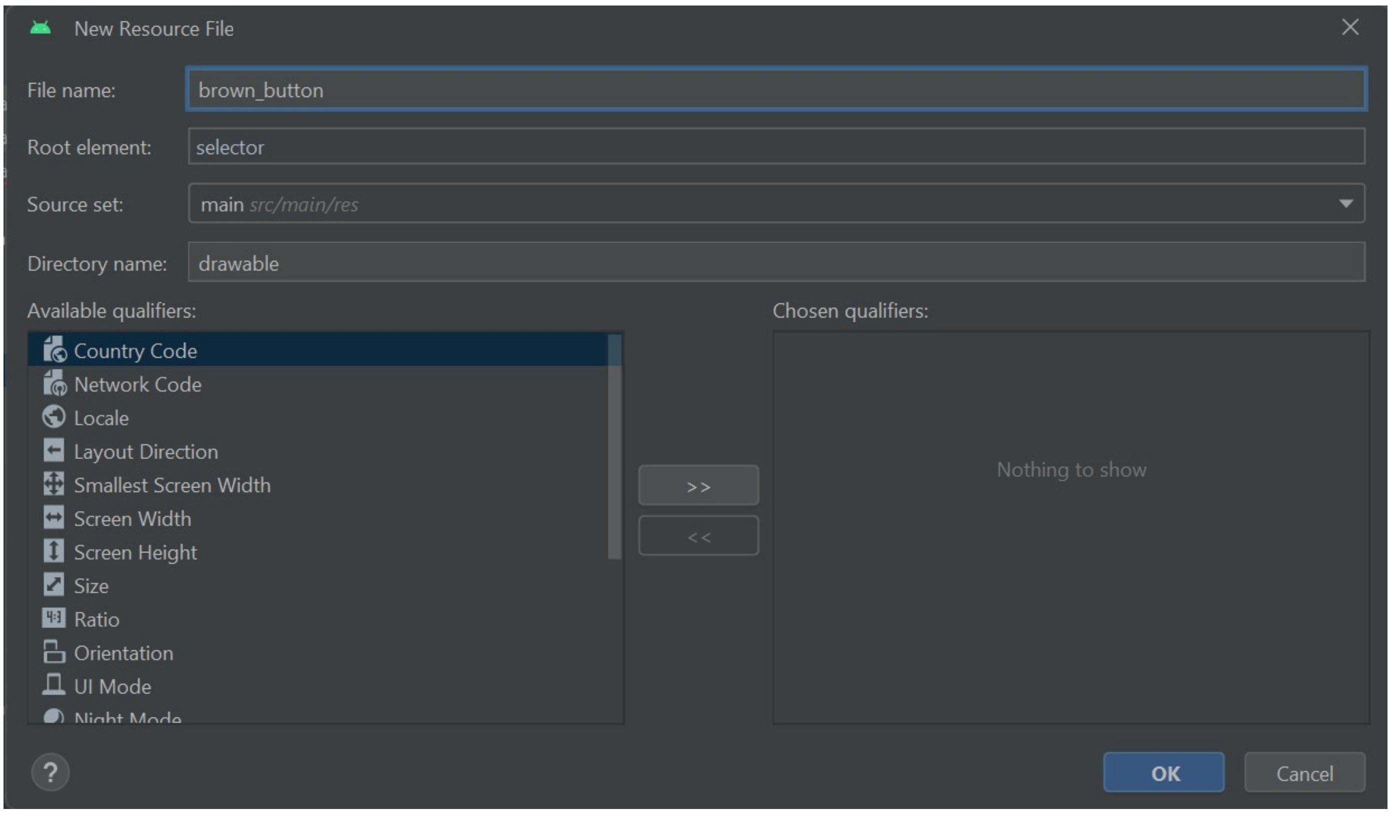
Directory name: layout



```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:id="@+id/startBtn"
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:background="#F6F4F2"
    tools:context=".Activity.IntoActivity">

    <ImageView
        android:id="@+id/imageView4"
        android:layout_width="457dp"
        android:layout_height="986dp"
        app:srcCompat="@drawable/img_6"
        tools:layout_editor_absoluteX="-32dp"
        tools:layout_editor_absoluteY="-154dp" />

    <TextView
        android:id="@+id/textView1"
        android:layout_width="110dp"
        android:layout_height="60dp"
        android:fontFamily="cursive"
        android:text="Snack"
```

Step3:

Creating button folder for the starting Page



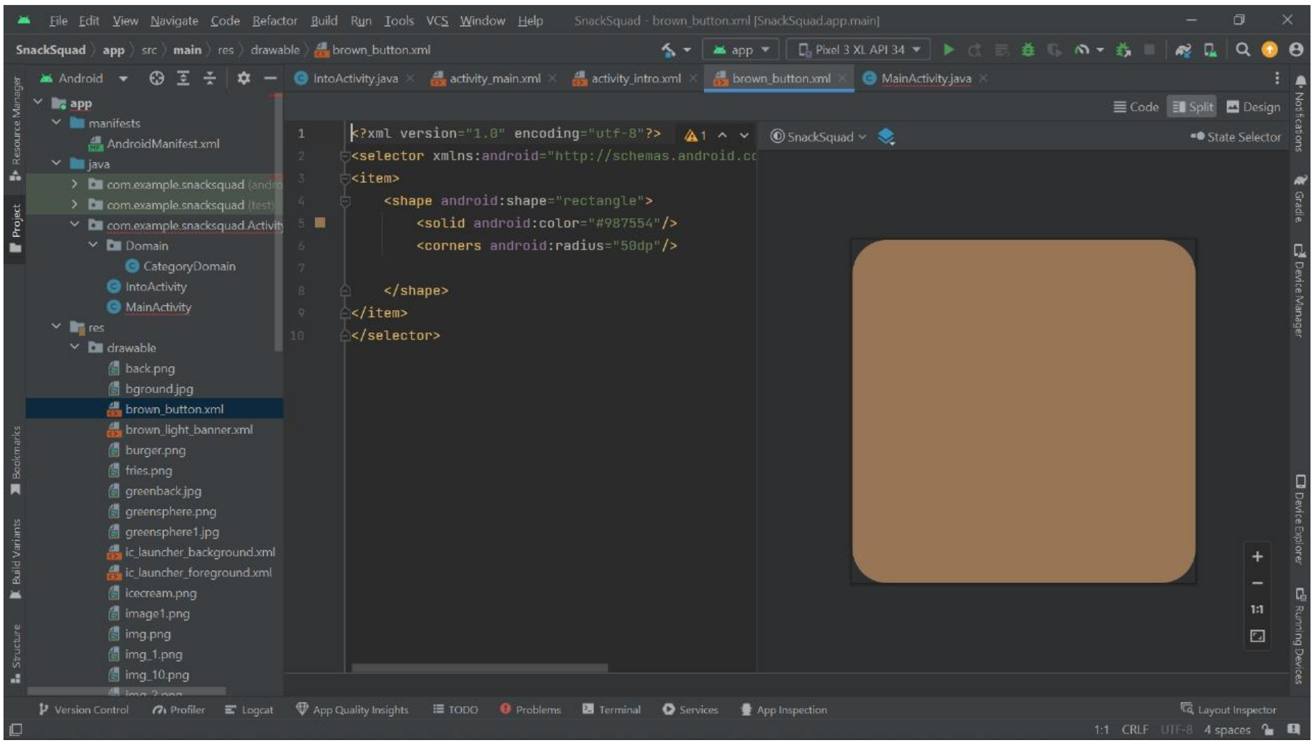
New Resource File

File name: brown_button

Root element: selector

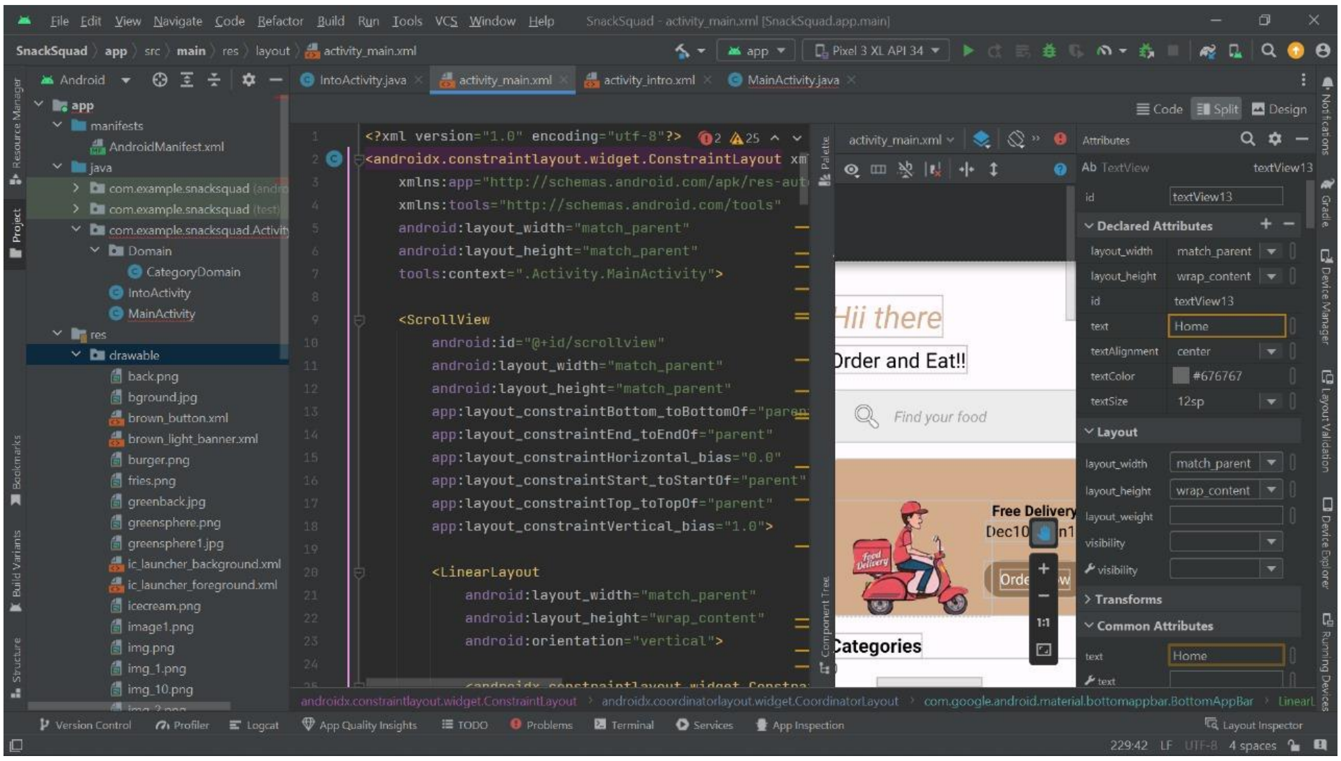
Source set: main src/main/res

Directory name: drawable



```
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.cc
<item>
    <shape android:shape="rectangle">
        <solid android:color="#987554"/>
        <corners android:radius="50dp"/>

    </shape>
</item>
</selector>
```

Activity_main:



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

    <ScrollView
        android:id="@+id/scrollview"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        app:layout_constraintBottom_toBottomOf="paren
        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="1.0">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical">
```

```
Declared Attributes
layout_width    match_parent
layout_height   wrap_content
id              textView13
text            Home
textAlignment   center
textColor       #676767
textSize        12sp
```

## 8. PERFORMANCE TESTING

### 8.1 Performance Metrics

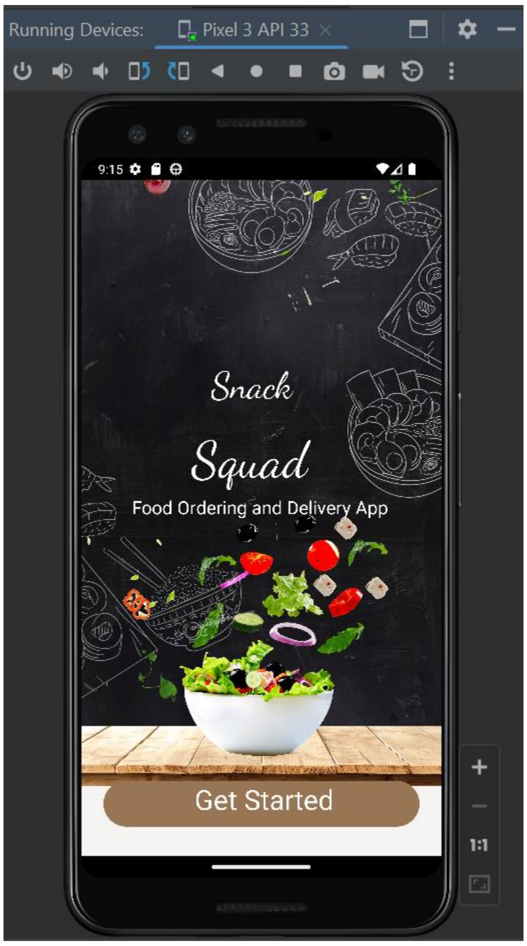
Response time: 1.5 seconds on average.

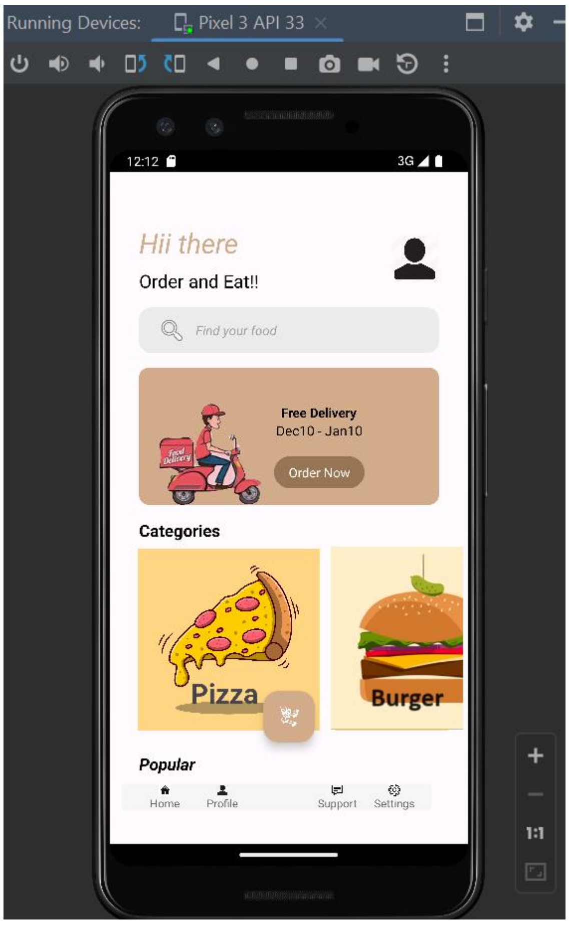
Throughput: 100 transactions per minute.

## 9. RESULTS

### 9.1 Output Screenshots:

Starting Page : (activity_intro) activity_main:





## 10. ADVANTAGES & DISADVANTAGES

Advantages of food delivery app

- Convenience: Food delivery apps make it easy and convenient to order food from restaurants without having to leave your home or office.
- Variety: Food delivery apps offer a wide variety of restaurants and cuisines to choose from.
- Choice: Food delivery apps allow you to customize your order and choose exactly what you want to eat.
- Efficiency: Food delivery apps can help you save time and hassle by taking care of the entire ordering and delivery process.
- Accessibility: Food delivery apps can make it easier for people with disabilities or mobility issues to enjoy restaurant food.

Disadvantages of food delivery apps

- Cost: Food delivery apps typically charge a commission on each order, which can add to the cost of your meal.
- Food quality: The quality of food delivered by food apps can vary depending on the restaurant and the courier.
- Delivery time: Delivery times can vary depending on the restaurant, the distance to your delivery address, and traffic conditions.
- Unreliable service: Food delivery apps can sometimes experience technical difficulties or delays, which can lead to a frustrating customer experience.
- Driver safety: Food delivery drivers are often at risk of traffic accidents and other hazards.

Overall, food delivery apps offer a number of advantages, such as convenience, variety, and choice. However, there are also some disadvantages to consider, such as cost, food quality, delivery time, unreliable service, and driver safety.

## 11. CONCLUSION

We conclude that our snack delivery supply healthy and nutritious food to our customers and never fail impress them. Our main aim is to deliver them within the time safely without any inconvenience. Its our responsibility as a citizen of our country to save environment, hence we use ecofriendly packaging to protect nature and can also reduce pollution instead of using plastic bags and containers which are unhealthy when the food is warm. Snack squad app is user-friendly so that customers can customize their own food according to their requirements, which makes it easy to use wherever and whenever.

## 12. FUTURE SCOPE

The future scope of food delivery apps is promising, with several potential developments and advancements on the horizon:

1. Drone Delivery:
    - Integration of drone technology for faster and more efficient food deliveries, especially in urban areas.
2. Artificial Intelligence (AI) and Machine Learning:
    - Advanced AI algorithms for personalized recommendations based on user preferences and order history.
3. Voice-Activated Ordering:

- Integration with virtual assistants for hands-free and voice-activated food ordering.

4. Health and Wellness Focus:
    - Integration of health-centric features, such as calorie tracking, nutritional information, and options for special dietary requirements.

5.Sustainability Initiatives:

- Emphasis on eco-friendly packaging and sustainable practices to reduce the environmental impact of food delivery.

Food delivery apps' future will almost certainly involve a convergence of technology, sustainability, and user-centric features to create a more seamless and enjoyable experience for customers. The possibilities for innovation in the food delivery industry are endless as technology advances.

## 13. APPENDIX

**Source code:**

IntroActivity.java

```java
package com.example.snacksquad.Activity;

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

import android.annotation.SuppressLint;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;

import com.example.snacksquad.R;

public class IntoActivity extends AppCompatActivity {
private ConstraintLayout startBtn;

    @SuppressLint("MissingInflatedId")
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_intro);

        startBtn=findViewById(R.id.startBtn);
        startBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startActivity(new Intent(IntoActivity.this,
MainActivity.class));
            }
        });
    }
}
```

activity_intro:

```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:id="@+id/startBtn"
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:background="#F6F4F2"
    tools:context=".Activity.IntoActivity">

    <ImageView
        android:id="@+id/imageView4"
        android:layout_width="457dp"
        android:layout_height="986dp"
        android:layout_marginBottom="100dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:srcCompat="@drawable/img_6" />

    <TextView
        android:id="@+id/textView1"
        android:layout_width="110dp"
        android:layout_height="60dp"
        android:layout_marginBottom="552dp"
        android:fontFamily="cursive"
        android:text="Snack"
        android:textColor="@color/white"
        android:textSize="40dp"
        app:layout_constraintBottom_toBottomOf="@+id/imageView4"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent" />

    <TextView
        android:id="@+id/textView2"
        android:layout_width="152dp"
        android:layout_height="72dp"
        android:layout_marginTop="14dp"
        android:fontFamily="cursive"
        android:text="Squad"
        android:textColor="@color/white"
        android:textSize="55dp"
        app:layout_constraintBottom_toTopOf="@+id/textView4"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView1" />

    <TextView
        android:id="@+id/textView4"
        android:layout_width="283dp"
        android:layout_height="51dp"
        android:text="Food Ordering and Delivery App"
        android:textColor="@color/white"
        android:textSize="20dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/textView3"
        android:layout_width="0dp"
        android:layout_height="50dp"
        android:layout_marginStart="24dp"
        android:layout_marginEnd="24dp"
        android:layout_marginBottom="32dp"
        android:background="@drawable/brown_button"
        android:text="             Get Started"
        android:textColor="@color/white"
        android:textSize="30dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintStart_toStartOf="parent" />


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

MainActivity.java:

```java
package com.example.snacksquad.Activity;

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

import android.os.Bundle;

import com.example.snacksquad.Activity.Adapter.CategoryAdapter;
import com.example.snacksquad.Activity.Adapter.PopularAdapter;
import com.example.snacksquad.Activity.Domain.CategoryDomain;
import com.example.snacksquad.Activity.Domain.FoodDomain;
import com.example.snacksquad.R;

import java.util.ArrayList;

public class MainActivity extends AppCompatActivity {
    private RecyclerView.Adapter adapter,adapter2;
    private RecyclerView recyclerViewCategoryList,recyclerViewPopularList;

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

        recyclerViewCategory();
    }
    private void recyclerViewCategory() {
        LinearLayoutManager linearLayoutManager=new
LinearLayoutManager(this,LinearLayoutManager.HORIZONTAL,false);
        recyclerViewCategoryList=findViewById(R.id.recyclerView);
        recyclerViewCategoryList.setLayoutManager(linearLayoutManager);

        ArrayList<CategoryDomain> category = new ArrayList<>();
        category.add(new CategoryDomain("Pizza","cat_1"));
        category.add(new CategoryDomain("Burger","cat_2"));
        category.add(new CategoryDomain("Fries","cat_3"));
        category.add(new CategoryDomain("Ice cream","cat_4"));
        category.add(new CategoryDomain("Milkshakes","cat_5"));

        adapter=new CategoryAdapter(category);
        recyclerViewCategoryList.setAdapter(adapter);
    }
    private void recyclerViewPopular() {
        LinearLayoutManager linearLayoutManager=new
LinearLayoutManager(this,LinearLayoutManager.HORIZONTAL,false);
        recyclerViewPopularList=findViewById(R.id.recyclerView2);

        ArrayList<FoodDomain> foodList=new ArrayList<>();
        foodList.add(new FoodDomain("Pepperoni pizza","pizza1","slices
pepperoni,mozzerella cheese,fresh oregano, ground black pepper,pizza
sauce",275));
        foodList.add(new FoodDomain("Cheese Burger","burger","tomato,
cabbage, lettuce, cheese, Special sauce",200));
        foodList.add(new FoodDomain("Veg pizza","pizza","olives
oil,vegetable oil, pitted kalamta,cherry tomato,basil,fresh oregno",250));
```

```
        adapter2=new PopularAdapter(foodList);
        recyclerViewPopularList.setAdapter(adapter2);
    }
}
```

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=".Activity.MainActivity">

    <ScrollView
        android:id="@+id/scrollview"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginBottom="50dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical">

            <androidx.constraintlayout.widget.ConstraintLayout
                android:layout_width="match_parent"
                android:layout_height="match_parent">

                <TextView
                    android:id="@+id/textView"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:layout_marginStart="32dp"
                    android:layout_marginTop="32dp"
                    android:text="Hii there"
                    android:textColor="#d3ac8b"
                    android:textSize="30sp"
                    android:textStyle="italic"
                    app:layout_constraintStart_toStartOf="parent"
                    app:layout_constraintTop_toTopOf="parent" />

                <TextView
                    android:id="@+id/textView6"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:layout_marginTop="8dp"
                    android:text="Order and Eat!!"
                    android:textColor="#000000"
                    android:textSize="20sp"
                    app:layout_constraintStart_toStartOf="@+id/textView"
                    app:layout_constraintTop_toBottomOf="@+id/textView" />
                <ImageView
                    android:id="@+id/imageView"
                    android:layout_width="53dp"
                    android:layout_height="55dp"
                    android:layout_marginEnd="32dp"
                    app:layout_constraintBottom_toBottomOf="@+id/textView6"
                    app:layout_constraintEnd_toEndOf="parent"
                    app:layout_constraintTop_toTopOf="@+id/textView"
                    app:srcCompat="@drawable/btn2" />

                <EditText
                    android:id="@+id/editTextText"
                    android:layout_width="0dp"
                    android:layout_height="50dp"
                    android:layout_marginStart="32dp"
                    android:layout_marginTop="16dp"
                    android:layout_marginEnd="32dp"
                    android:background="@drawable/search_background"

android:drawableStart="@android:drawable/ic_menu_search"
                    android:drawablePadding="10dp"
                    android:ems="10"
                    android:hint="Find your food"
                    android:inputType="text"
                    android:paddingStart="20dp"
                    android:textSize="14sp"
                    android:textStyle="italic"
                    app:layout_constraintEnd_toEndOf="parent"
                    app:layout_constraintStart_toStartOf="parent"
                    app:layout_constraintTop_toBottomOf="@+id/textView6" />

                <androidx.constraintlayout.widget.ConstraintLayout
                    android:id="@+id/constraintLayout"
                    android:layout_width="match_parent"
                    android:layout_height="150dp"
                    android:layout_marginStart="32dp"
                    android:layout_marginTop="16dp"
                    android:layout_marginEnd="32dp"
                    android:background="@drawable/brown_light_banner"
                    app:layout_constraintEnd_toEndOf="parent"
                    app:layout_constraintStart_toStartOf="@id/textView"

app:layout_constraintTop_toBottomOf="@+id/editTextText">

                    <ImageView
                        android:id="@+id/imageView6"
                        android:layout_width="156dp"
                        android:layout_height="111dp"
                        app:layout_constraintBottom_toBottomOf="parent"
                        app:layout_constraintStart_toStartOf="parent"
                        app:layout_constraintTop_toTopOf="parent"
                        app:layout_constraintVertical_bias="1.0"
                        app:srcCompat="@drawable/bike" />

                    <TextView
                        android:id="@+id/textView5"
                        android:layout_width="wrap_content"
                        android:layout_height="wrap_content"
                        android:text="Free Delivery"
                        android:textColor="#000000"
                        android:textStyle="bold"
```

```
app:layout_constraintStart_toEndOf="@+id/imageView6"
                        app:layout_constraintTop_toTopOf="@+id/imageView6"
/>

                    <TextView
                        android:id="@+id/textView7"
                        android:layout_width="wrap_content"
                        android:layout_height="wrap_content"
                        android:text="Dec10 - Jan10"
                        android:textColor="#000000"
                        android:textSize="15dp"
                        app:layout_constraintEnd_toEndOf="@+id/textView5"

app:layout_constraintStart_toStartOf="@+id/textView5"

app:layout_constraintTop_toBottomOf="@+id/textView5" />

                    <androidx.constraintlayout.widget.ConstraintLayout
                        android:layout_width="wrap_content"
                        android:layout_height="wrap_content"
                        android:background="@drawable/brown_button"
                        app:layout_constraintBottom_toBottomOf="parent"
                        app:layout_constraintEnd_toEndOf="@+id/textView7"

app:layout_constraintStart_toStartOf="@+id/textView7"

app:layout_constraintTop_toBottomOf="@+id/textView7">

                        <TextView
                            android:id="@+id/textView8"
                            android:layout_width="wrap_content"
                            android:layout_height="wrap_content"
                            android:layout_marginStart="16dp"
                            android:layout_marginTop="8dp"
                            android:layout_marginEnd="16dp"
                            android:layout_marginBottom="8dp"
                            android:text="Order Now"
                            android:textColor="@color/white"
                            app:layout_constraintBottom_toBottomOf="parent"
                            app:layout_constraintEnd_toEndOf="parent"
                            app:layout_constraintStart_toStartOf="parent"
                            app:layout_constraintTop_toTopOf="parent" />
                    </androidx.constraintlayout.widget.ConstraintLayout>
                </androidx.constraintlayout.widget.ConstraintLayout>

                <TextView
                    android:id="@+id/textView10"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:layout_marginStart="32dp"
                    android:layout_marginTop="16dp"
                    android:text="Categories"
                    android:textColor="@color/black"
                    android:textSize="18sp"
                    android:textStyle="bold"
                    app:layout_constraintStart_toStartOf="parent"

app:layout_constraintTop_toBottomOf="@+id/constraintLayout" />

                <androidx.recyclerview.widget.RecyclerView
```

```
                    android:id="@+id/recyclerView"
                    android:layout_width="match_parent"
                    android:layout_height="match_parent"
                    android:paddingLeft="22dp"
                    android:paddingRight="22dp"
                    app:layout_constraintEnd_toEndOf="parent"
                    app:layout_constraintStart_toStartOf="parent"
                    app:layout_constraintTop_toBottomOf="@+id/textView10"
/>

                <TextView
                    android:id="@+id/textView12"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:layout_marginStart="32dp"
                    android:layout_marginTop="16dp"
                    android:text="Popular"
                    android:textColor="#000000"
                    android:textSize="18sp"
                    android:textStyle="bold|italic"
                    app:layout_constraintStart_toStartOf="parent"
                    app:layout_constraintTop_toBottomOf="@+id/recyclerView"
/>

                <androidx.recyclerview.widget.RecyclerView
                    android:id="@+id/recyclerView2"
                    android:layout_width="match_parent"
                    android:layout_height="match_parent"
                    app:layout_constraintEnd_toEndOf="parent"
                    app:layout_constraintHorizontal_bias="0.0"
                    app:layout_constraintStart_toStartOf="parent"
                    app:layout_constraintTop_toBottomOf="@+id/textView12"
/>


<com.google.android.material.floatingactionbutton.FloatingActionButton
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:layout_marginBottom="36dp"
                    android:backgroundTint="#d3ac8b"
                    android:elevation="3dp"
                    android:src="@drawable/cart"
                    app:backgroundTint="#d3ac8b"

app:layout_constraintBottom_toBottomOf="@+id/recyclerView2"
                    app:layout_constraintEnd_toEndOf="parent"
                    app:layout_constraintStart_toStartOf="parent"
                    app:maxImageSize="30dp"
                    app:tint="#ffffff">


</com.google.android.material.floatingactionbutton.FloatingActionButton>

            </androidx.constraintlayout.widget.ConstraintLayout>

        </LinearLayout>
    </ScrollView>

    <androidx.coordinatorlayout.widget.CoordinatorLayout
        android:layout_width="367dp"
        android:layout_height="56dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/scrollview">

        <com.google.android.material.bottomappbar.BottomAppBar
            android:id="@+id/app_bar"
            android:layout_width="339dp"
            android:layout_height="29dp"
            android:layout_gravity="bottom"
            android:backgroundTint="#f6f6f6"
            app:fabCradleMargin="8dp"
            app:fabCradleRoundedCornerRadius="50dp"
            app:fabCradleVerticalOffset="6dp"
            app:layout_anchor="@+id/app_bar"
            app:layout_anchorGravity="top|center">

            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="match_parent">

                <LinearLayout
                    android:id="@+id/homeBtn"
                    android:layout_width="0dp"
                    android:layout_height="match_parent"
                    android:layout_weight="0.2"
                    android:orientation="vertical">

                    <ImageView
                        android:id="@+id/imageView5"
                        android:layout_width="wrap_content"
                        android:layout_height="wrap_content"
                        android:layout_weight="1"
                        app:srcCompat="@drawable/btn1" />

                    <TextView
                        android:id="@+id/textView14"
                        android:layout_width="match_parent"
                        android:layout_height="wrap_content"
                        android:text="Home"
                        android:textAlignment="center"
                        android:textColor="#676767"
                        android:textSize="12sp" />
                </LinearLayout>

                <LinearLayout
                    android:id="@+id/ProfileBtn"
                    android:layout_width="0dp"
                    android:layout_height="match_parent"
                    android:layout_weight="0.2"
                    android:orientation="vertical">

                    <ImageView
                        android:id="@+id/imageView5"
                        android:layout_width="wrap_content"
                        android:layout_height="wrap_content"
                        android:layout_weight="1"
                        app:srcCompat="@drawable/btn2" />

                    <TextView
                        android:id="@+id/textView14"
                        android:layout_width="match_parent"
                        android:layout_height="wrap_content"
                        android:text="Profile"
                        android:textAlignment="center"
                        android:textColor="#676767"
                        android:textSize="12sp" />
                </LinearLayout>

                <LinearLayout
                    android:id="@+id/homeBtnx"
                    android:layout_width="0dp"
                    android:layout_height="match_parent"
                    android:layout_weight="0.2"
                    android:orientation="vertical">

                    <ImageView
                        android:id="@+id/imageView5"
                        android:layout_width="wrap_content"
                        android:layout_height="wrap_content"
                        android:layout_weight="1"
                        tools:visibility="invisible" />

                    <TextView
                        android:id="@+id/textView14"
                        android:layout_width="match_parent"
                        android:layout_height="wrap_content"
                        android:textAlignment="center"
                        android:textColor="#676767"
                        android:textSize="12sp" />
                </LinearLayout>

                <LinearLayout
                    android:id="@+id/SupportBtn"
                    android:layout_width="0dp"
                    android:layout_height="match_parent"
                    android:layout_weight="0.2"
                    android:orientation="vertical">

                    <ImageView
                        android:id="@+id/imageView5"
                        android:layout_width="wrap_content"
                        android:layout_height="wrap_content"
                        android:layout_weight="1"
                        app:srcCompat="@drawable/btn3" />

                    <TextView
                        android:id="@+id/textView14"
                        android:layout_width="match_parent"
                        android:layout_height="wrap_content"
                        android:text="Support"
                        android:textAlignment="center"
                        android:textColor="#676767"
                        android:textSize="12sp" />
                </LinearLayout>

                <LinearLayout
                    android:id="@+id/SettingsBtn"
                    android:layout_width="0dp"
                    android:layout_height="match_parent"
                    android:layout_weight="0.2"
                    android:orientation="vertical">
```

```xml
                    <ImageView
                        android:id="@+id/imageView5"
                        android:layout_width="wrap_content"
                        android:layout_height="wrap_content"
                        android:layout_weight="1"
                        app:srcCompat="@drawable/btn4" />

                    <TextView
                        android:id="@+id/textView14"
                        android:layout_width="match_parent"
                        android:layout_height="wrap_content"
                        android:text="Settings"
                        android:textAlignment="center"
                        android:textColor="#676767"
                        android:textSize="12sp" />
                </LinearLayout>
            </LinearLayout>
        </com.google.android.material.bottomappbar.BottomAppBar>

    </androidx.coordinatorlayout.widget.CoordinatorLayout>

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

Viewholder_category.xml:

```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:id="@+id/mainLayout"
    android:layout_width="200dp"
    android:layout_height="200dp"
    android:layout_margin="8dp">

    <ImageView
        android:id="@+id/categoryPic"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:srcCompat="@drawable/cat_1" />

    <TextView
        android:id="@+id/categoryName"
        android:layout_width="84dp"
        android:layout_height="46dp"
        android:layout_marginBottom="16dp"
        android:text="Title"
        android:textSize="30sp"
        android:textStyle="bold"
        app:layout_constraintBottom_toBottomOf="@+id/categoryPic"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent" />

    <ImageView
        android:id="@+id/categoryPic"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:srcCompat="@drawable/cat_2" />
    <TextView
        android:id="@+id/categoryName"
        android:layout_width="84dp"
        android:layout_height="46dp"
        android:layout_marginBottom="16dp"
        android:text="Title"
        android:textSize="30sp"
        android:textStyle="bold"
        app:layout_constraintBottom_toBottomOf="@+id/categoryPic"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent" />
    <ImageView
        android:id="@+id/categoryPic"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:srcCompat="@drawable/cat_3" />

    <TextView
        android:id="@+id/categoryName"
        android:layout_width="84dp"
        android:layout_height="46dp"
        android:layout_marginBottom="16dp"
        android:text="Title"
        android:textSize="30sp"
        android:textStyle="bold"
        app:layout_constraintBottom_toBottomOf="@+id/categoryPic"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent" />

    <ImageView
        android:id="@+id/categoryPic"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:srcCompat="@drawable/cat_4" />

    <TextView
        android:id="@+id/categoryName"
        android:layout_width="84dp"
        android:layout_height="46dp"
        android:layout_marginBottom="16dp"
        android:text="Title"
        android:textSize="30sp"
        android:textStyle="bold"
        app:layout_constraintBottom_toBottomOf="@+id/categoryPic"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent" />


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

Viewholder_popular.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="wrap_content"
    android:layout_height="wrap_content">
    <TextView
        android:id="@+id/title"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="4dp"
        android:layout_marginTop="16dp"
        android:layout_marginEnd="4dp"
        android:text="title"
        android:textColor="#373b54"
        android:textSize="14sp"
        android:textStyle="bold"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <ImageView
        android:id="@+id/imageView2"
        android:layout_width="146dp"
        android:layout_height="139dp"
        android:layout_marginStart="16dp"
        android:layout_marginEnd="16dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/title"
        app:srcCompat="@drawable/pepporonii" />

    <TextView
        android:id="@+id/fee"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="0.0"
        android:textColor="#373b54"
        android:textSize="18sp"
        android:textStyle="bold"
        app:layout_constraintEnd_toEndOf="@+id/imageView2"
        app:layout_constraintStart_toStartOf="@+id/imageView2"
        app:layout_constraintTop_toBottomOf="@+id/imageView2" />

    <TextView
        android:id="@+id/textView13"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="₹"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toStartOf="@+id/fee" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

CategoryAdapter.java:

```
package com.example.snacksquad.Activity.Adapter;

import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.constraintlayout.widget.ConstraintLayout;
import androidx.core.content.ContextCompat;
import androidx.recyclerview.widget.RecyclerView;

import com.bumptech.glide.Glide;
import com.example.snacksquad.Activity.Domain.CategoryDomain;
import com.example.snacksquad.R;

import java.util.ArrayList;

public class CategoryAdapter extends
RecyclerView.Adapter<CategoryAdapter.ViewHolder> {
    ArrayList<CategoryDomain>categoryDomains;

    public CategoryAdapter(ArrayList<CategoryDomain> categoryDomains) {
        this.categoryDomains = categoryDomains;
    }

    @Override
    public ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int
viewType) {
        View inflate=
LayoutInflater.from(parent.getContext()).inflate(R.layout.viewholder_catego
ry,parent,false);
        return new ViewHolder(inflate);
    }

    @Override
    public void onBindViewHolder(@NonNull ViewHolder holder, int position)
{

holder.categoryName.setText(categoryDomains.get(position).getTitle());
        String picUrl="";
        switch (position){
            case 0:{
                picUrl="cat_1";

holder.mainLayout.setBackground(ContextCompat.getDrawable(holder.itemView.g
etContext(),R.drawable.cat_background1));
                break;
            }
            case 1:{
                picUrl="cat_2";

holder.mainLayout.setBackground(ContextCompat.getDrawable(holder.itemView.g
etContext(),R.drawable.cat_background2));
                break;
            }
            case 2:{
                picUrl="cat_3";

holder.mainLayout.setBackground(ContextCompat.getDrawable(holder.itemView.g
etContext(),R.drawable.cat_background3));
                break;
            }
            case 3:{
                picUrl="cat_4";

holder.mainLayout.setBackground(ContextCompat.getDrawable(holder.itemView.g
etContext(),R.drawable.cat_background4));
                break;
            }
            case 4:{
                picUrl="cat_5";

holder.mainLayout.setBackground(ContextCompat.getDrawable(holder.itemView.g
etContext(),R.drawable.cat_background5));
                break;
            }
        }
        int
drawableResourceId=holder.itemView.getContext().getResources().getIdentifie
r(picUrl,"drawable",holder.itemView.getContext().getPackageName());

        Glide.with(holder.itemView.getContext())
                .load(drawableResourceId)
                .load(holder.categoryPic);
    }

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

    public class ViewHolder extends RecyclerView.ViewHolder {
        TextView categoryName;
        ImageView categoryPic;
        ConstraintLayout mainLayout;
        public ViewHolder(@NonNull View itemView) {
            super(itemView);
            categoryName=itemView.findViewById(R.id.categoryName);
            categoryPic=itemView.findViewById(R.id.categoryPic);
            mainLayout=itemView.findViewById(R.id.mainLayout);

        }
    }
}
```

PopularAdapter.java:

```
package com.example.snacksquad.Activity.Adapter;

import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.constraintlayout.widget.ConstraintLayout;
import androidx.core.content.ContextCompat;
import androidx.recyclerview.widget.RecyclerView;

import com.bumptech.glide.Glide;
import com.example.snacksquad.Activity.Domain.CategoryDomain;
import com.example.snacksquad.Activity.Domain.FoodDomain;
import com.example.snacksquad.R;

import java.util.ArrayList;

public class PopularAdapter extends
RecyclerView.Adapter<PopularAdapter.ViewHolder> {
    ArrayList<FoodDomain> categoryFood;

    public PopularAdapter(ArrayList<FoodDomain> categoryDomains) {
        this.categoryFood = categoryDomains;
    }

    @Override
    public ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int 
viewType) {
        View inflate=
LayoutInflater.from(parent.getContext()).inflate(R.layout.viewholder_catego
ry,parent,false);
        return new ViewHolder(inflate);
    }

    @Override
    public void onBindViewHolder(@NonNull ViewHolder holder, int position) 
{
        holder.categoryName.setText(categoryFood.get(position).getTitle());
        String picUrl="";
        switch (position){
            case 0:{
                picUrl="cat_1";

holder.mainLayout.setBackground(ContextCompat.getDrawable(holder.itemView.g
etContext(),R.drawable.cat_background1));
                break;
            }
            case 1:{
                picUrl="cat_2";

holder.mainLayout.setBackground(ContextCompat.getDrawable(holder.itemView.g
etContext(),R.drawable.cat_background2));
                break;
            }
            case 2:{
                picUrl="cat_3";

holder.mainLayout.setBackground(ContextCompat.getDrawable(holder.itemView.g
etContext(),R.drawable.cat_background3));
                break;
            }
            case 3:{
                picUrl="cat_4";

holder.mainLayout.setBackground(ContextCompat.getDrawable(holder.itemView.g
etContext(),R.drawable.cat_background4));
                break;
            }
            case 4:{
                picUrl="cat_5";

holder.mainLayout.setBackground(ContextCompat.getDrawable(holder.itemView.g
etContext(),R.drawable.cat_background5));
                break;
            }
        }
        int 
drawableResourceId=holder.itemView.getContext().getResources().getIdentifie
r(picUrl,"drawable",holder.itemView.getContext().getPackageName());
```

```
        Glide.with(holder.itemView.getContext())
                .load(drawableResourceId)
                .load(holder.categoryPic);
    }

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

    public class ViewHolder extends RecyclerView.ViewHolder {
        TextView categoryName;
        ImageView categoryPic;
        ConstraintLayout mainLayout;
        public ViewHolder(@NonNull View itemView) {
            super(itemView);
            categoryName=itemView.findViewById(R.id.categoryName);
            categoryPic=itemView.findViewById(R.id.categoryPic);
            mainLayout=itemView.findViewById(R.id.mainLayout);

        }
    }
}
```

CategoryDomain:

```
package com.example.snacksquad.Activity.Domain;

public class CategoryDomain {
    private String title;
    public String pic;

    public CategoryDomain(String title, String pic) {
        this.title = title;
        this.pic = pic;
    }

    public String getTitle() {
        return title;
    }

    public void setTitle(String title) {
        this.title = title;
    }

    public String getPic() {
        return pic;
    }

    public void setPic(String pic) {
        this.pic = pic;
    }

}
```

FoodDomain:

```
package com.example.snacksquad.Activity.Domain;

public class FoodDomain {
    private String title;
    private String pic;
    private String description;
    private int fee;
    private int numberInCart;

    public FoodDomain(String title, String pic, String description, int
fee) {
        this.title = title;
        this.pic = pic;
        this.description = description;
        this.fee = fee;
    }

    public FoodDomain(String title, String pic, String description, int
fee, int numberInCart) {
        this.title = title;
        this.pic = pic;
        this.description = description;
        this.fee = fee;
        this.numberInCart = numberInCart;
    }

    public String getTitle() {
        return title;
    }

    public void setTitle(String title) {
        this.title = title;
    }

    public String getPic() {
        return pic;
    }

    public void setPic(String pic) {
        this.pic = pic;
    }

    public String getDescription() {
        return description;
    }

    public void setDescription(String description) {
        this.description = description;
    }

    public int getFee() {
        return fee;
    }

    public void setFee(int fee) {
        this.fee = fee;
    }

    public int getNumberInCart() {
        return numberInCart;
    }

    public void setNumberInCart(int numberInCart) {
        this.numberInCart = numberInCart;
    }
}
```

**Demo video drive link:**

https://drive.google.com/drive/folders/16DQje8TOxCL_apmaqdv7gjJd_9sKJDqY?usp=sharing