A Final Year Report Project on

E-COMMERCE ONLINE SHOPPING APPLICATION (SINGLE VENDOR)

Submitted in partial fulfilment of the

BACHELOR IN COMPUTER APPLICATIONS
By

Aman Ahuja
Enrolment Number. AJU/180081
&
Akash deep
Enrolment Number. AJU/180303

Under the esteemed guidance of

Prof. Arvind Pandey

(HOD)

DEPARTMENT OF COMPUTER SCIENCE & IT

ARKA JAIN UNIVERSITY, JHARKHAND

Jamshedpur

2018-2021

ARKA JAIN UNIVERSITY, JHARKHAND

DEPARTMENT OF COMPUTER SCIENCE & IT

CERTIFICATE

This is to certify that the project entitled, " **e-commerce online shopping application (single vendor)** ", is bonafied work of **Aman ahuja & Akash Deep** bearing **Enrolment no- AJU/180081** & **Enrolment no- AJU/180303** submitted in partial fulfilment of the requirements for the award of degree of BACHELOR OF COMPUTER APPLICATIONS (BCA) from ARKA JAIN University, JHARKHAND.

HOD Principal

Date university Seal

ABSTRACT

This project is a Android based E-commerce shopping system for a single vendor. The project objective is to deliver the online shopping application into android platform.

This project is an attempt to provide the advantages of online shopping to customers of a real shop. It helps buying the products in the shop anywhere through internet by using an android device. Thus, the customer will get the service of online shopping and home delivery from a All-in-One vendor. This system can be implemented to any shop in the locality or to multinational branded shops having retail outlet chains to a single admin who can manage all the products online and delivers the product to the customers.

If shops are providing an online portal where their customers can enjoy easy shopping from anywhere, the shops won't be losing any more customers to the trending online shops such as flipkart or eBay. Since the application is available in the Smartphone it is easily accessible and always available.

From this project, vendor can handle the details of customer, stock, sales, shop billing and login details.

ACKNOWLEDGEMENT

It is a genuine pleasure to express my profound gratitude and deep regard to my guide “**Mr. Arvind Pandey”** our **Head of the Department** for his exemplary guidance, monitoring and constant encouragement.

Also, who gave me the golden opportunity to do this wonderful project on the topic “**E-COMMERCE SHOPPING APPLICATION**”, which helped me in research and I came to know about so many things.

Table of Contents

Chapter 1

1. Introduction

project is aimed at developing a Web application that depicts online Shopping of mobiles and purchasing using Payment Gateway. Using this software, companies can improve the efficiency of their services. Online Shopping is one of the applications to improve the marketing of the company's products. This web application involves all the features of the online shopping.

1.1. Overview

The project entitled "**E-COMMERCE ONLINE SHOPPING APPLICATION**" enables customer to buy mobiles or accessories from anywhere through online. This application advertises some of the products for shopping. To buy products, customer has to create an account. Those who does not have an account, they can only view the available product. They can't buy it. Once the customer has created account, not only he can view the products, he can also add the product to the cart and also, he can place an order to buy those products. This application then generates bill for that particular customer. After the confirmation, the customer has to enter his credit card details to buy those products.

Overall description:

1.1 Product Perspective

The product will be developed completely independent and dynamic website. Customer must have an account to purchase the product. This application stores all the information in the database which can be retrieved whenever needed and all the validations are performed during the entry of the data by the user thus ensuring that the user cannot enter any wrong data which could cause problem later.

1.2 Product Function

Initially customer has to register to the website to access most of the features of the application. The customer has to enter the details like username, password…etc. After registration customer will be able to purchase products and the purchased product can be added to the cart. Later customer has to enter his credit card details to buy the products. The confirmation of debit card numbers will be handled by the Pay pal website.

1.1.1 User Characteristics: The user of this product is supposed to be fairly educated about the usage of the computers. He should understand how to store products and he should have knowledge about various products so that they could be saved. A person who has no knowledge of computers will find it difficult to understand the system. But with a little knowledge it will be very easy to handle the project.

Admin: Admin adds the new product and accessories and stores in the database which can be retrieved and used whenever needed and all the validation are performed during the entry of the data. Thus, it ensures that the user cannot enter any wrong data which would cause problem later.

User: This application allows the user to access all the products available. To buy the products, customer must create an account in this website.

1.1.2 General constraints: The constraints of this project are - system must support the runtime files of visual studio 2008 and must be able to run all the web pages.

1.1.3 Assumption and dependencies: The project depends on the user's ability to understand the features of the online shopping and able to use the best of it. If the internet connection is not proper then this application will not work.

2. Functional requirements:

Login Module:

Input: Admin enters the Login Id and password.

Process Definition: Check's login Id and password is valid or not.

Output: Admin is directed to next page where he can add, delete or update the products.

New user Module:

Input: Customer enters their details to create new account.

Process Definition: Checks whether he entered all the details or not.

Output: Customer is directed to the next page where he can view all the available products

Product Module:

Input: Admin adds the new product which can be viewed by the users.

Process Definition: Checks whether all the fields are entered properly or not.

Output: Records will be added to the database.

Search Module:

Input: This module helps the customer to ease his search based on his interest. The search can be done on different categories like mobile model name, model number, colour, price etc

Process Definition: It retrieves the selected category from the database

Output: Displays what the subscribers like to search.

Cart Module:

Input: User can select any number of Mobile and add to the cart. He can also remove from the cart if he dislikes it later.

Process Definition: Checks whether all the fields are entered properly or not.

Output: Records will be added to the database.

Payment Module:

Input: This module describes the payment done by the customer. The payment information can include information like the model purchased, quantity, mode of payment etc.

Process Definition: Checks whether all the fields are entered or not.

Output: Displays the payment done by the customer.

Chapter 2

2.1. Objective of the project:

This application helps customer to find different mobiles, their features, and new updates easily. It is designed such a way that one can view all the updates of the mobile from any place through online. The software will help in easy maintaining and updating products in the website for the administrator. Also, quick and easy comparison of different products for the customers.

2.2. Scope of the project:

This system will reduce the manual operation required to maintain all the records of booking information. And also generates the various reports for analysis. Main concept of the project is to enter transaction reports and to maintain customer records. Hence this software can be used in any mobile showroom to maintain their record easily.

2.3. Software Requirements:

- Android Studio
- Java
- MySQL
- Laravel - The **PHP** Framework for Web Artisans.
- HTML
- CSS
- RestAPI

2.4. Hardware Requirements:

- Processor: i5 Gen 8th
- RAM: 4 GB or above
- Hard disk: 40 GB or above

2.5. Languages used:

- **Front End:** Java, Kotlin, JavaScript, HTML, CSS, BOOTSTRAP

- **Back End:** MySQL, Laravel
- **Local host:** XAMPP/WAMP/LAMP/MAMP

Chapter 3

3. MODULE DISCRIPTION

3.1 ADMINISTRATOR

The administrator is the super user of this application. Only admin have access into this admin page. Admin may be the owner of the shop. The administrator has all the information about all the users and about all the products.

The module is divided into different sub-modules:

- Manage Products
- Manage Users
- Manage Orders

3.1.1 Manage Products

- Add Products
  - The shopping cart contains different kind of products. The products can be classified into different categories by name. Admin can add new products into the existing system with all its details including an image.
- Delete Products
  - Administrator can delete the products based on the stock of that particular product.

3.1.2 Manager User

- View Users
  - The admin will have a list view of all the users registered in the system. Admin can view all the details of each user in the list.
- Add Users
  - Admin has privileges to add a user directly by providing the details.
- Delete Users
  - Administrator has the right to delete a user.

3.1.3 Manage Order

- View Order

  - Administrator can view the orders which is generated by the users. He can verify the details of the purchase.
- Delete Order
  - Admin can delete order from the orders list when the product is taken for delivery.

3.1.4 Manage Report

- Stock Report
  - Administrator can view the Report generated by the stock and inventory IN/OUT by the manufacturer
- Customer Order Report
  - Admin can view the and analyse the Customer orders report and check the reviews of the stocks which is approved or which is cancelled or return
- Sales Report
  - Admin can analyse the total sale for required date wise report and predict the valuable information matrices

3.1.5 Manage Payment Method

- Add Payment Method
  - Admin can add the new payment method to make the transaction more feasible for the users
- Delete Payment Method
  - Admin can delete the payment method for users

3.2 USERS

3.2.1 Registration

A new user will have to register in the system by providing essential details in order to view the products in the system.

3.2.2 Login

A user must login with his user's name and password to the system after registration.

3.2.3 Edit Profile

The user can view and edit the profile.

3.2.4 View Products

User can view the list of products based on their names after successful login. A detailed description of a particular product with product name, product details, product image, price can be viewed by users.

3.2.5 Search Product

Users can search for a particular product in the list by name.

3.2.6 Add to Cart

The user can add the desired product into his cart by clicking add to cart option on the product. He can view his cart by clicking on the cart button. All products added by cart can be viewed in the cart. User can remove an item from the cart by clicking remove.

3.2.7 Orders

In the order's Section the user will have a view of pending orders.

3.2.8 Select Address

After confirming the items in the cart, the user can submit the cart by providing a delivery address. On successful submitting the cart will become empty.

3.2.9 Choose Payment Method

After selecting the address user can choose the payment method and provide the payment details like UPI, card details.

3.2.10 Enter Coupon (if any)

Users can also enter the coupon if any and get the amazing discount on their current order.

CHAPTER 4

4.1. SYSTEM ANALYSIS

System analysis is the process of gathering and interpreting facts, diagnosing problems and using the information to recommend improvements on the system. System analysis is a problem-solving activity that requires intensive communication between the system users and system developers. System analysis or study is an important phase of any system development process. The system is viewed as a whole, the inputs are identified and the system is subjected to close study to identify the problem areas. The solutions are given as a proposal. The proposal is reviewed on user request and suitable changes are made. This loop ends as soon as the user is satisfied with the proposal.

4.2. Existing system

The current system for shopping is to visit the shop manually and from the available product choose the item customer want and buying the item by payment of the price of the item.

- User must go to shop and select products.
- It is difficult to identify the required product.
- Description of the product is limited.
- It is a time-consuming process.
- Not in reach of distant users.

4.3. Proposed system

In the proposed system customer need not go to the shop for buying the products. He can order the product he wish to buy through the application in his Smartphone. The shop owner will be admin of the system. Shop owner can appoint moderators who will help owner in managing the customers and product orders. The system also recommends a home delivery system for the purchased products.

4.4. Feasibility study

Feasibility requirements are undertaken to have a view of understanding the features that are incorporated in terms of the requirements to design and implementation. The references of the feasibility study will be divided into multiple sections so that the understandability can be organized so for example if we have to design software the required investment and flow of cash has to be properly analysed in the same way technical associations are acquired to be discussed.

- Technical feasibility.
- Operational feasibility.
- Economic feasibility.

4.5. Technical feasibility

The customization that has to be operational when a particular identity or field is changed has to be replicated in real time which has to be properly designed and tested for the workability. All types of working references are provided in terms of the resources within the company premises. The integrated settings that are required for the accompanied sharing and alert system will also be associated properly and the vendor integrations will be tested for security provisions. The accessibility of the standards that are required will be properly define so as to acknowledge the formats of reports which has to be in-clubbed in different types of reporting sections that are incorporated.

4.6. Operational feasibility

Apprehension of working is required to be properly planed as when the system will be provided to multiple users a proper support has to be provided as multiple integrated functionalities are provided. The applications that may arise has to be drafted it so that in the real time working it can be properly acknowledged as multiple Global finance will be incorporated to the system. A proper documentation will be provided with all variations of report and the standing in the customization options that are provided under the setting

4.7. Financial feasibility

Financial return on investment calculation so will be done by the financial team with the help of multiple associated software that are available will be used. Considerations of the cash flow will be determined as the source of income has to be properly acknowledged

4.8. SYSTEM REQUIREMENT SPECIFICATION

4.8.1 GENERAL DESCRIPTION

- **Product Description**:

  The system consists of two parts. A web application which can provide the online shopping service and an android application for the customer to access the web service from his Smartphone. Web application should be able to help the customer for selecting his item and to help the owner in managing the orders from the customers.

- **Problem Statement:**

  As online shopping became a trend nowadays the regular shops are losing their customers to online brands. Customers have effortless shopping experience and saving time through shopping online. For competing with those online brands, if shops are providing an online portal where their customers can shop through internet and get the products at their doors it will increase the number of customers.

4.8.2 SYSTEM OBJECTIVES

- To provide an android application for online shopping of products in an existing shop.
- To provide a online shopping web site for the same shop.

4.9. NON-FUNCTIONAL REQUIREMENTS

4.9.1 EFFICIENCY REQUIREMENT

- When an online shopping cart android application implemented customer can purchase product in an efficient manner.

4.9.2 RELIABILITY REQUIREMENT

- The system should provide a reliable environment to both customers and owner. All orders should be reaching at the admin without any errors.

4.9.3 USABILITY REQUIREMENT

- The android application is designed for user friendly environment and ease of use.

4.9.4 IMPLEMENTATION REQUIREMENT

- Implementation of the system using CSS and html in front end with node.js as back end and it will be used for database connectivity. And the database part is developed by MySQL. Responsive web designing is used for making the website compatible for any type of screen.

4.9.5 DELIVERY REQUIREMENT

- The whole system is expected to be delivered in four months of time with a weekly evaluation by the project guide.

4.9.6 FUNCTIONAL REQUIREMENTS

4.10. USER

4.10.1. USER LOGIN

Description of feature

- This feature used by the user to login into system. A user must login with his user's name and password to the system after registration. If they are invalid, the user not allowed to enter the system.

Functional requirement

- Username and password will be provided after user registration is confirmed.
- Password should be hidden from others while typing it in the field

4.10.2. REGISTER NEW USER

Description of feature

- A new user will have to register in the system by providing essential details in order to view the products in the system. The admin must accept a new user by unblocking him.

Functional requirement

- System must be able to verify and validate information.
- The system must encrypt the password of the customer to provide security.

4.10.3. PURCHASING AN ITEM

Description of feature

- The user can add the desired product into his cart by clicking add to cart option on the product. He can view his cart by clicking on the cart button. All products added by cart can be viewed in the cart. User can remove an item from the cart by clicking remove. After confirming the items in the cart, the user can submit the cart by providing a delivery address. On successful submitting the cart will become empty.

Functional requirement

- System must ensure that, only a registered customer can purchase items.

4.11. ADMIN

4.11.1. MANAGE USER

Description of feature

- The administrator can add user, delete user, view user and block user.

4.11.2. MANAGE MANUFACTURER

Description of feature

- The administrator can add Manufacturer, delete manufacturer, block manufacturer and search for a manufacturer.

4.11.3. MANAGE PRODUCTS

Description of feature

- The administrator can add product, delete product and view product.

4.11.4. MANAGE ORDERS

Description of feature

- The administrator can view orders and delete orders.

Functional requirements

- The system must identify the login of the admin.-Admin account should be secured so that only owner of the shop can access that account

Chapter 5

5.1. System Design

System design is the solution for the creation of a new system. This phase focuses on the detailed implementation of the feasible system. It emphasis on translating design. Specifications to performance specification. System design has two phases of development

- Logical design
- Physical design

During logical design phase the analyst describes inputs (sources), output s(destinations), databases (data sores) and procedures (data flows) all in a format that meets the user requirements. The analyst also specifies the needs of the user at a level that virtually determines the information flow in and out of the system and the data resources. Here the logical design is done through data flow diagrams and database design. The physical design is followed by physical design or coding. Physical design produces the working system by defining the design specifications, which specify

exactly what the candidate system must do. The programmers write the necessary programs that accept input from the user, perform necessary processing on accepted data and produce the required report on a hard copy or display it on the screen.

5.2. INPUT AND OUTPUT DESIGN

INPUT DESIGN:

Input design is the link that ties the information system into the world of its users.

The input design involves determining the inputs, validating the data, minimizing the data entry and provides a multi-user facility. Inaccurate inputs are the most common cause of errors in data processing. Errors entered by the data entry operators can be controlled by input design. The user-originated inputs are converted to a computer-based format in the input design. Input data are collected and organized into groups of similar data. Once identified, the appropriate input media are selected for processing. All the input data are validated and if any data violates any conditions, the user is warned by a message. If the data satisfies all the conditions, it is transferred to the appropriate tables in the database. In this project the student details are to be entered at the time of registration. A page is designed for this purpose which is user friendly and easy to use. The design is done such that users get appropriate messages when exceptions occur.

OUTPUT DESIGN:

Computer output is the most important and direct source of information to the user. Output design is a very important phase since the output needs to be in an efficient manner. Efficient and intelligible output design improves the system relationship with the user and helps in decision making. Allowing the user to view the sample screen is important because the user is the ultimate judge of the quality of output. The output module of this system is the selected notifications.

5.3. DATABASE DESIGN:

Databases are the storehouses of data used in the software systems. The data is stored in tables inside the database. Several tables are created for the manipulation of the data for the system. Two essential settings for a database are

- the field that is unique for all the record occurrences.
- the field used to set relation between tables.

Normalization is a technique to avoid redundancy in the tables.

5.3.1 SYSTEM TOOLS

- The various system tools that have been used in developing both the front end and the back end of the project are being discussed in this chapter.

5.3.2 FRONT END:

- Android Studio, Kotlin, Laravel, HTML, CSS, JAVA SCRIPT are utilized to implement the frontend.

5.3.2.1 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:

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

5.3.2.2. Kotlin

Kotlin is a general purpose, free, open source, statically typed “pragmatic” programming language initially designed for the JVM (Java Virtual Machine) and Android that combines object-oriented and functional programming features.

JetBrains uses Kotlin in many of its products including its flagship IntelliJ IDEA.

5.3.2.3 Laravel

Laravel is a PHP framework. In layman’s terms, instead of programming everything from the first line to the last, Laravel offers something like a “skeleton” of a web application.

It has many advantages like, Authorization and authentication, session and cookie management, fixing the most common web vulnerabilities and it is also the cost efficient and easy to learn and implement

5.3.2.4 HTML (Hyper Text Markup Language)

HTML is a syntax used to format a text document on the web.

5.3.2.5 CSS (Cascading Style Sheets)

CSS is a style sheet language used for describing the look and formatting of a document written in a markup language.

5.3.2.6 Java Script

JS is a dynamic computer programming language. It is most commonly used as part of web browsers, whose implementations allow client-side scripts to interact with the user, control the browser, communicate asynchronously, and alter the document content that is displayed.

Java Script is used to create pop-up windows displaying different alerts in the system like “User registered successfully”,” Product added to cart” etc.

5.3.3. BACK END

- The back end is implemented using MySQL which is used to design the databases.

5.3.3.1 MySQL

MySQL is the world's second most widely used open-source relational database management system (RDBMS). The SQL phrase stands for Structured Query Language.

An application software called Navicat was used to design the tables in MySQL.

5.4. Android Activities

An activity is a single, focused thing that the user can do. Almost all activities interact with the user, so the Activity class takes care of creating a window for you in which you can place your UI with setContentView. While activities are often presented to the user as full-screen windows, they can also be used in other ways: as floating windows (via a theme with android.R.attr#windowIsFloating set), Multi-Window mode or embedded into other windows.[37] There are two methods almost all subclasses of Activity will implement:

- onCreate is where you initialize your activity. Most importantly, here you will usually call setContentView(int) with a layout resource defining your UI, and using findViewById to retrieve the widgets in that UI that you need to interact with programmatically.
- onPause is where you deal with the user pausing active interaction with the activity. Any changes made by the user should at this point be committed (usually to the android.content.ContentProvider holding the data). In this state the activity is still visible on screen.

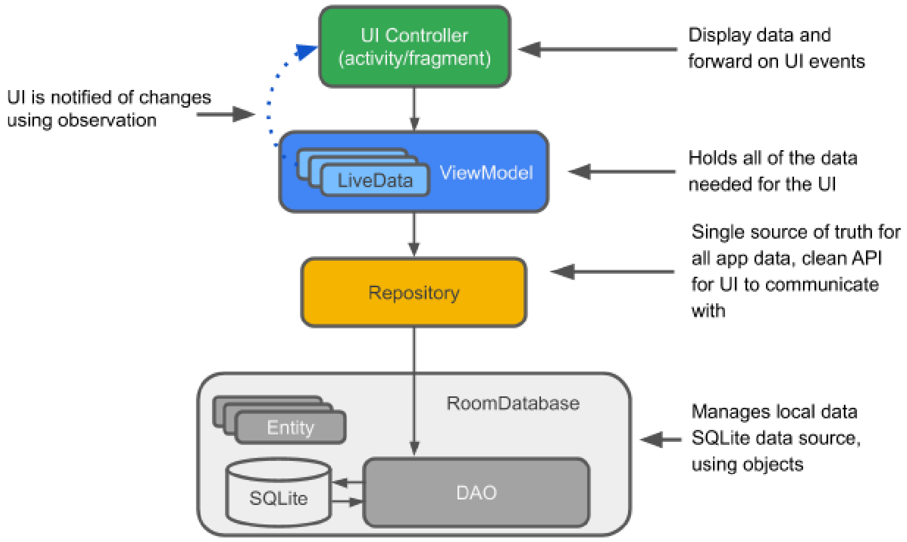
To be of use with Context.startActivity(), all activity classes must have a corresponding ¡activity¿ declaration in their package's AndroidManifest.xml.

5.4.1 Fragments

A Fragment represents a behaviour or a portion of user interface in a Fragment Activity. You can combine multiple fragments in a single activity to build a multi-pane UI and reuse a fragment in multiple activities. You can think of a fragment as a modular section of an activity, which has its own lifecycle, receives its own input events, and which you can add or remove while the activity is running (sort of like a" sub activity" that you can reuse in different activities).

A fragment must always be hosted in an activity and the fragment's lifecycle is directly affected by the host activity's lifecycle. For example, when the activity is paused, so are all fragments in it, and when the activity is destroyed, so are all fragments. However, while an activity is running, you can manipulate each fragment independently, such as add or remove them. When you perform such a fragment transaction, you can also add it to a back stack that's managed by the activity—each back stack entry in the activity is a record of the fragment transaction that occurred. The back stack allows the user to reverse a fragment transaction (navigate backwards), by pressing the Back button.

When you add a fragment as a part of your activity layout, it lives in a ViewGroup inside the activity's view hierarchy and the fragment defines its own view layout. You can insert a fragment into your activity layout by declaring the fragment in the activity's layout file, as a ¡fragment¿ element, or from your application code by adding it to an existing ViewGroup.


UI Controller (activity/fragment)
Display data and forward on UI events
UI is notified of changes using observation
LiveData
ViewModel
Holds all of the data needed for the UI
Repository
Single source of truth for all app data, clean API for UI to communicate with
RoomDatabase
Entity
SQLite
DAO
Manages local data SQLite data source, using objects

Chapter 6

6.1. DFD Diagram

DATA FLOW DIAGRAM

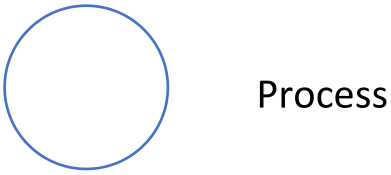
A Data Flow Diagram (DFD) is a structured analysis and design tool that can be used for flowcharting. A DFD is a network that describes the flow of data and the processes that change or transform the data throughout a system. This network is constructed by using a set of symbols that do not imply any physical implementation. It has the purpose of clarifying system requirements and identifying major transformations. So, it is the starting point of the design phase that functionally decomposes the requirements specifications down to the lowest level of detail. DFD can be considered to an abstraction of the logic of an information oriented or a process-oriented system flow-chart. For these reasons DFD's are often referred to as logical data flow diagrams.

EXTERNAL ENTITY

An external entity is a source or destination of a data flow. Only those entities which originate or receive data are represented on a data flow diagram. The symbol used is a rectangular box.

PROCESS

A process shows a transformation or manipulation of data flow within the system. The symbol used is an oval shape.

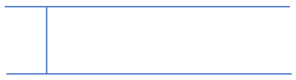


DATAFLOW

The data flow shows the flow of information from a source to its destination. Data flow is represented by a line, with arrowheads showing the direction of flow. Information always flows to or from a process and may be written, verbal or electronic. Each data flow may be referenced by the processes or data stores at its head and tail, or by a description of its contents.

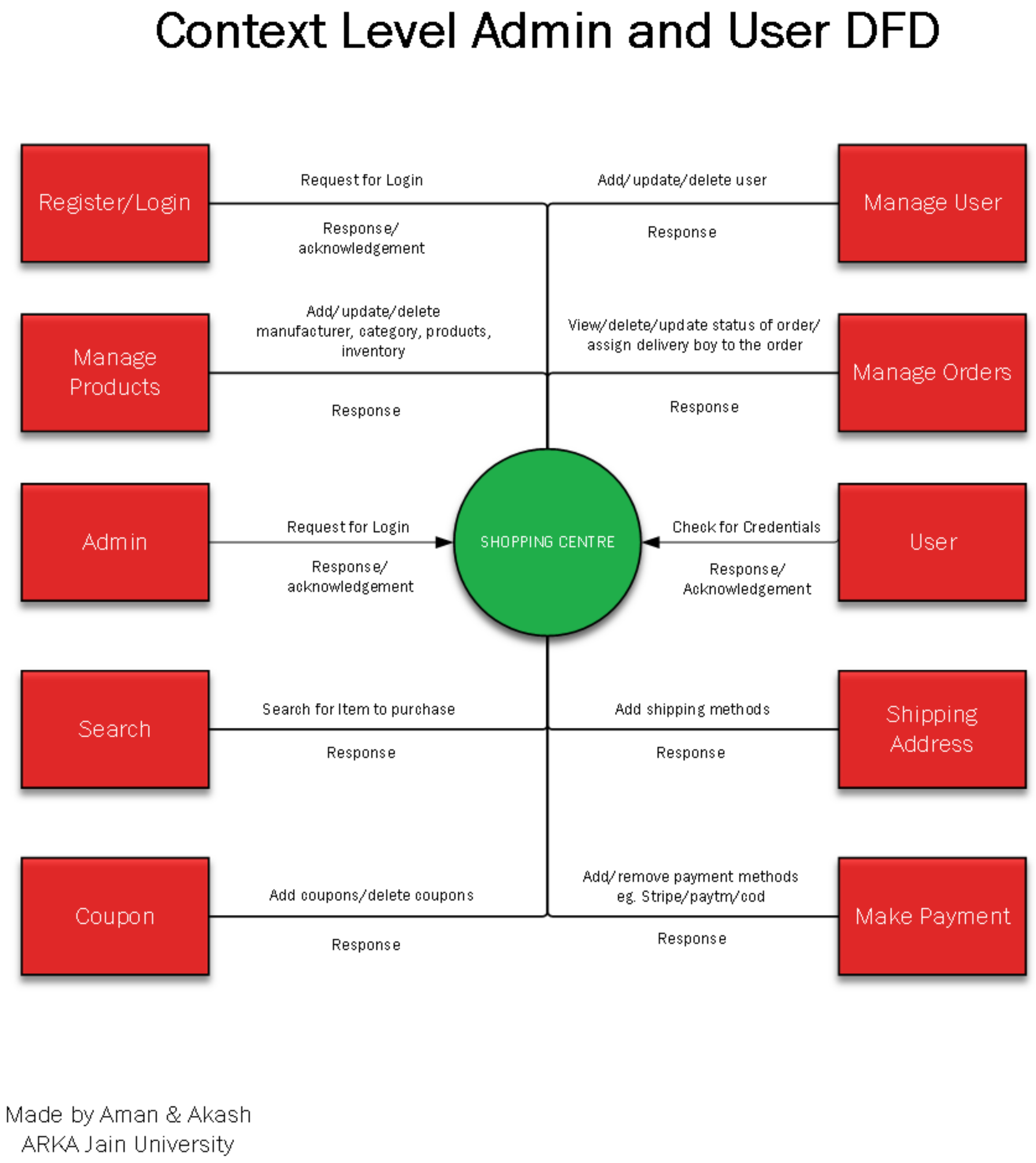
DATA STORE

A data store is a holding place for information within the system: It is represented by an open-ended narrow rectangle. Data stores may be long-term files such as sales ledgers, or may be short-term accumulations: for example, batches of documents that are waiting to be processed. Each data store should be given a reference followed by an arbitrary number.



Data store

6.1.1. Data Flow Diagram context level



Context Level Admin and User DFD

Register/Login
Request for Login
Response/ acknowledgement
Add/ update/delete user
Response
Manage User
Manage Products
Add/ update/delete manufacturer, category, products, inventory
Response
View/delete/update status of order/ assign delivery boy to the order
Response
Manage Orders
Admin
Request for Login
Response/ acknowledgement
SHOPPING CENTRE
Check for Credentials
Response/ Acknowledgement
User
Search
Search for Item to purchase
Response
Add shipping methods
Response
Shipping Address
Coupon
Add coupons/delete coupons
Response
Add/ remove payment methods eg. Stripe/paytm/cod
Response
Make Payment

Made by Aman & Akash
ARKA Jain University

fig 1.0

Admin

6.1.2. Data Flow Diagram level 1

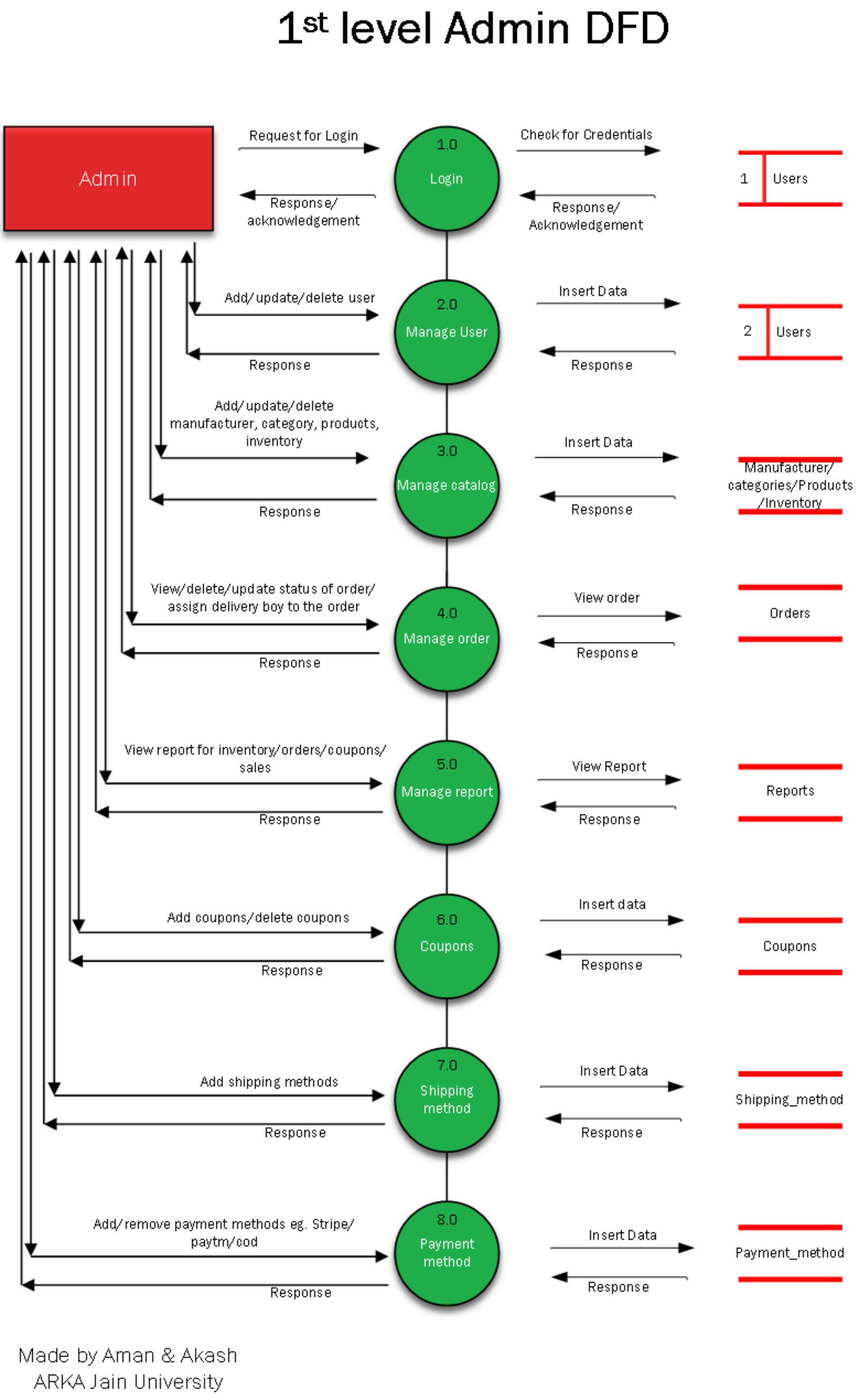


fig 2.0

6.1.3. Data Flow Diagram level 2 Admin

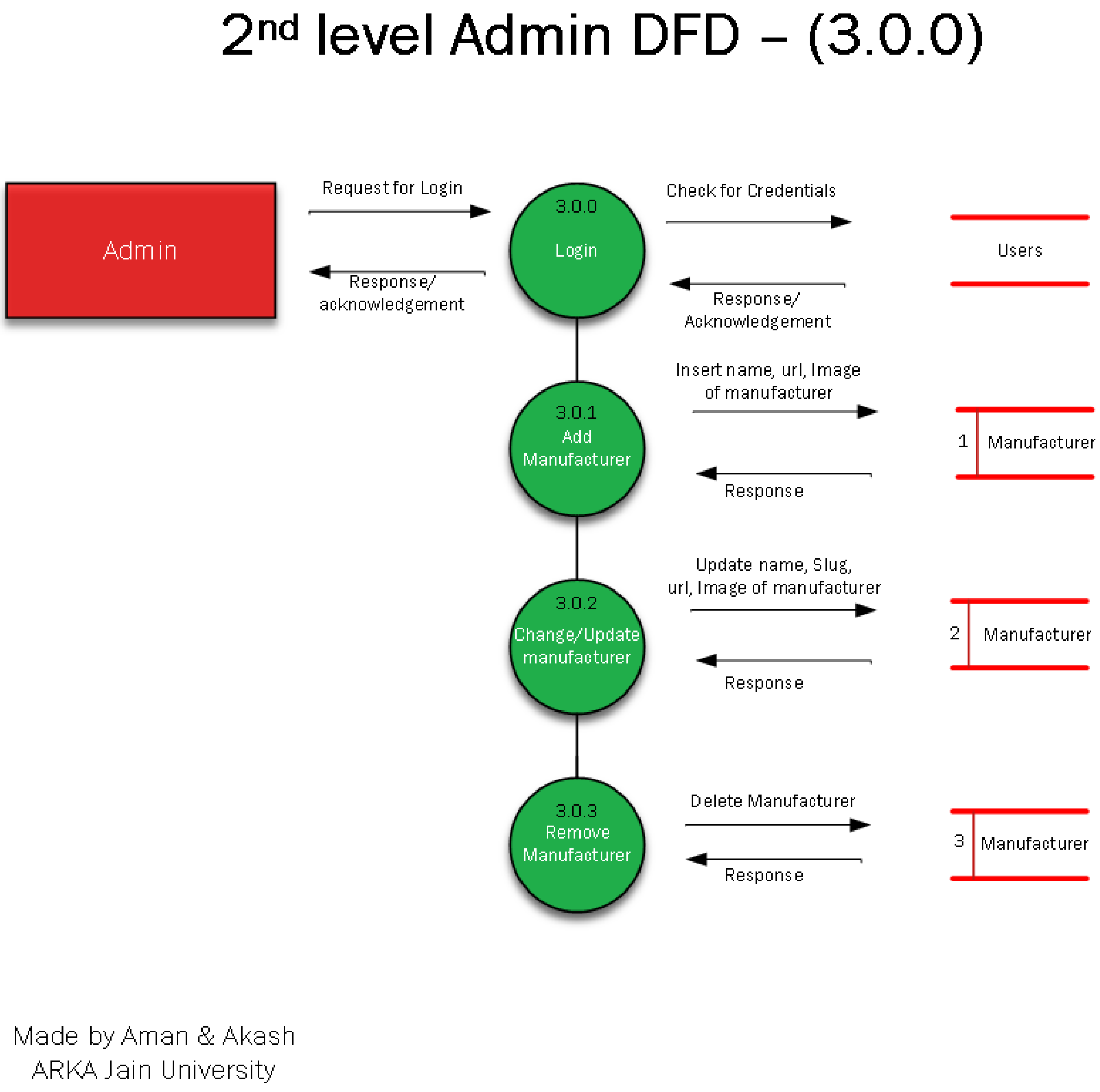

2nd level Admin DFD – (3.0.0)
Admin
Request for Login
Response/ acknowledgement
3.0.0 Login
Check for Credentials
Response/ Acknowledgement
Users
3.0.1 Add Manufacturer
Insert name, url, Image of manufacturer
Response
1 Manufacturer
3.0.2 Change/Update manufacturer
Update name, Slug, url, Image of manufacturer
Response
2 Manufacturer
3.0.3 Remove Manufacturer
Delete Manufacturer
Response
3 Manufacturer
Made by Aman & Akash
ARKA Jain University

fig 3.0

6.1.4. Data Flow Diagram level 2 Admin

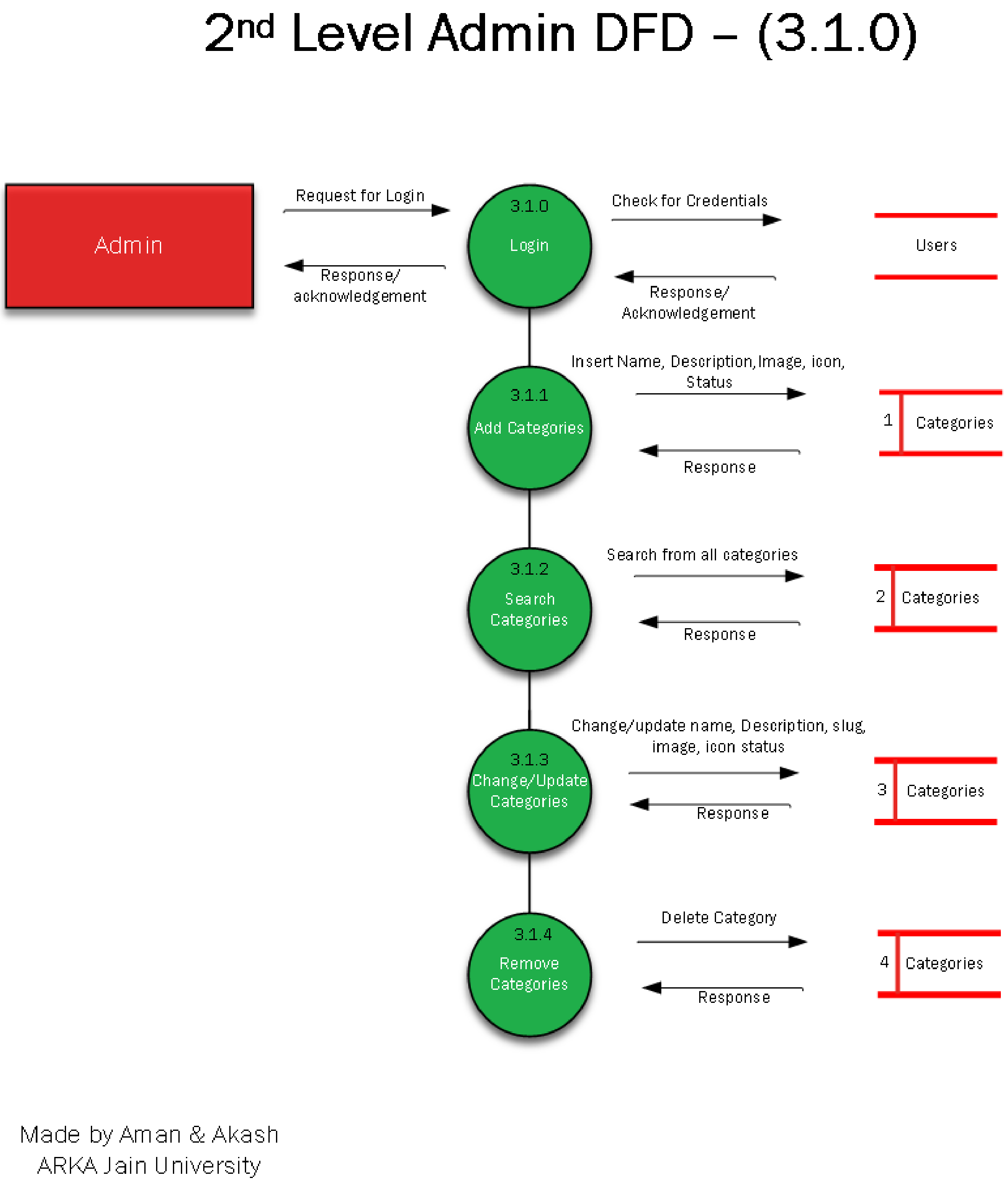


fig 4.0

6.1.5. Data Flow Diagram level 2 Admin

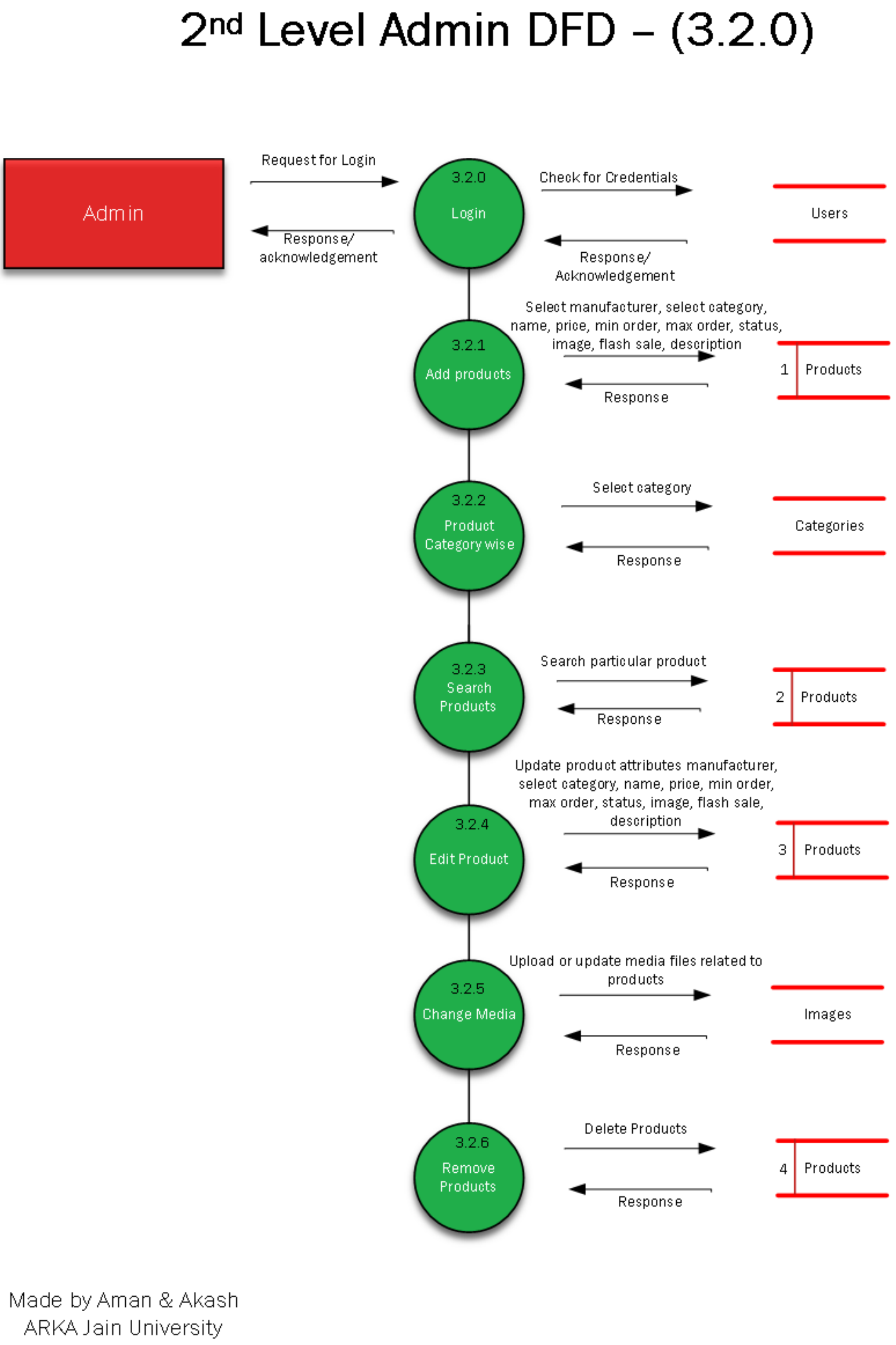

2nd Level Admin DFD – (3.2.0)
Admin
Request for Login
Response/ acknowledgement
3.2.0
Login
Check for Credentials
Response/ Acknowledgement
Users
3.2.1
Add products
Select manufacturer, select category, name, price, min order, max order, status, image, flash sale, description
Response
1
Products
3.2.2
Product Category wise
Select category
Response
Categories
3.2.3
Search Products
Search particular product
Response
2
Products
3.2.4
Edit Product
Update product attributes manufacturer, select category, name, price, min order, max order, status, image, flash sale, description
Response
3
Products
3.2.5
Change Media
Upload or update media files related to products
Response
Images
3.2.6
Remove Products
Delete Products
Response
4
Products
Made by Aman & Akash
ARKA Jain University

Fig 5.0

6.1.6. Data Flow Diagram level 2 Admin

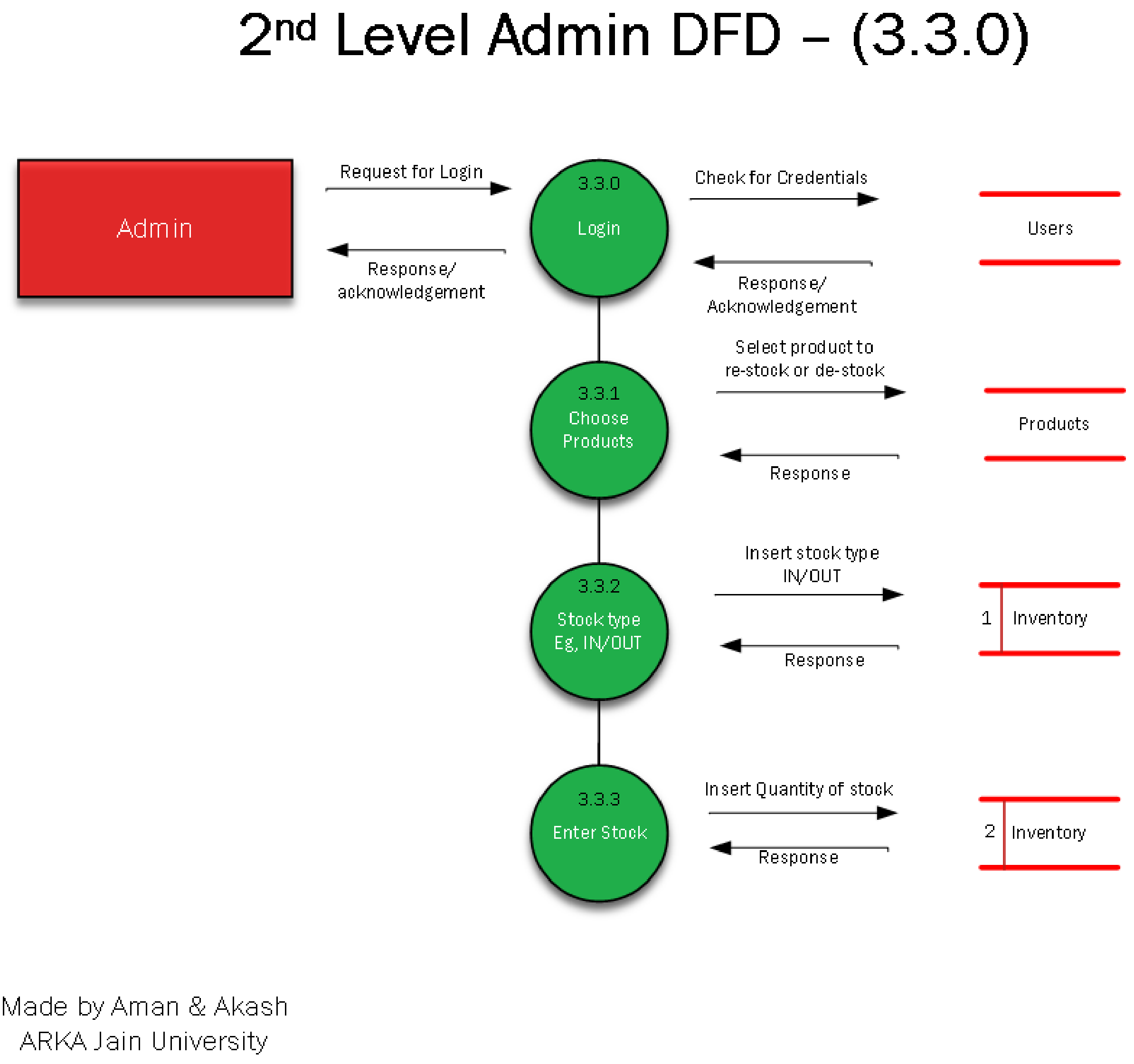


Fig 6.0

6.1.7. Data Flow Diagram level 2 Admin

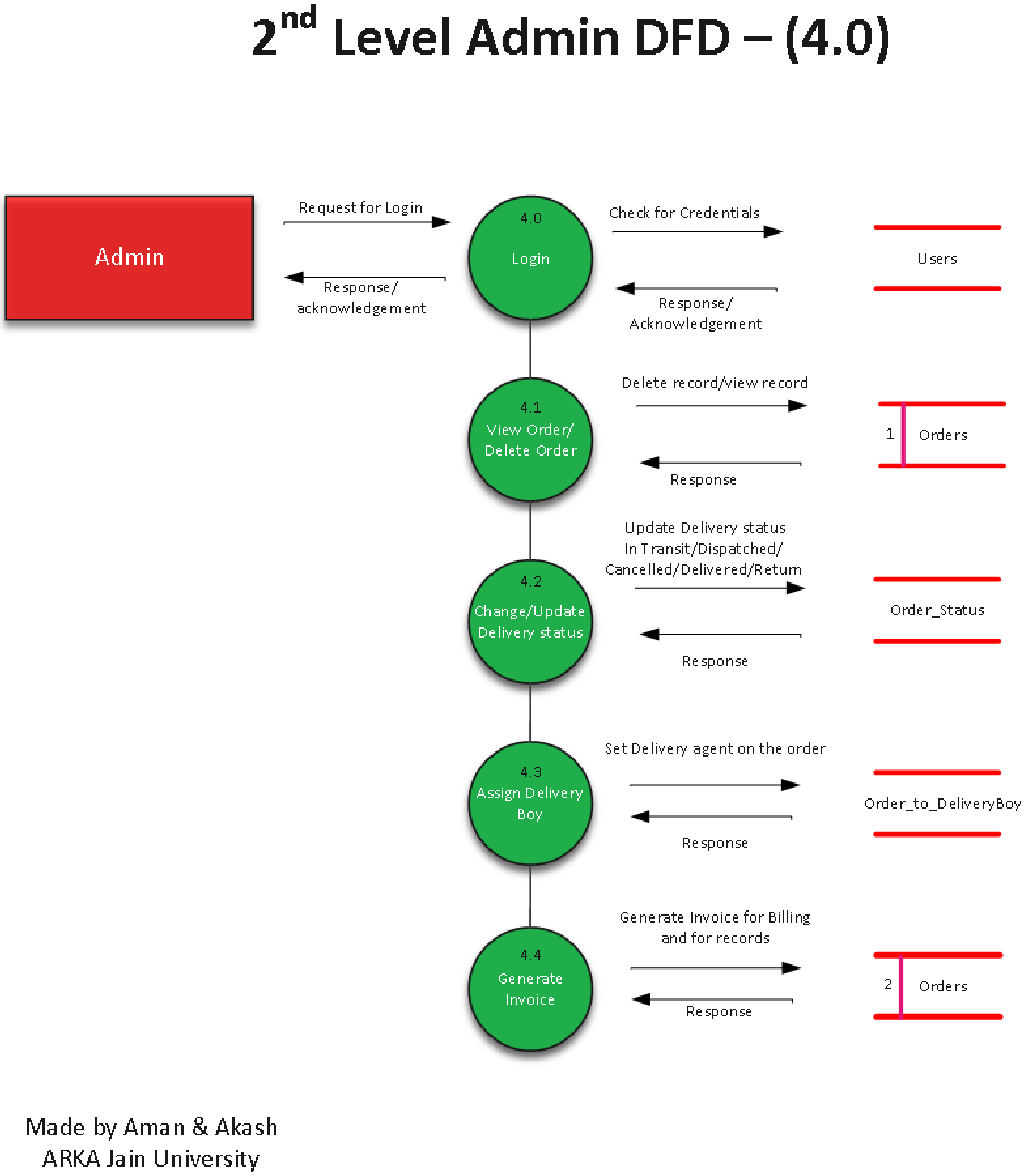

2nd Level Admin DFD – (4.0)
Admin
Request for Login
Response/ acknowledgement
4.0 Login
Check for Credentials
Response/ Acknowledgement
Users
4.1 View Order/ Delete Order
Delete record/view record
Response
1 Orders
4.2 Change/Update Delivery status
Update Delivery status In Transit/Dispatched/ Cancelled/Delivered/Return
Response
Order_Status
4.3 Assign Delivery Boy
Set Delivery agent on the order
Response
Order_to_DeliveryBoy
4.4 Generate Invoice
Generate Invoice for Billing and for records
Response
2 Orders
Made by Aman & Akash
ARKA Jain University

Fig 7.0

6.1.8. Data Flow Diagram level 2 Admin

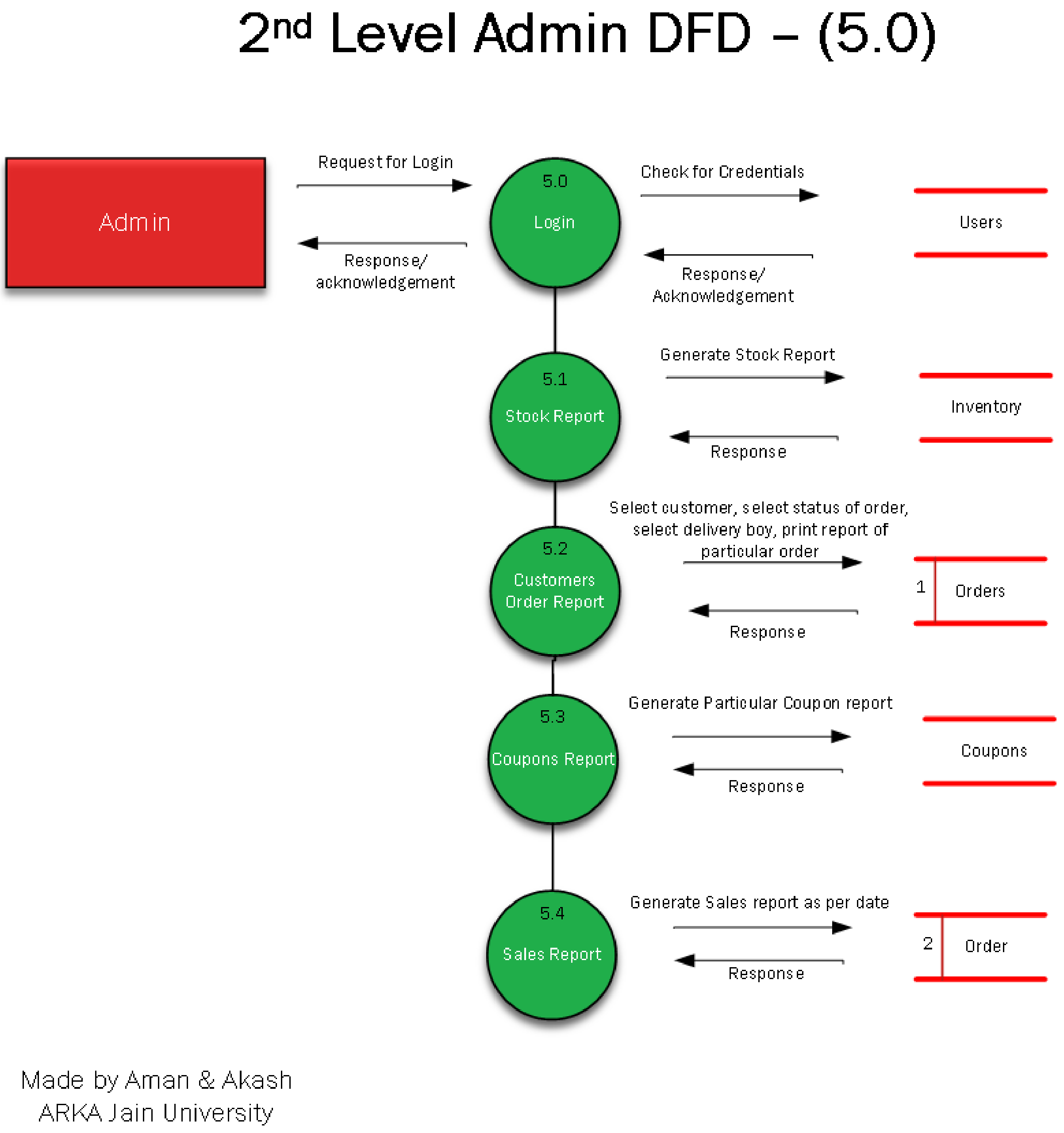


Fig 8.0

6.1.9. Data Flow Diagram level 2 Admin

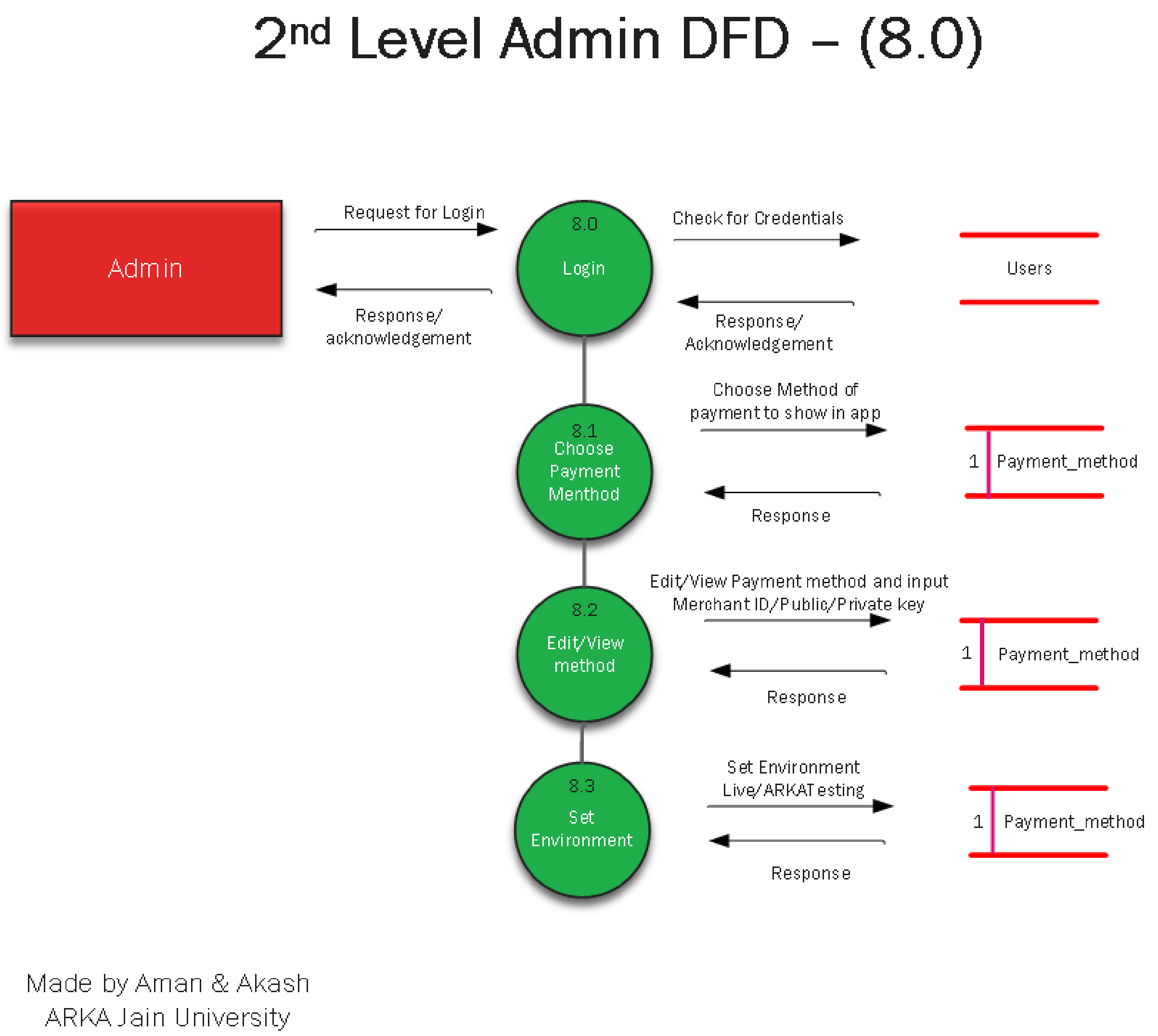


Fig 9.0

Users

6.2.1. Level 1 DFD for Users

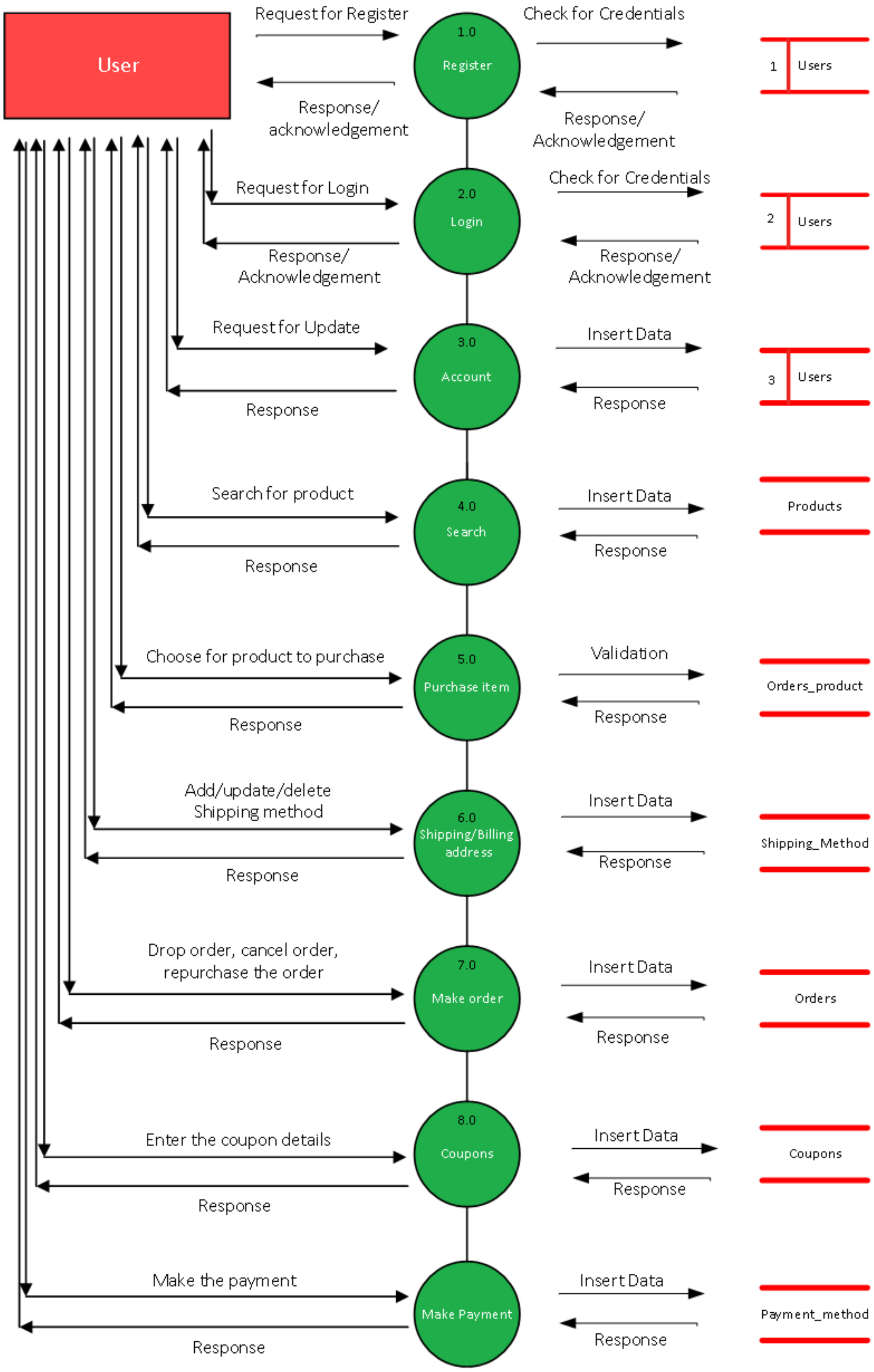


Fig 1.0

6.2.2. Level 2 DFD for Users

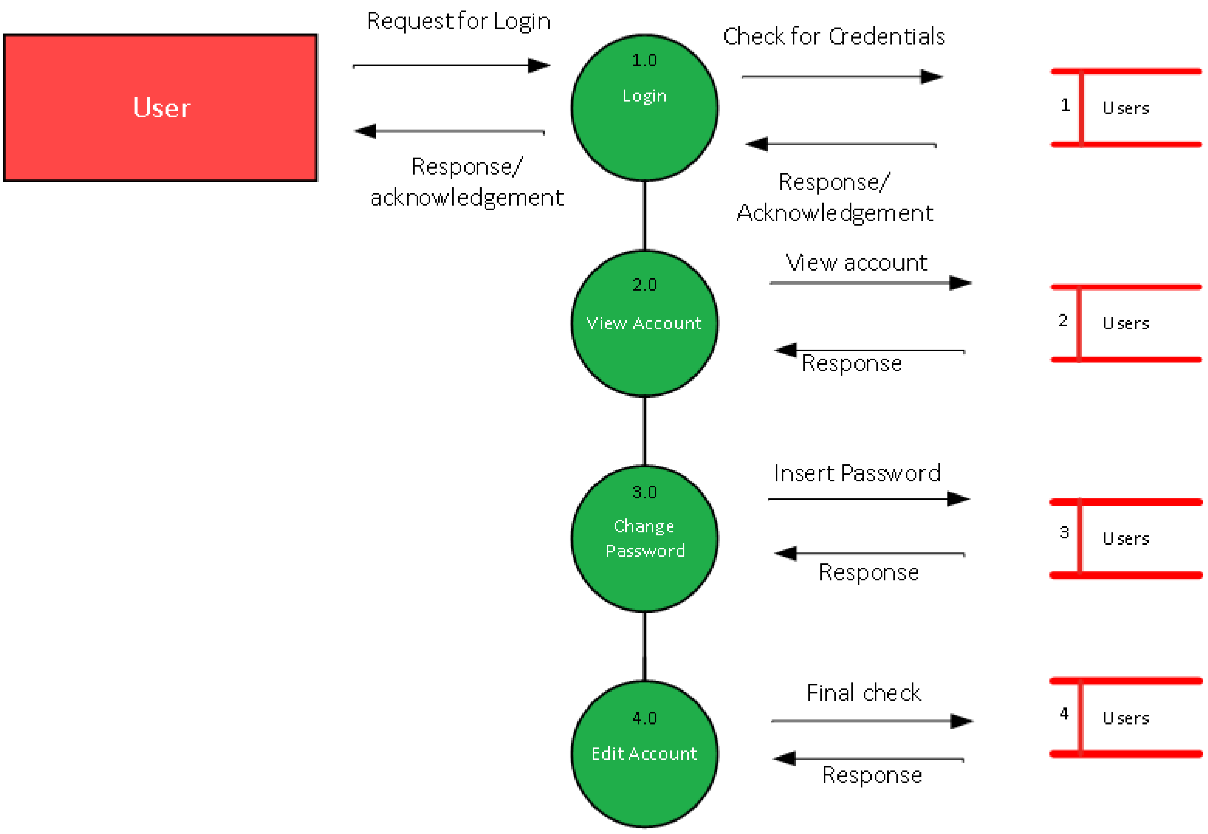


Fig 2.0

6.2.3. Level 2 DFD for Users

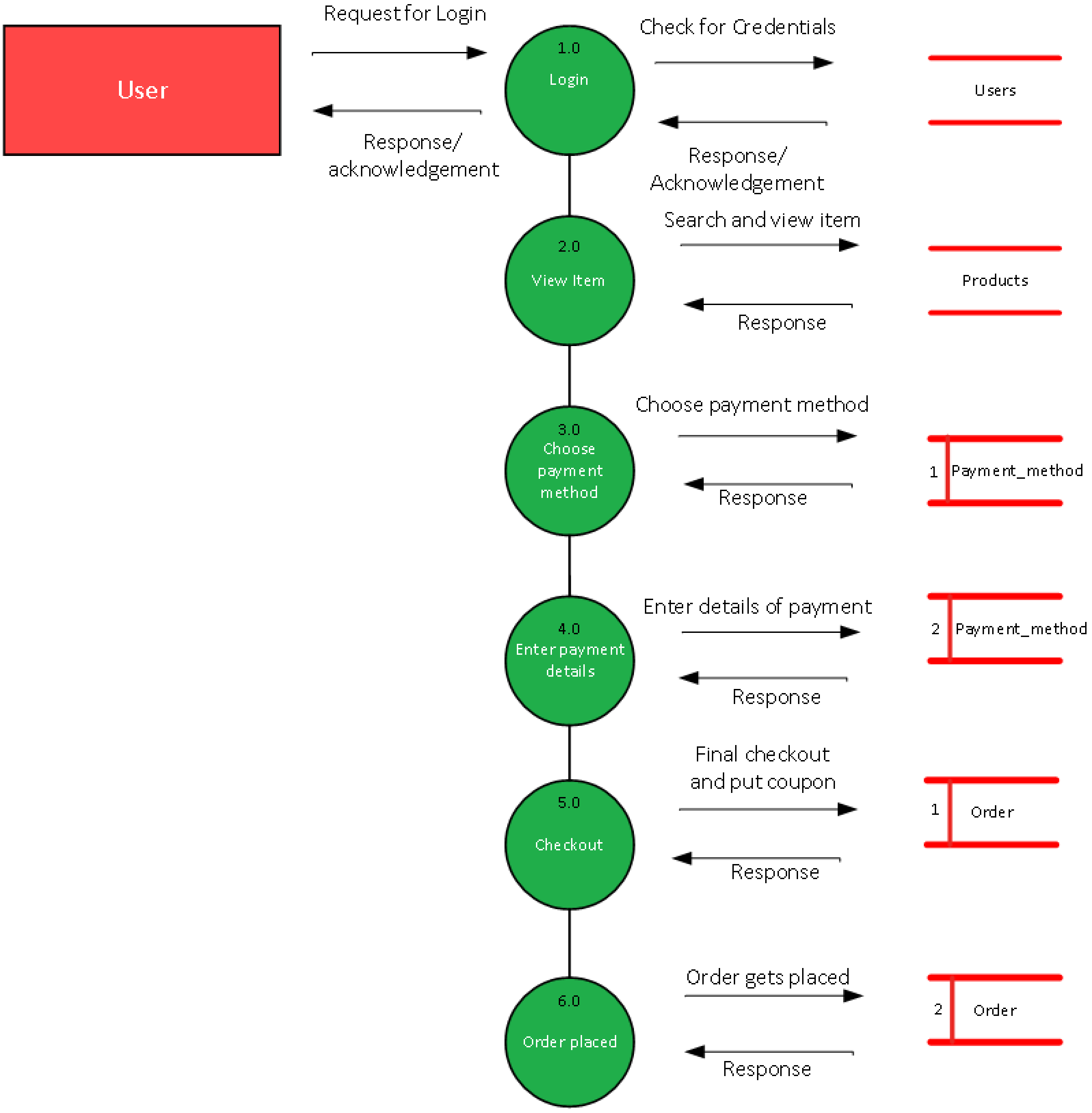
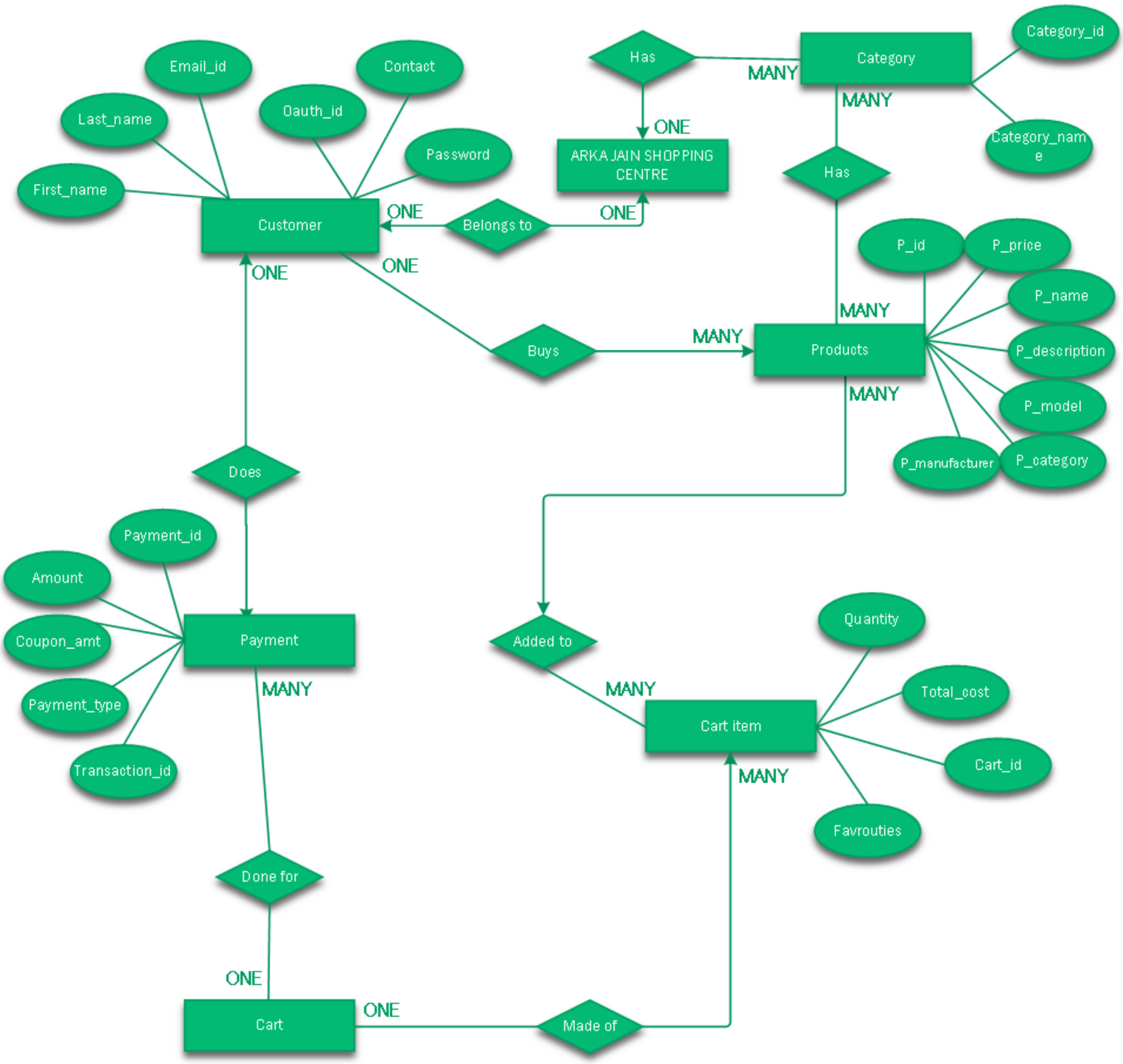


Fig 3.0

6.3. Entity Relationship

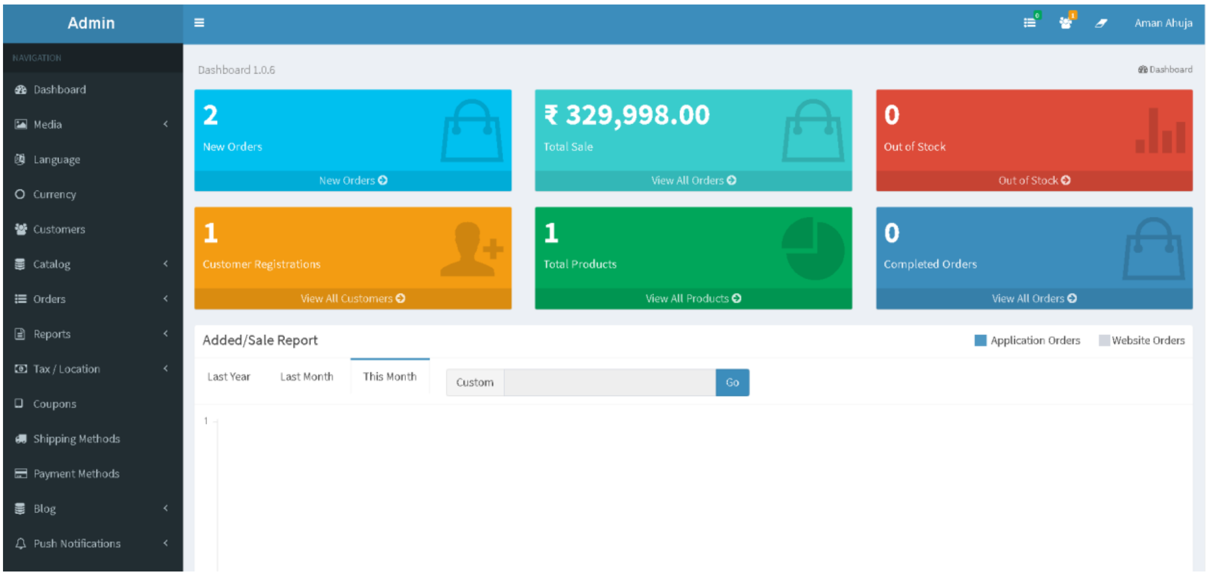
Entity Relationship For SHOPPING CENTRE



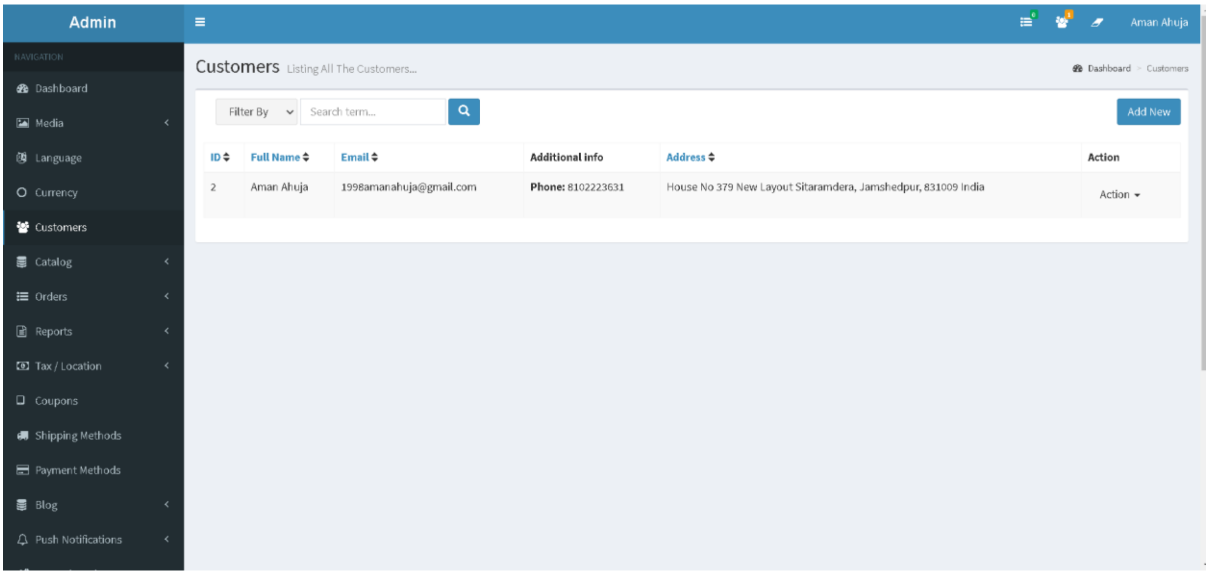
Made by Aman & Akash
ARKA Jain University

SCREENSHOTS

7.1. ADMIN

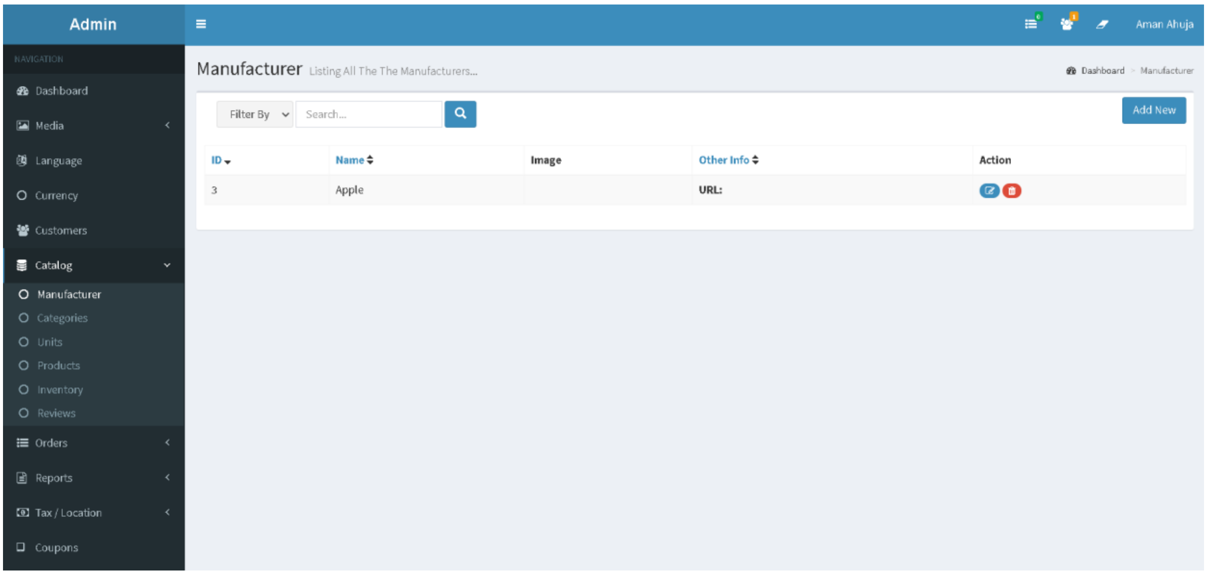


Dashboard



ID	Full Name	Email	Additional info	Address	Action
2	Aman Ahuja	1998amanahuja@gmail.com	**Phone:** 8102223631	House No 379 New Layout Sitaramdera, Jamshedpur, 831009 India	Action

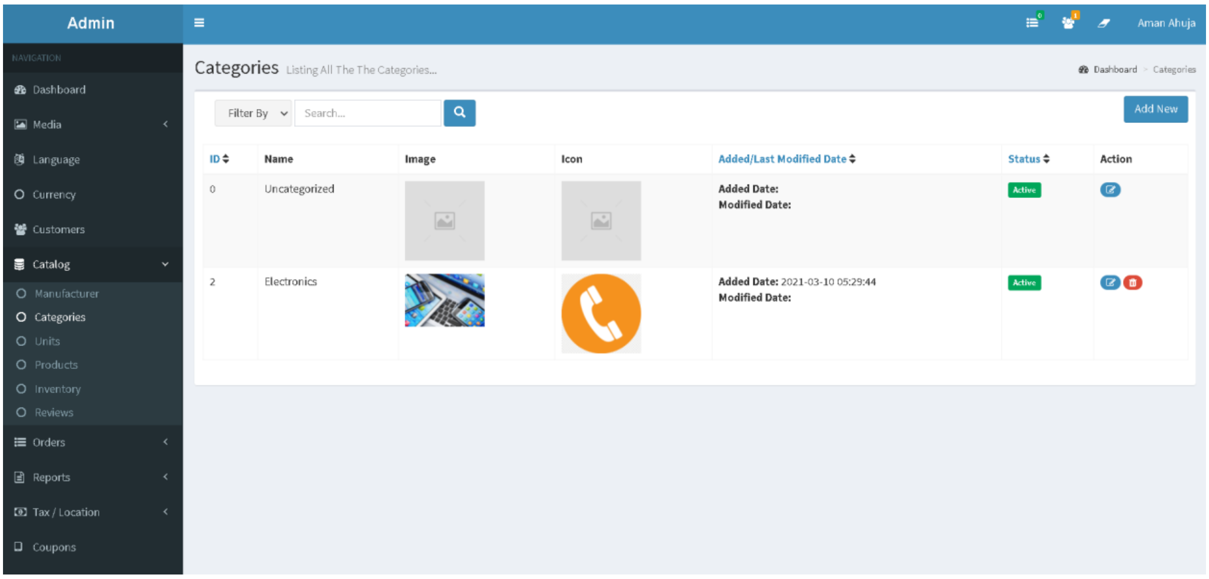
Users



Manufacturer Listing All The The Manufacturers...

ID	Name	Image	Other Info	Action
3	Apple		URL:	

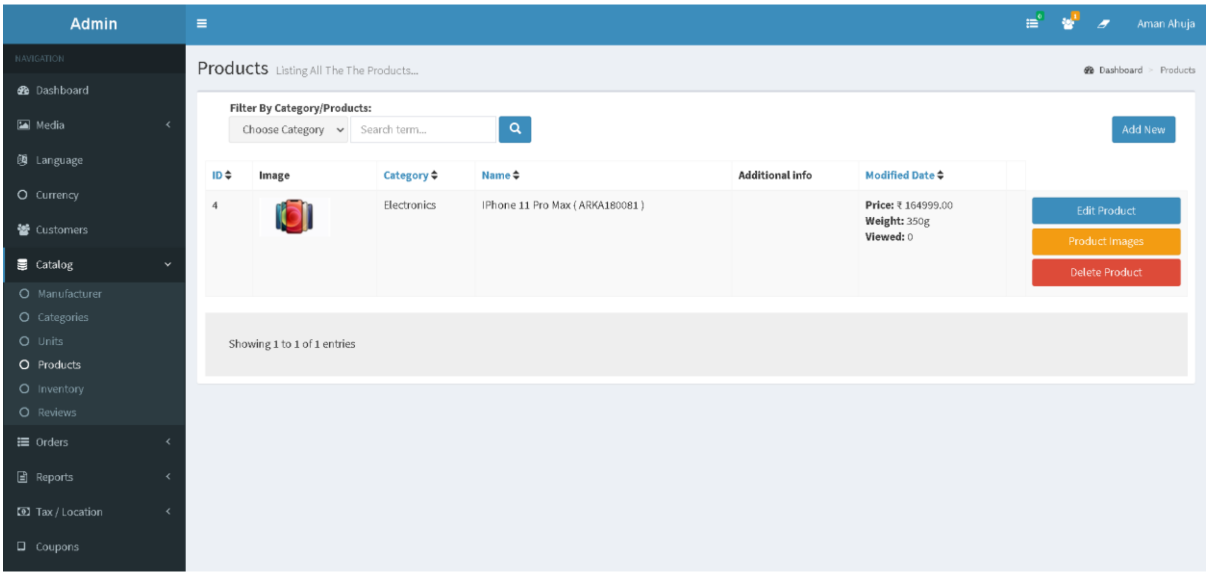
Manufacturer



Categories Listing All The The Categories...

ID	Name	Image	Icon	Added/Last Modified Date	Status	Action
0	Uncategorized			**Added Date:** **Modified Date:**	Active	
2	Electronics			**Added Date:** 2021-03-10 05:29:44 **Modified Date:**	Active	

Categories



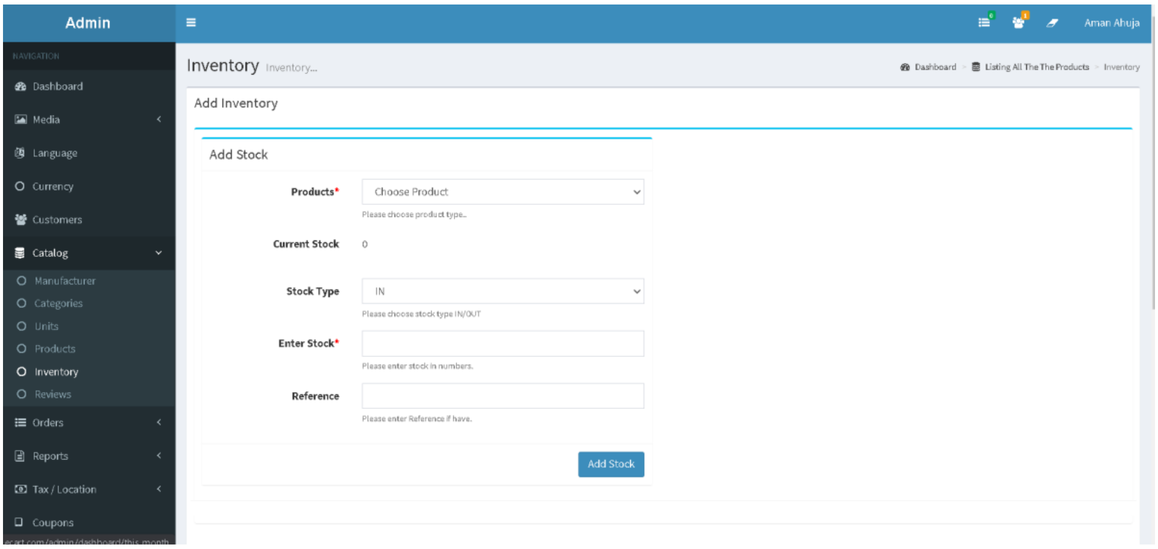
Products Listing All The The Products...

Filter By Category/Products: Choose Category | Search term...

ID	Image	Category	Name	Additional info	Modified Date	
4		Electronics	IPhone 11 Pro Max (ARKA180081)		**Price:** ₹ 164999.00 **Weight:** 350g **Viewed:** 0	

Showing 1 to 1 of 1 entries

Products



Inventory Inventory...

Add Inventory

Add Stock

Products*: Choose Product
Please choose product type..

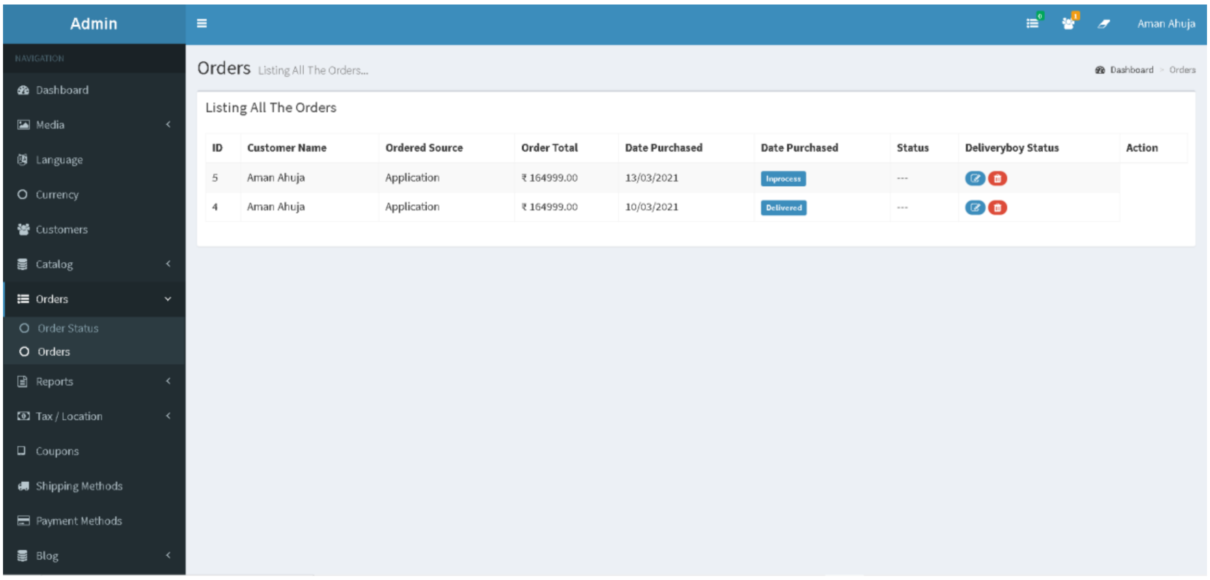
Current Stock: 0

Stock Type: IN
Please choose stock type IN/OUT

Enter Stock*:
Please enter stock in numbers.

Reference:
Please enter Reference if have.

Inventory

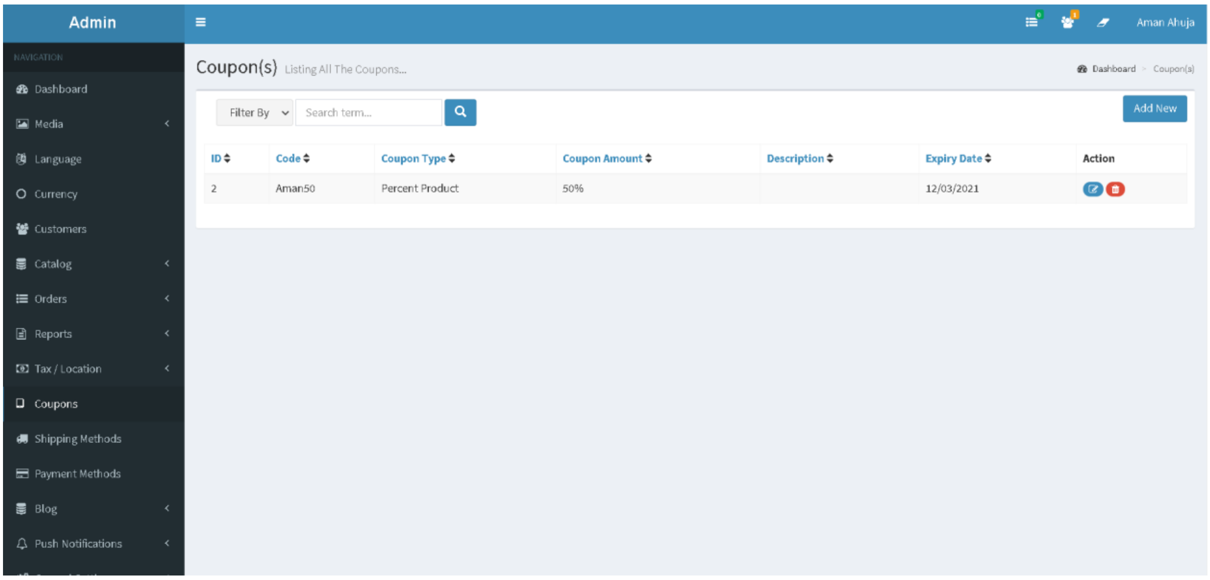


Orders Listing All The Orders...

Listing All The Orders

ID	Customer Name	Ordered Source	Order Total	Date Purchased	Date Purchased	Status	Deliveryboy Status	Action
5	Aman Ahuja	Application	₹ 164999.00	13/03/2021	Inprocess	---		
4	Aman Ahuja	Application	₹ 164999.00	10/03/2021	Delivered	---		

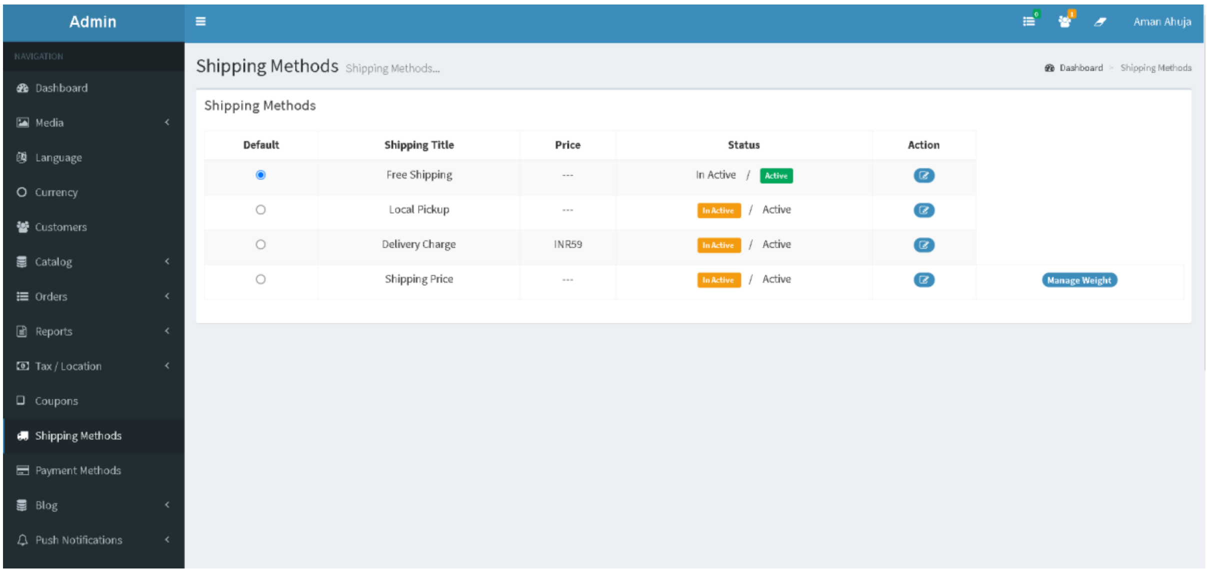
Orders



Coupon(s) Listing All The Coupons...

ID	Code	Coupon Type	Coupon Amount	Description	Expiry Date	Action
2	Aman50	Percent Product	50%		12/03/2021	

Coupons

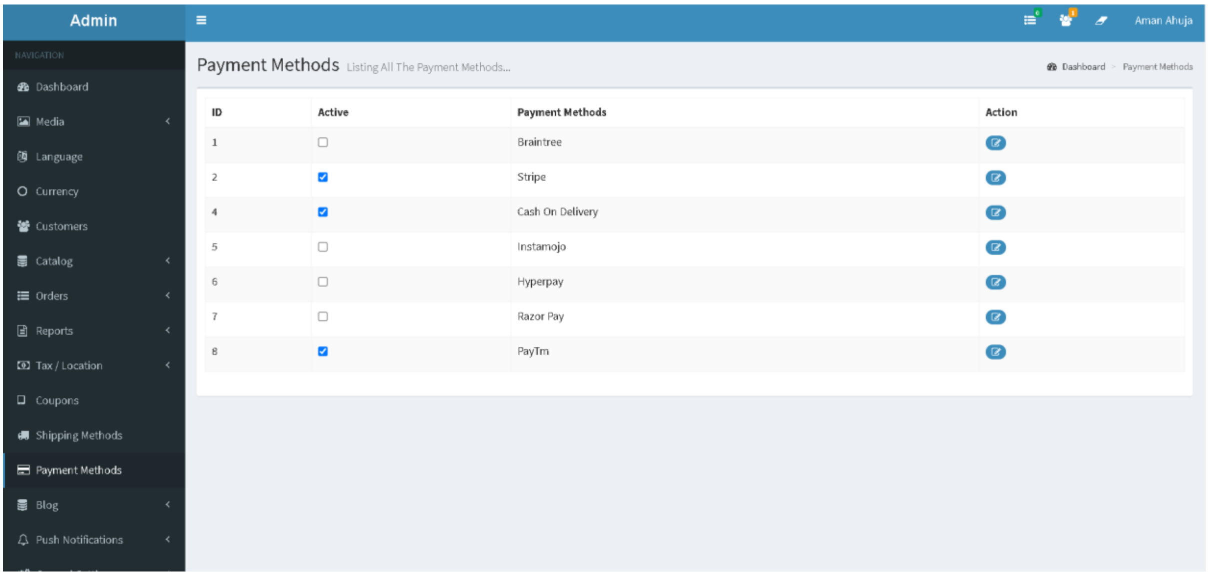


Shipping Methods Shipping Methods...

Shipping Methods

Default	Shipping Title	Price	Status	Action	
●	Free Shipping	---	In Active / Active	✎	
○	Local Pickup	---	In Active / Active	✎	
○	Delivery Charge	INR59	In Active / Active	✎	
○	Shipping Price	---	In Active / Active	✎	Manage Weight

Shipping Methods

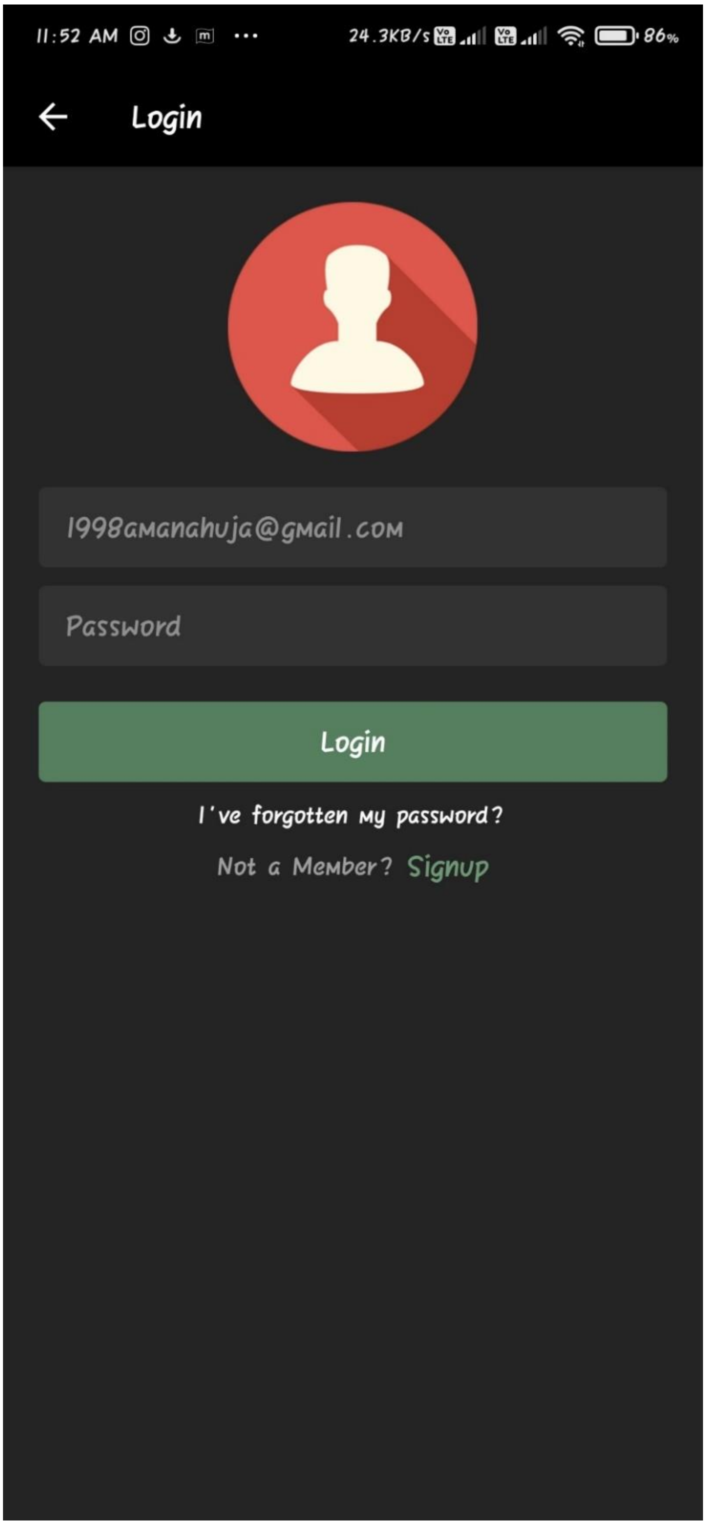


Payment Methods Listing All The Payment Methods...

ID	Active	Payment Methods	Action
1	☐	Braintree	✎
2	☑	Stripe	✎
4	☑	Cash On Delivery	✎
5	☐	Instamojo	✎
6	☐	Hyperpay	✎
7	☐	Razor Pay	✎
8	☑	PayTm	✎

Payment's method

7.2. User



11:52 AM

24.3KB/s 86%

Login

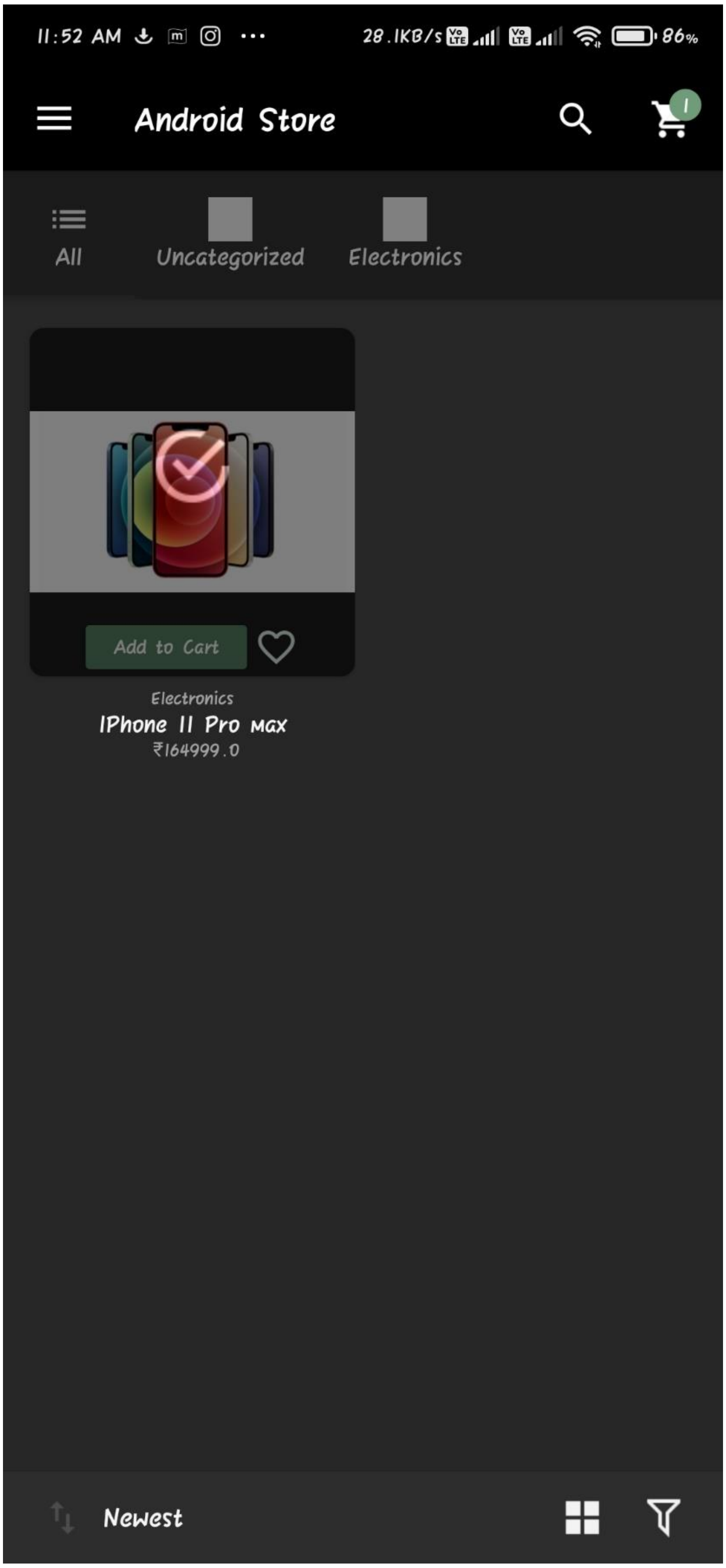
1998amanahuja@gmail.com

Password

I've forgotten my password?

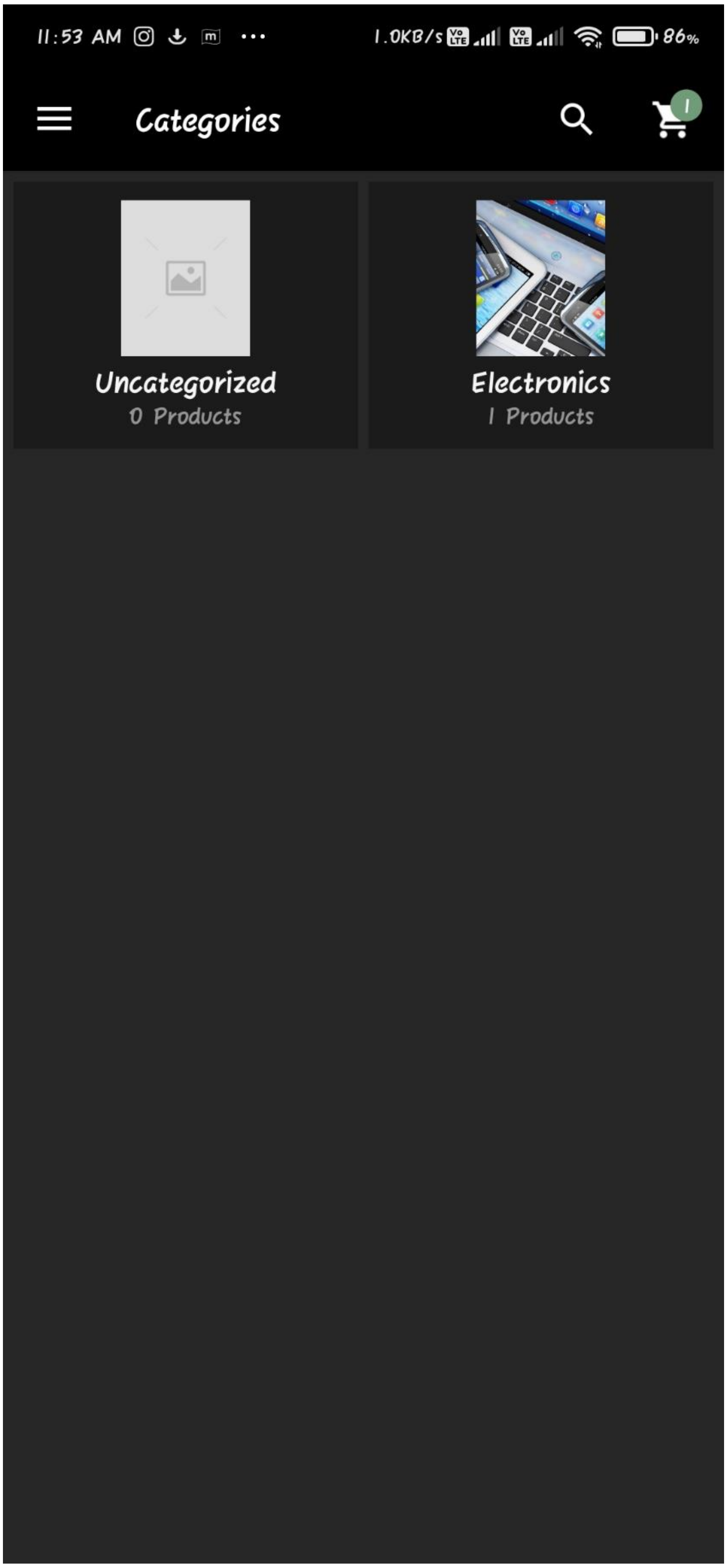
Not a Member? Signup

Login

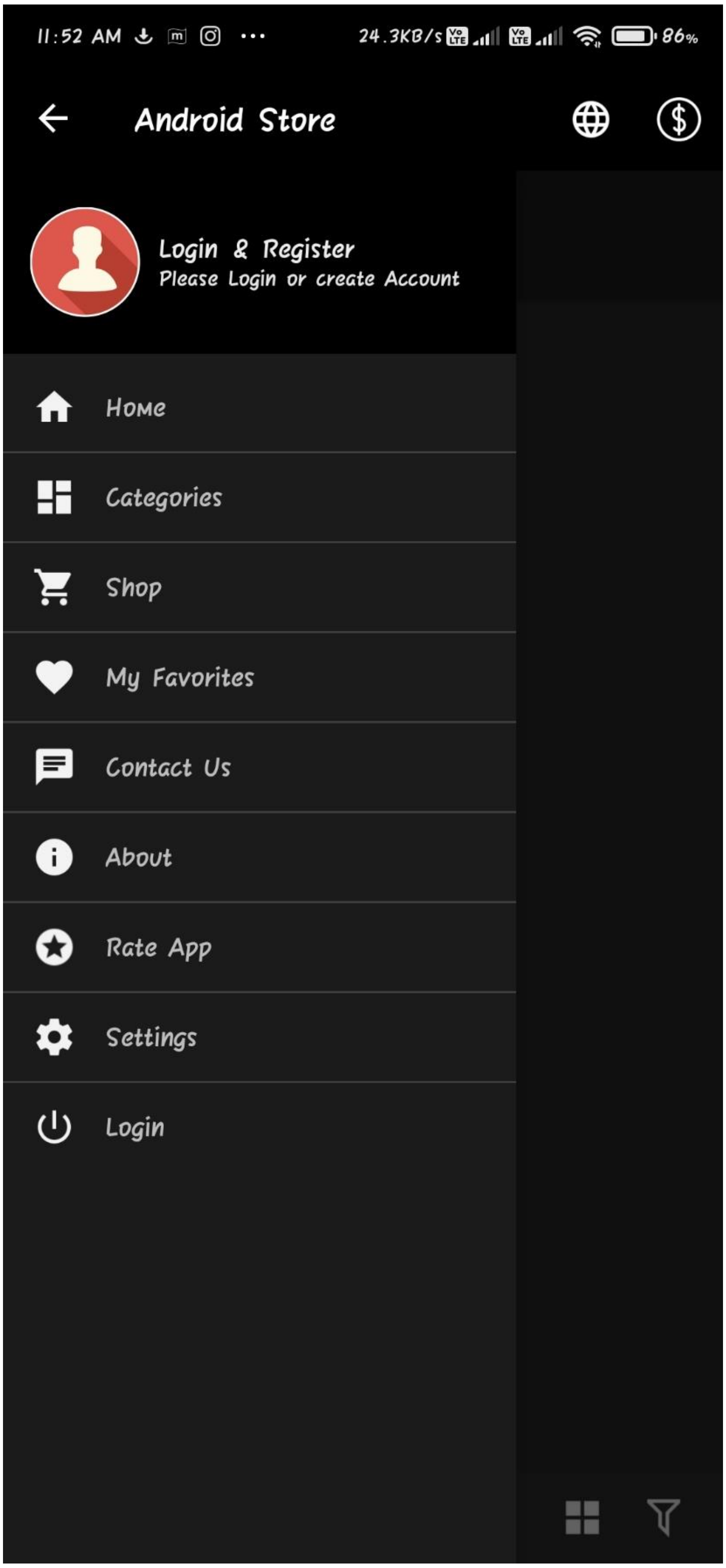


11:52 AM
28.1KB/s
86%
Android Store
1
All
Uncategorized
Electronics
Add to Cart
Electronics
IPhone 11 Pro Max
₹164999.0
Newest

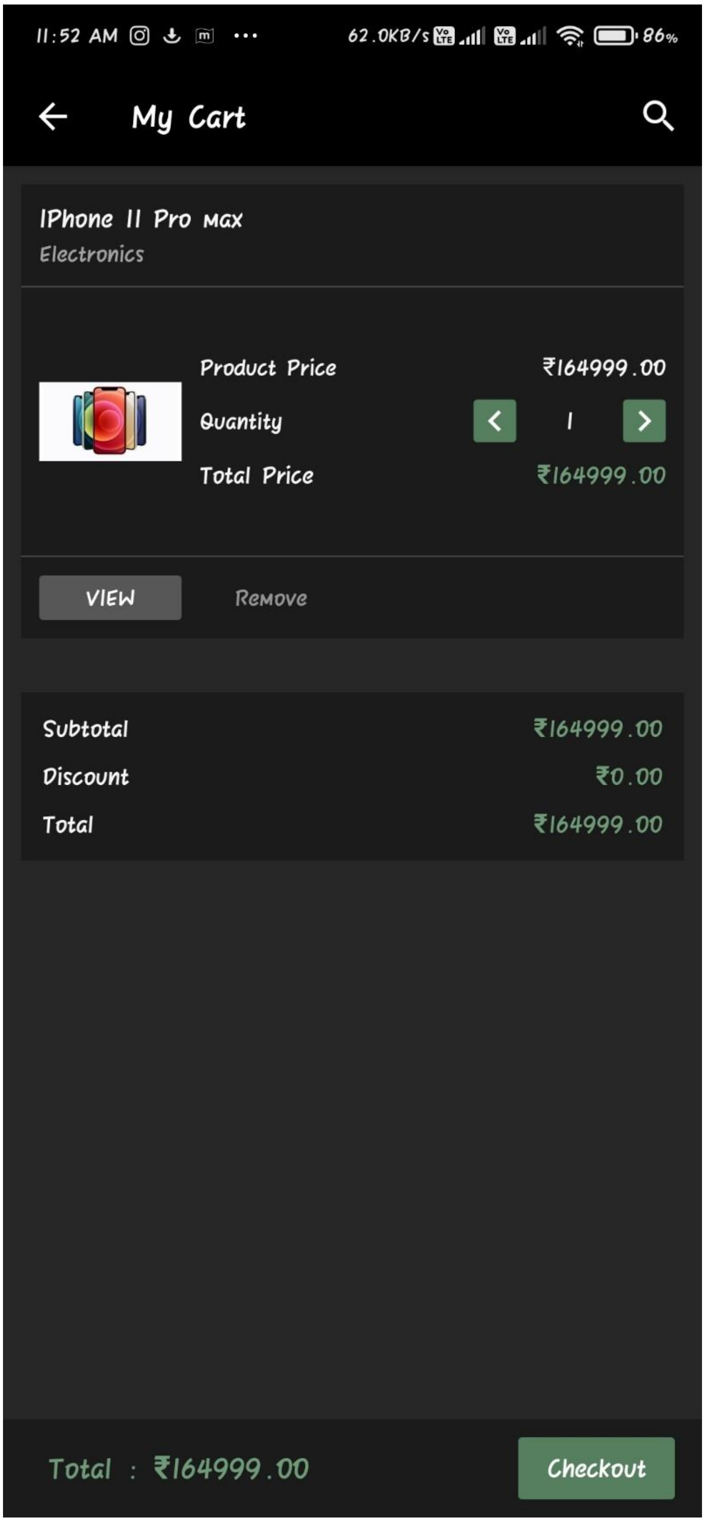
Home


11:53 AM
1.0KB/s 86%
Categories
1
Uncategorized
0 Products
Electronics
1 Products

Categories



All Pages Sidebar



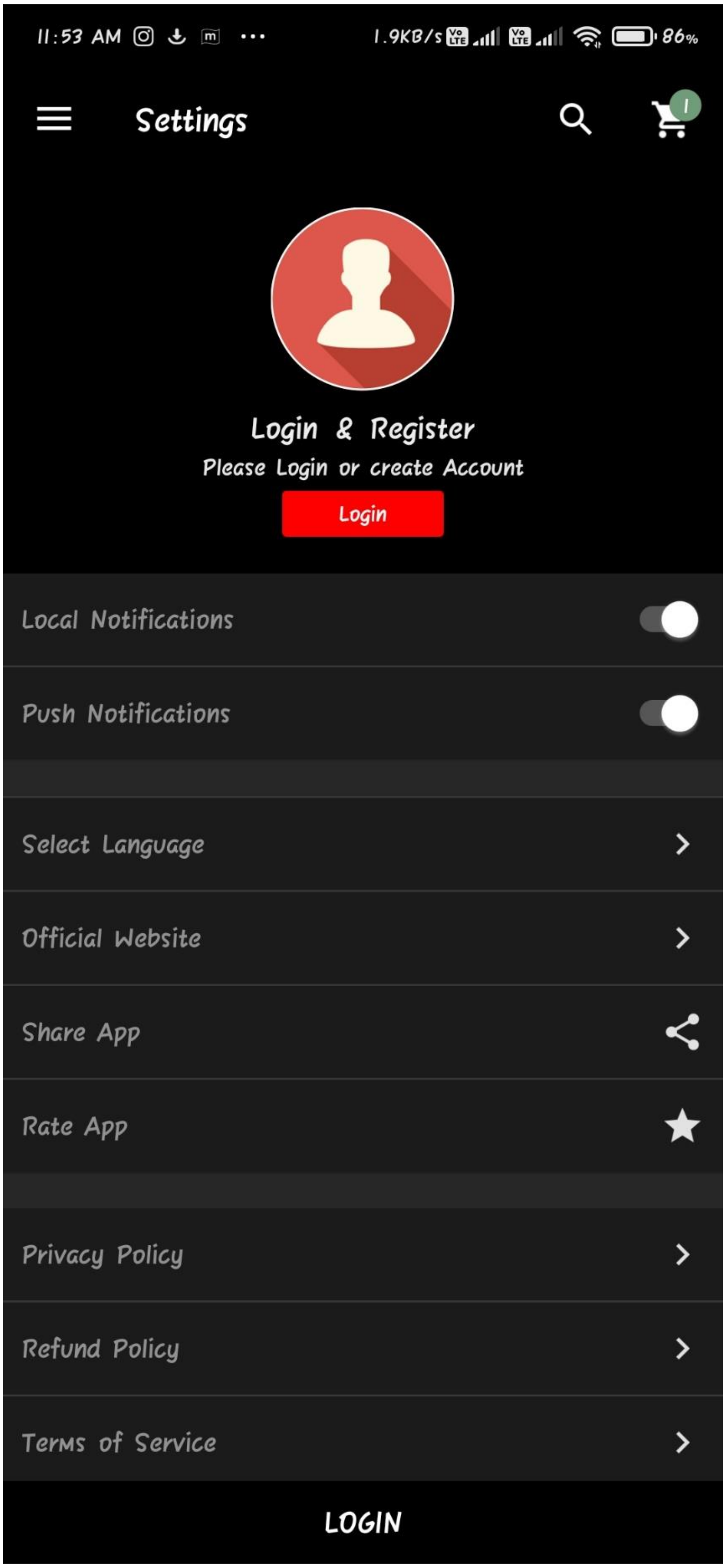
IPhone 11 Pro max
Electronics

Product Price	₹164999.00
Quantity	1
Total Price	₹164999.00

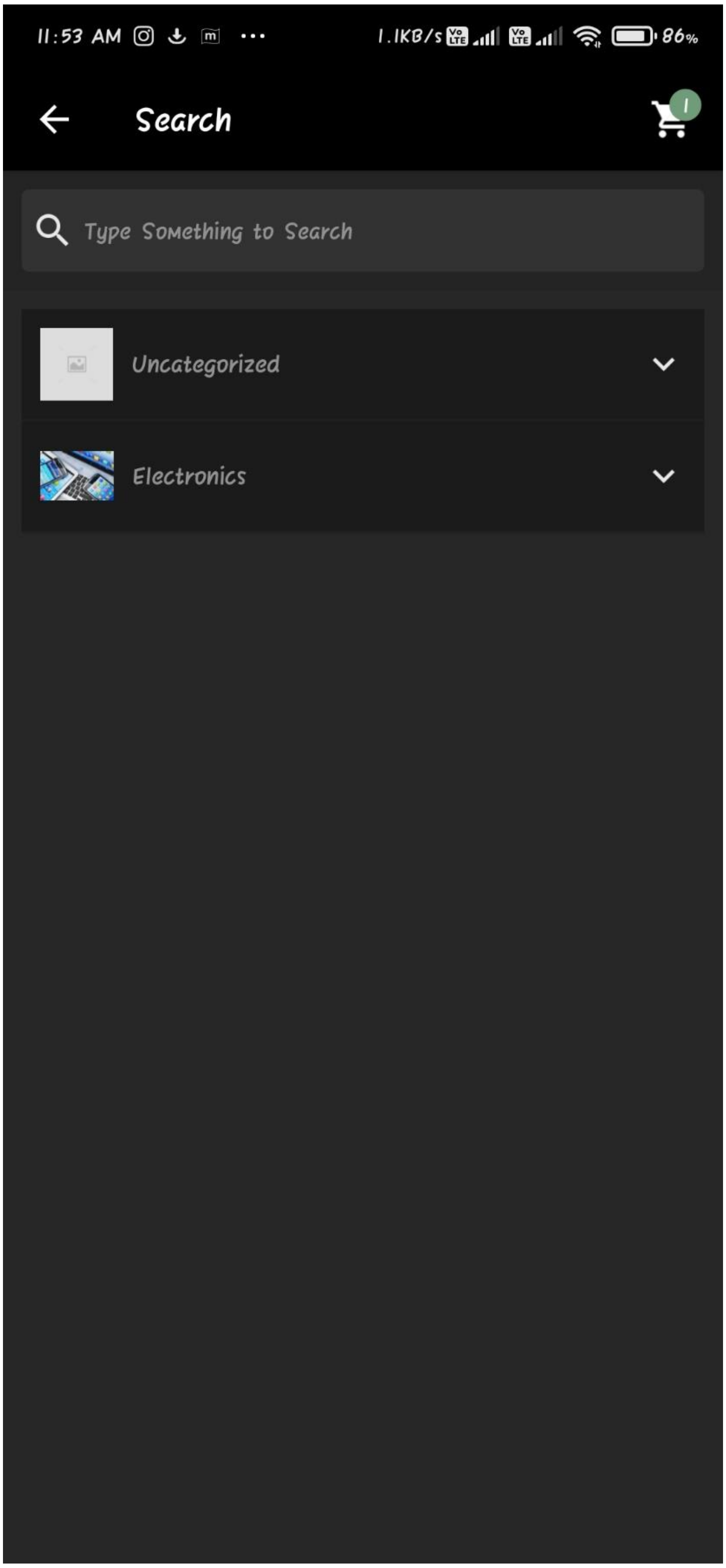
Subtotal	₹164999.00
Discount	₹0.00
Total	₹164999.00

Total : ₹164999.00

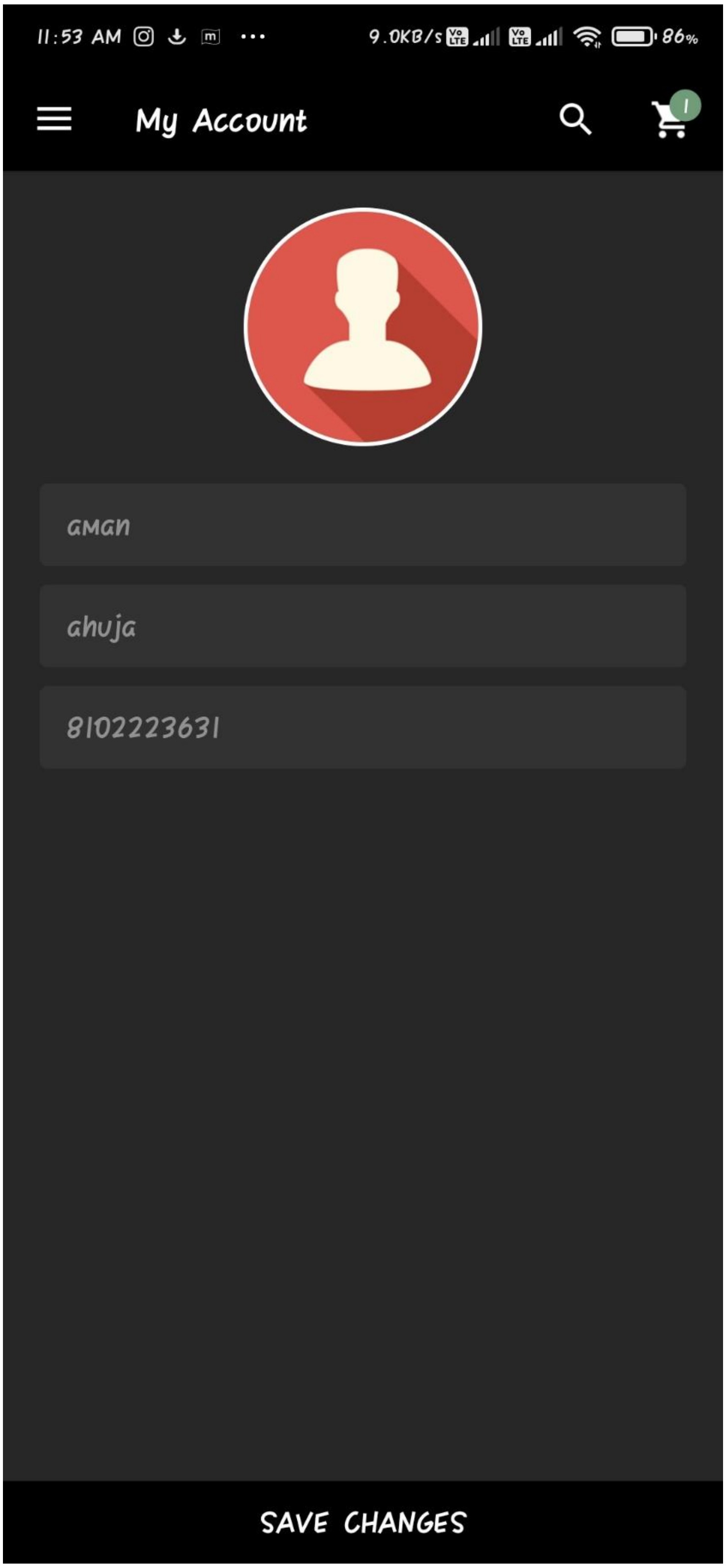
Cart


11:53 AM
1.9KB/s 86%
Settings
1
Login & Register
Please Login or create Account
Login
Local Notifications
Push Notifications
Select Language
Official Website
Share App
Rate App
Privacy Policy
Refund Policy
Terms of Service
LOGIN

Settings


11:53 AM
1.1KB/s
86%
Search
1
Type Something to Search
Uncategorized
Electronics

Search

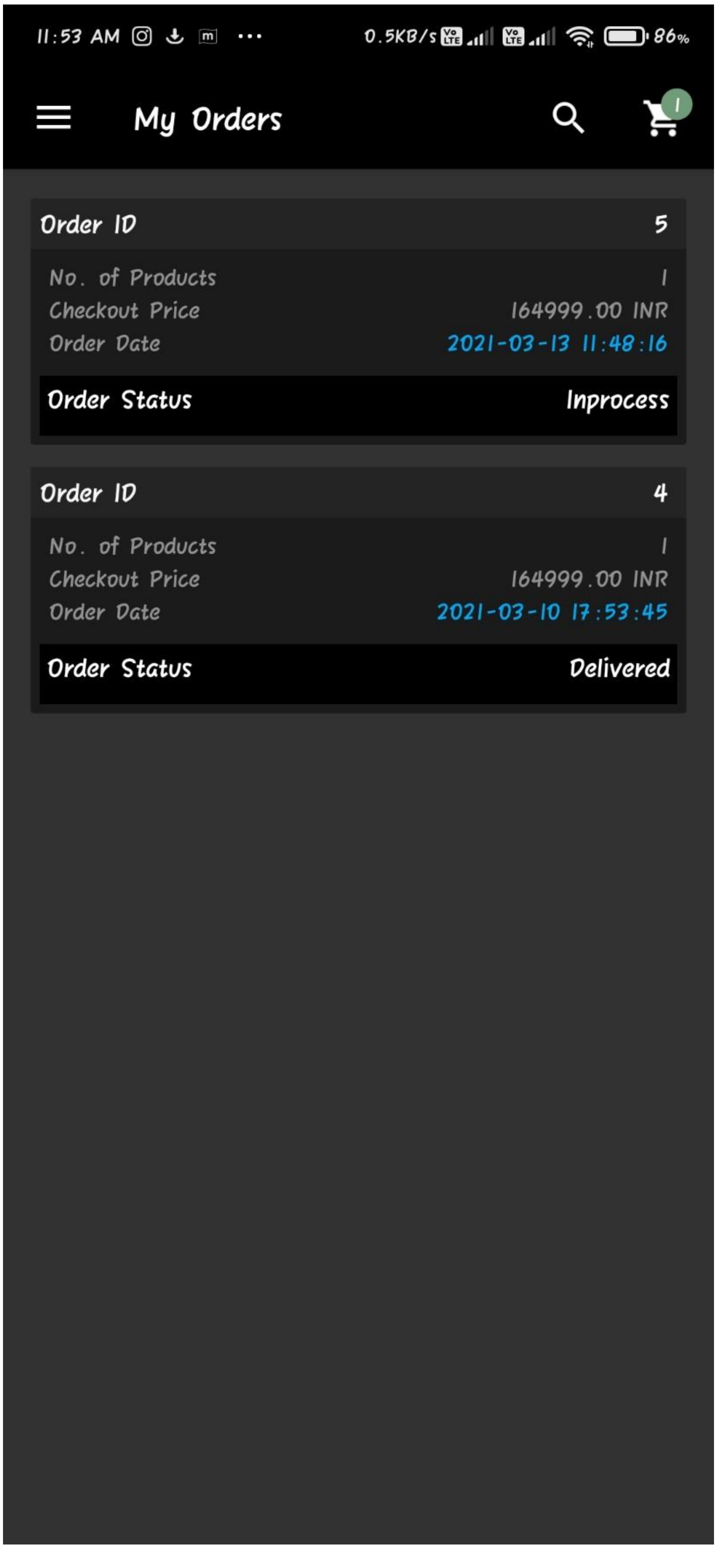


aman

ahuja

8102223631

Update Account



Order ID 5

No. of Products 1

Checkout Price 164999.00 INR

Order Date 2021-03-13 11:48:16

Order Status Inprocess

Order ID 4

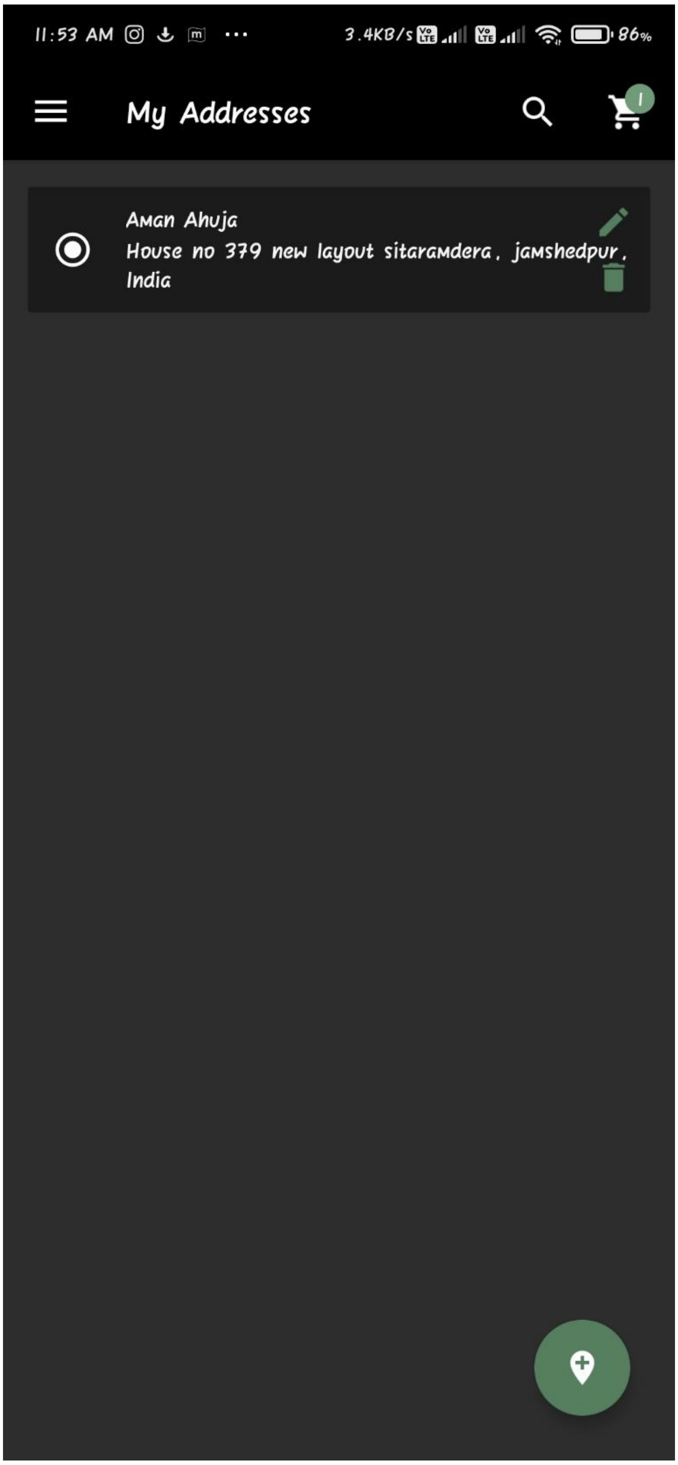
No. of Products 1

Checkout Price 164999.00 INR

Order Date 2021-03-10 17:53:45

Order Status Delivered

My orders



Aman Ahuja
House no 379 new layout sitaramdera, jamshedpur, India

My Address

7.3. Source Code

MainActivity.java

```java
public class IntroScreen extends AppIntro {

    MyAppPrefsManager myAppPrefsManager;


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setTheme(ConstantValues.THEME_ID);
        NoInternetDialog noInternetDialog = new NoInternetDialog.Builder(IntroScreen.this).build();
        //noInternetDialog.show();

        myAppPrefsManager = new MyAppPrefsManager(IntroScreen.this);


        addSlide(new Intro_Slide_1());
        addSlide(new Intro_Slide_2());
        addSlide(new Intro_Slide_3());
        addSlide(new Intro_Slide_4());
        addSlide(new Intro_Slide_5());


        // Hide StatusBar
        showStatusBar(false);
        showSkipButton(true);
        setProgressButtonEnabled(true);

        setBarColor(ContextCompat.getColor(IntroScreen.this, R.color.white));
        setSeparatorColor(ContextCompat.getColor(IntroScreen.this, R.color.colorPrimaryLight));

        setColorDoneText(ContextCompat.getColor(IntroScreen.this, R.color.colorPrimary));
        setColorSkipButton(ContextCompat.getColor(IntroScreen.this, R.color.colorPrimary));
        setNextArrowColor(ContextCompat.getColor(IntroScreen.this, R.color.colorPrimary));

        setIndicatorColor(ContextCompat.getColor(IntroScreen.this, R.color.colorPrimary),
                ContextCompat.getColor(IntroScreen.this, R.color.iconsLight));


        // Set Slide Change Animations
//        setFadeAnimation();
//        setZoomAnimation();
//        setFlowAnimation();
//        setDepthAnimation();
//        setSlideOverAnimation();
```

```
}



//*********** Called when the Skip Button pressed on IntroScreen ********//

@Override
public void onSkipPressed(Fragment currentFragment) {
    super.onSkipPressed(currentFragment);

    if (myAppPrefsManager.isFirstTimeLaunch()) {
        // Navigate to MainActivity
        startActivity(new Intent(IntroScreen.this, MainActivity.class));
        finish();
        overridePendingTransition(R.anim.enter_from_right, R.anim.exit_out_right);
    }
    else {
        // Finish this Activity
        finish();
    }

}



//*********** Called when the Done Button pressed on IntroScreen ********//

@Override
public void onDonePressed(Fragment currentFragment) {
    super.onDonePressed(currentFragment);

    if (myAppPrefsManager.isFirstTimeLaunch()) {
        // Navigate to MainActivity
        startActivity(new Intent(IntroScreen.this, MainActivity.class));
        finish();
        overridePendingTransition(R.anim.enter_from_right, R.anim.exit_out_right);
    }
    else {
        // Finish this Activity
        finish();
    }
}



//*********** Called when the active Slide Changes ********//

@Override
public void onSlideChanged(@Nullable Fragment oldFragment, @Nullable Fragment
newFragment) {
    super.onSlideChanged(oldFragment, newFragment);
    // Do something when the slide changes.
}



//*********** Set the Base Context for the ContextWrapper ********//

@Override
protected void attachBaseContext(Context newBase) {

    String languageCode = ConstantValues.LANGUAGE_CODE;
    if ("".equalsIgnoreCase(languageCode))
        languageCode = ConstantValues.LANGUAGE_CODE = "en";

    super.attachBaseContext(LocaleHelper.wrapLocale(newBase, languageCode));
}



//*********** Called when the Activity has detected the User pressed the Back key ********//

@Override
public void onBackPressed() {
    super.onBackPressed();

    if (myAppPrefsManager.isFirstTimeLaunch()) {
        // Navigate to MainActivity
        startActivity(new Intent(IntroScreen.this, MainActivity.class));
        finish();
        overridePendingTransition(R.anim.enter_from_right, R.anim.exit_out_right);
    }
    else {
        // Finish this Activity
        finish();
    }

}

}
```

Login.java

```
import android.app.Activity;
import android.app.AlertDialog;
import android.app.Dialog;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.text.TextUtils;
import android.util.Log;
import android.view.MenuItem;
import android.view.View;
import android.view.Window;
import android.view.inputmethod.InputMethodManager;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.TextView;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.appcompat.app.ActionBar;
import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.AppCompatButton;
import androidx.appcompat.widget.Toolbar;

import com.amanakashcorporatio.ecomstore.R;
import com.facebook.AccessToken;
import com.facebook.CallbackManager;
import com.facebook.FacebookCallback;
import com.facebook.FacebookException;
import com.facebook.FacebookSdk;
import com.facebook.login.LoginManager;
import com.facebook.login.LoginResult;
import com.google.android.gms.auth.api.Auth;
import com.google.android.gms.auth.api.signin.GoogleSignInAccount;
import com.google.android.gms.auth.api.signin.GoogleSignInOptions;
import com.google.android.gms.auth.api.signin.GoogleSignInResult;
import com.google.android.gms.common.ConnectionResult;
import com.google.android.gms.common.api.GoogleApiClient;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.android.gms.tasks.TaskExecutors;
import com.google.android.material.snackbar.Snackbar;
import com.google.firebase.FirebaseException;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseAuthInvalidCredentialsException;
import com.google.firebase.auth.PhoneAuthCredential;
import com.google.firebase.auth.PhoneAuthProvider;
import com.google.gson.Gson;
import com.amanakash.ecommerce.app.MyAppPrefsManager;
import com.amanakash.ecommerce.constant.ConstantValues;
import com.amanakash.ecommerce.customs.DialogLoader;
import com.amanakash.ecommerce.databases.User_Info_DB;
import com.amanakash.ecommerce.models.user_model.UserData;
import com.amanakash.ecommerce.models.user_model.UserDetails;
import com.amanakash.ecommerce.network.APIClient;
import com.amanakash.ecommerce.network.StartAppRequests;
import com.amanakash.ecommerce.utils.LocaleHelper;
import com.amanakash.ecommerce.utils.ValidateInputs;

import java.util.Arrays;
import java.util.concurrent.TimeUnit;

import am.appwise.components.ni.NoInternetDialog;
import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

/**
 * Login activity handles User's Email, Facebook and Google Login
 **/


public class Login extends AppCompatActivity implements
GoogleApiClient.OnConnectionFailedListener {

    View parentView;
    Toolbar toolbar;
    ActionBar actionBar;

    EditText user_email, user_password;
    TextView forgotPasswordText, signupText;
    Button loginBtn, facebookLoginBtn, googleLoginBtn, phoneLoginBtn;
    LinearLayout layoutEmailLogin;

    User_Info_DB userInfoDB;
    DialogLoader dialogLoader;

    SharedPreferences.Editor editor;
    SharedPreferences sharedPreferences;

    private CallbackManager callbackManager;

    private GoogleApiClient mGoogleApiClient;
    private GoogleSignInOptions mGoogleSignInOptions;

    private static final int RC_SIGN_IN = 100;


    private UserDetails userDetails;
    private Dialog dialogOTP;
    private FirebaseAuth mAuth;
    EditText ed_otp;
    private int TEMP_USER_TYPE;
    private static String user_current_phone_number;

    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setTheme(ConstantValues.THEME_ID);
        NoInternetDialog noInternetDialog = new
         NoInternetDialog.Builder(Login.this).build();
        // noInternetDialog.show();

        mAuth = FirebaseAuth.getInstance();

        // Initialize Facebook SDk for Facebook Login
        FacebookSdk.sdkInitialize(getApplicationContext());

        // Initializing Google SDK for Google Login
        mGoogleSignInOptions = new
```

```
GoogleSignInOptions.Builder(GoogleSignInOptions.DEFAULT_SIGN_IN)
        .requestIdToken(getString(R.string.client_id))
        .requestEmail()
        .build();


setContentView(R.layout.login);


// setting Toolbar
toolbar = (Toolbar) findViewById(R.id.myToolbar);
setSupportActionBar(toolbar);
actionBar = getSupportActionBar();
actionBar.setTitle(getString(R.string.login));
actionBar.setDisplayShowHomeEnabled(true);
actionBar.setDisplayHomeAsUpEnabled(true);


// Binding Layout Views
user_email = (EditText) findViewById(R.id.user_email);
user_password = (EditText) findViewById(R.id.user_password);
loginBtn = (Button) findViewById(R.id.loginBtn);
facebookLoginBtn = (Button) findViewById(R.id.facebookLoginBtn);
googleLoginBtn = (Button) findViewById(R.id.googleLoginBtn);
phoneLoginBtn = (Button) findViewById(R.id.phoneLoginBtn);
signupText = (TextView) findViewById(R.id.login_signupText);
forgotPasswordText = (TextView) findViewById(R.id.forgot_password_text);
layoutEmailLogin = (LinearLayout) findViewById(R.id.layout_email_login);

parentView = signupText;


if (ConstantValues.IS_GOOGLE_LOGIN_ENABLED) {
    googleLoginBtn.setVisibility(View.VISIBLE);
} else {
    googleLoginBtn.setVisibility(View.GONE);
}

if (ConstantValues.IS_FACEBOOK_LOGIN_ENABLED) {
    facebookLoginBtn.setVisibility(View.VISIBLE);
} else {
    facebookLoginBtn.setVisibility(View.GONE);
}

if (ConstantValues.IS_PHONE_LOGIN_ENABLED) {
    phoneLoginBtn.setVisibility(View.VISIBLE);
} else {
    phoneLoginBtn.setVisibility(View.GONE);
}

if (ConstantValues.IS_EMAIL_LOGIN_ENABLED) {
    layoutEmailLogin.setVisibility(View.VISIBLE);
} else {
    layoutEmailLogin.setVisibility(View.GONE);
}

dialogLoader = new DialogLoader(Login.this);
```

```
        userInfoDB = new User_Info_DB();
        sharedPreferences = getSharedPreferences("UserInfo", MODE_PRIVATE);


        user_email.setText(sharedPreferences.getString("userEmail", null));


        // Register Callback for Facebook LoginManager
        callbackManager = CallbackManager.Factory.create();
        LoginManager.getInstance().registerCallback(callbackManager, new
                 FacebookCallback<LoginResult>() {
            @Override
            public void onSuccess(LoginResult loginResult) {

                // Get Access Token and proceed Facebook Registration
                String accessToken = loginResult.getAccessToken().getToken();
                processFacebookRegistration(accessToken);

            }

            @Override
            public void onCancel() {
                // If User Canceled
            }

            @Override
            public void onError(FacebookException e) {
                // If Login Fails
                Toast.makeText(Login.this, "FacebookException : " + e,
        Toast.LENGTH_LONG).show();
            }
        });
    }
```

Singup.java

```
import android.Manifest;
import android.app.Activity;
import android.app.DatePickerDialog;
import android.app.Dialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.database.Cursor;
import android.graphics.Bitmap;
import android.graphics.Color;
import android.net.Uri;
import android.os.Build;
import android.os.Bundle;
import android.provider.MediaStore;
import android.text.TextUtils;
import android.view.MenuItem;
import android.view.MotionEvent;
import android.view.View;
import android.view.Window;
import android.view.WindowManager;
import android.view.inputmethod.InputMethodManager;
import android.webkit.WebSettings;
import android.webkit.WebView;
import android.widget.Button;
import android.widget.DatePicker;
import android.widget.EditText;
import android.widget.FrameLayout;
import android.widget.ImageButton;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.appcompat.app.ActionBar;
import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.AppCompatButton;
import androidx.appcompat.widget.Toolbar;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

import com.amanakashcorporatio.ecomstore.R;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.android.gms.tasks.TaskExecutors;
import com.google.android.material.snackbar.Snackbar;
import com.google.firebase.FirebaseException;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseAuthInvalidCredentialsException;
import com.google.firebase.auth.PhoneAuthCredential;
import com.google.firebase.auth.PhoneAuthProvider;
import com.amanakash.ecommerce.constant.ConstantValues;
import com.amanakash.ecommerce.customs.CircularImageView;
import com.amanakash.ecommerce.customs.DialogLoader;
import com.amanakash.ecommerce.models.user_model.UserData;
import com.amanakash.ecommerce.network.APIClient;
import com.amanakash.ecommerce.utils.CheckPermissions;
import com.amanakash.ecommerce.utils.ImagePicker;
import com.amanakash.ecommerce.utils.LocaleHelper;
import com.amanakash.ecommerce.utils.ValidateInputs;

import java.io.ByteArrayOutputStream;
import java.io.File;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Locale;
```

```java
import java.util.concurrent.TimeUnit;

import am.appwise.components.ni.NoInternetDialog;
import okhttp3.MediaType;
import okhttp3.RequestBody;
import retrofit2.Call;
import retrofit2.Callback;

/*
import com.google.android.gms.ads.AdListener;
import com.google.android.gms.ads.AdRequest;
import com.google.android.gms.ads.AdSize;
import com.google.android.gms.ads.AdView;
import com.google.android.gms.ads.MobileAds;
*/



/**
 * SignUp activity handles User's Registration
 **/

public class Signup extends AppCompatActivity {

    View parentView;
    File profileImage;
    private static final int PICK_IMAGE_ID = 360;        // the number doesn't matter

    Toolbar toolbar;
    ActionBar actionBar;

    DialogLoader dialogLoader;

    //AdView mAdView;
    Button signupBtn;
    FrameLayout banner_adView;
    TextView signup_loginText;
    TextView service_terms, privacy_policy, refund_policy, and_text;
    CircularImageView user_photo;
    //FloatingActionButton user_photo_edit_fab;
    EditText user_firstname, user_lastname, user_email, user_password, user_mobile, input_dob;
    Spinner genderSpinner;

    //Custom Dialog Vies
    Dialog dialogOTP;
    EditText ed_otp;
    //It is the verification id that will be sent to the user
    private String mVerificationId;
    //firebase auth object
    private FirebaseAuth mAuth;
```

```
@Override
protected void onCreate(@Nullable Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setTheme(ConstantValues.THEME_ID);
    setContentView(R.layout.signup);

    //MobileAds.initialize(this, ConstantValues.ADMOBE_ID);

    NoInternetDialog noInternetDialog = new NoInternetDialog.Builder(Signup.this).build();
    //noInternetDialog.show();

    //initializing objects
    mAuth = FirebaseAuth.getInstance();
    // setting Toolbar
    toolbar = (Toolbar) findViewById(R.id.myToolbar);
    setSupportActionBar(toolbar);
    actionBar = getSupportActionBar();
    actionBar.setTitle(getString(R.string.signup));
    actionBar.setDisplayHomeAsUpEnabled(true);


    // Binding Layout Views
    user_photo = (CircularImageView) findViewById(R.id.user_photo);
    user_firstname = (EditText) findViewById(R.id.user_firstname);
    user_lastname = (EditText) findViewById(R.id.user_lastname);
    user_email = (EditText) findViewById(R.id.user_email);
    user_password = (EditText) findViewById(R.id.user_password);
    user_mobile = (EditText) findViewById(R.id.user_mobile);
    signupBtn = (Button) findViewById(R.id.signupBtn);
    and_text = (TextView) findViewById(R.id.and);
    service_terms = (TextView) findViewById(R.id.service_terms);
    privacy_policy = (TextView) findViewById(R.id.privacy_policy);
    refund_policy = (TextView) findViewById(R.id.refund_policy);
    signup_loginText = (TextView) findViewById(R.id.signup_loginText);
    banner_adView = (FrameLayout) findViewById(R.id.banner_adView);
    input_dob = (EditText) findViewById(R.id.et_dob);
    //user_photo_edit_fab = (FloatingActionButton) findViewById(R.id.user_photo_edit_fab);
    genderSpinner = findViewById(R.id.genderSpinner);
```

Order Tracking.java

```
import android.os.Bundle;
import android.widget.Toast;

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

import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.OnMapReadyCallback;
import com.google.android.gms.maps.SupportMapFragment;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.Marker;
import com.google.android.gms.maps.model.MarkerOptions;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;
import com.amanakashcorporatio.ecomstore.R;
import com.amanakash.ecommerce.models.order_model.LocationData;

public class Order_Tracking_Activity extends AppCompatActivity implements
OnMapReadyCallback {


    private GoogleMap mGoogleMap;
    Marker deliveryBoyMarker;

    String customerLatitude = "0.0";
    String customerLongitude = "0.0";
    String deliveryBoyName = "";
    String deliveryBoyID = "";

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

        SupportMapFragment supportMapFragment = (SupportMapFragment)
         getSupportFragmentManager().findFragmentById(R.id.order_tracking_map);
        assert supportMapFragment != null;
        supportMapFragment.getMapAsync(Order_Tracking_Activity.this);

        Bundle b = getIntent().getExtras();
        if (b != null) {
            customerLatitude = b.getString("Customer_Delivery_Lat");
            customerLongitude = b.getString("Customer_Delivery_Lng");
            deliveryBoyID = b.getString("Delivery_Boy_Pincode");
            deliveryBoyName = b.getString("Delivery_Boy_Name");
        }

    }

    @Override
    public void onMapReady(GoogleMap googleMap) {

        mGoogleMap = googleMap;

        mGoogleMap.setMapType(GoogleMap.MAP_TYPE_NORMAL);
        //googleMap.setMyLocationEnabled(true);
        mGoogleMap.setTrafficEnabled(false);
        mGoogleMap.setIndoorEnabled(false);
        mGoogleMap.setBuildingsEnabled(true);
        mGoogleMap.getUiSettings().setZoomControlsEnabled(true);
        mGoogleMap.getUiSettings().setZoomGesturesEnabled(true);

        if (!deliveryBoyID.isEmpty()) {
```

```
    MarkerOptions deliveryBoyMarkerOption = new MarkerOptions();
    deliveryBoyMarkerOption.position(new LatLng(0, 0));
    deliveryBoyMarkerOption.title(deliveryBoyName);
    deliveryBoyMarkerOption.snippet("Delivery Boy");
    deliveryBoyMarkerOption.draggable(false);

//deliveryBoyMarkerOption.icon(BitmapDescriptorFactory.fromResource(R.dra
wable.deliveryboymarker));
    deliveryBoyMarker = mGoogleMap.addMarker(deliveryBoyMarkerOption);

    DatabaseReference ref = FirebaseDatabase.getInstance().getReference().
         child("location").child(deliveryBoyID);

    ref.addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(DataSnapshot dataSnapshot) {
            if (dataSnapshot.exists()) {

                LocationData location =
dataSnapshot.getValue(LocationData.class);

                deliveryBoyMarker.setPosition(new
LatLng(location.getLatitude(), location.getLongitude()));


mGoogleMap.animateCamera(CameraUpdateFactory.newLatLngZoom(new
LatLng(location.getLatitude(), location.getLongitude()), 15));
            } else {
                Toast.makeText(Order_Tracking_Activity.this, "Delivery boy
details are missing", Toast.LENGTH_SHORT).show();
            }
        }

        @Override
        public void onCancelled(DatabaseError databaseError) {
            System.out.println("The read failed: " +
databaseError.getCode());
        }
    });
} else {
    Toast.makeText(this, "Your rider info is empty",
Toast.LENGTH_SHORT).show();
}


MarkerOptions vendorMarkerOpiton = new MarkerOptions();
vendorMarkerOpiton.position(
        new LatLng(
                Double.parseDouble(customerLatitude)
                , Double.parseDouble(customerLongitude)));
vendorMarkerOpiton.title("Customer Location");
vendorMarkerOpiton.snippet("Customer location");
vendorMarkerOpiton.draggable(false);

//vendorMarkerOpiton.icon(BitmapDescriptorFactory.fromResource(R.drawable
.vendormarker));
mGoogleMap.addMarker(vendorMarkerOpiton);
```

```
        mGoogleMap.moveCamera(CameraUpdateFactory.newLatLngZoom(new LatLng(
                Double.parseDouble(customerLatitude)
                , Double.parseDouble(customerLongitude)), 15));
        // Zoom in, animating the camera.
        mGoogleMap.animateCamera(CameraUpdateFactory.zoomIn());
        // Zoom out to zoom level 10, animating with a duration of 2 seconds.
        mGoogleMap.animateCamera(CameraUpdateFactory.zoomTo(15), 500, null);

    }
}
```

Card Item Adaptar.java

```
public class CartItemsAdapter extends
RecyclerView.Adapter<CartItemsAdapter.MyViewHolder> {

    private Context context;
    private My_Cart cartFragment;
    private List<CartProduct> cartItemsList = new ArrayList<>();

    private User_Cart_DB user_cart_db;
    private User_Recents_DB recents_db;

    private ProductAttributeValuesAdapter attributesAdapter;


    public CartItemsAdapter(Context context, List<CartProduct> cartItemsList,
         My_Cart cartFragment) {
        this.context = context;
        this.cartItemsList = cartItemsList;
        this.cartFragment = cartFragment;

        user_cart_db = new User_Cart_DB();
        recents_db = new User_Recents_DB();
    }



    //********** Called to Inflate a Layout from XML and then return the Holder
*********//

    @Override
    public MyViewHolder onCreateViewHolder(final ViewGroup parent, int viewType) {

        View itemView;

        if(viewType == R.layout.layout_button){
            itemView =
        LayoutInflater.from(parent.getContext()).inflate(R.layout.layout_button,
        parent, false);
        }
        else {
            itemView =
        LayoutInflater.from(parent.getContext()).inflate(R.layout.layout_card_car
        t_items, parent, false);
        }
        return new MyViewHolder(itemView);
    }



    //********** Called by RecyclerView to display the Data at the specified
        Position *********//

    @Override
    public void onBindViewHolder(final MyViewHolder holder, int position) {

        if (position == cartItemsList.size()) {
            holder.custom_button.setText(context.getString(R.string.explore));

            holder.custom_button.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View view) {

                    Bundle bundle = new Bundle();
                    bundle.putBoolean("isSubFragment", false);

                    // Navigate to Products Fragment
                    Fragment fragment = new Products();
                    fragment.setArguments(bundle);
                    FragmentManager fragmentManager = ((MainActivity)
        context).getSupportFragmentManager();
                    fragmentManager.beginTransaction()
                            .add(R.id.main_fragment, fragment)

        .setTransition(FragmentTransaction.TRANSIT_FRAGMENT_FADE)

        .addToBackStack(context.getString(R.string.actionCart)).commit();
                }
            });

        }
        else {
            // Get the data model based on Position
            final CartProduct cartProduct = cartItemsList.get(position);

            // Set Product Image on ImageView with Glide Library
            Glide.with(context).load(ConstantValues.ECOMMERCE_URL+
        cartProduct.getCustomersBasketProduct().getProductsImage()).into(holder.c
        art_item_cover);


        holder.cart_item_title.setText(cartProduct.getCustomersBasketProduct().ge
        tProductsName());

        holder.cart_item_category.setText(cartProduct.getCustomersBasketProduct()
        .getCategoryNames());
            holder.cart_item_quantity.setText("" +
        cartProduct.getCustomersBasketProduct().getCustomersBasketQuantity());
            holder.cart_item_base_price.setText(ConstantValues.CURRENCY_SYMBOL +
        new
        DecimalFormat("#0.00").format(Double.parseDouble(cartProduct.getCustomers
        BasketProduct().getProductsPrice())));
            holder.cart_item_sub_price.setText(ConstantValues.CURRENCY_SYMBOL +
        new
        DecimalFormat("#0.00").format(Double.parseDouble(cartProduct.getCustomers
        BasketProduct().getTotalPrice())));


            List<Value> selectedAttributeValues= new ArrayList<>();
            List<CartProductAttributes> productAttributes= new ArrayList<>();

            productAttributes = cartProduct.getCustomersBasketProductAttributes();

            for (int i=0;  i<productAttributes.size();  i++) {

        selectedAttributeValues.add(productAttributes.get(i).getValues().get(0));
            }

            // Initialize the ProductAttributeValuesAdapter for RecyclerView
            attributesAdapter = new ProductAttributeValuesAdapter(context,
        selectedAttributeValues);

            holder.attributes_recycler.setAdapter(attributesAdapter);
            holder.attributes_recycler.setLayoutManager(new
        LinearLayoutManager(context));

            attributesAdapter.notifyDataSetChanged();



            // Holds Product Quantity
            final int[] number = {1};
            number[0] =
        cartProduct.getCustomersBasketProduct().getCustomersBasketQuantity();


            // Decrease Product Quantity
            holder.cart_item_quantity_minusBtn.setOnClickListener(new
        View.OnClickListener() {
                @Override
                public void onClick(View view) {
                    // Check if the Quantity is greater than the minimum Quantity
                    if (number[0] > 1)
                    {
                        // Decrease Quantity by 1
                        number[0] = number[0] - 1;
                        holder.cart_item_quantity.setText(""+ number[0]);

                        // Calculate Product Price with selected Quantity
                        double price =
        Double.parseDouble(cartProduct.getCustomersBasketProduct().getProductsFin
        alPrice()) * number[0];

                        // Set Final Price and Quantity
                        cartProduct.getCustomersBasketProduct().setTotalPrice(""+
        price);

        cartProduct.getCustomersBasketProduct().setCustomersBasketQuantity(number
        [0]);
```

```
                // Update CartItem in Local Database using static method
of My_Cart
                My_Cart.UpdateCartItem
                        (
                                cartProduct
                        );

                // Calculate Cart's Total Price Again
                setCartTotal();

                notifyItemChanged(holder.getAdapterPosition());
            }
        }
    });


    // Increase Product Quantity
    holder.cart_item_quantity_plusBtn.setOnClickListener(new
View.OnClickListener() {
        @Override
        public void onClick(View view) {
            // Check if the Quantity is less than the maximum or stock
Quantity
            if (number[0] <
cartProduct.getCustomersBasketProduct().getProductsQuantity()) {
                // Increase Quantity by 1
                number[0] = number[0] + 1;
                holder.cart_item_quantity.setText(""+ number[0]);

                // Calculate Product Price with selected Quantity
                double price =
Double.parseDouble(cartProduct.getCustomersBasketProduct().getProductsFin
alPrice()) * number[0];

                // Set Final Price and Quantity
                cartProduct.getCustomersBasketProduct().setTotalPrice(""+
price);

cartProduct.getCustomersBasketProduct().setCustomersBasketQuantity(number
[0]);


                // Update CartItem in Local Database using static method
of My_Cart
                My_Cart.UpdateCartItem
                        (
                                cartProduct
                        );

                // Calculate Cart's Total Price Again
                setCartTotal();

                notifyItemChanged(holder.getAdapterPosition());
            }
        }
    });
```

```
            // View Product Details
            holder.cart_item_viewBtn.setOnClickListener(new View.OnClickListener()
        {
                @Override
                public void onClick(View v) {
                    // Get Product Info
                    Bundle itemInfo = new Bundle();
                    itemInfo.putInt("itemID",
        cartProduct.getCustomersBasketProduct().getProductsId());

                    // Navigate to Product_Description of selected Product
                    Fragment fragment = new Product_Description();
                    fragment.setArguments(itemInfo);
                    FragmentManager fragmentManager = ((MainActivity)
        context).getSupportFragmentManager();
                    fragmentManager.beginTransaction()
                            .add(R.id.main_fragment, fragment)

        .setTransition(FragmentTransaction.TRANSIT_FRAGMENT_FADE)
                            .addToBackStack(null)
                            .commit();


                    // Add the Product to User's Recently Viewed Products
                    if
(!recents_db.getUserRecents().contains(cartProduct.getCustomersBasketProduct().get
ProductsId())) {

recents_db.insertRecentItem(cartProduct.getCustomersBasketProduct().getProductsId(
));
                    }
                }
            });



            // Delete relevant Product from Cart
            holder.cart_item_removeBtn.setOnClickListener(new
View.OnClickListener() {
                @Override
                public void onClick(View view) {

                    holder.cart_item_removeBtn.setClickable(false);

                    // Delete CartItem from Local Database using static method of
My_Cart
                    My_Cart.DeleteCartItem
                            (
                                    cartProduct.getCustomersBasketId()
                            );


                    // Calculate Cart's Total Price Again
                    setCartTotal();

                    // Remove CartItem from Cart List
                    cartItemsList.remove(holder.getAdapterPosition());
                    // Notify that item at position has been removed
                    notifyItemRemoved(holder.getAdapterPosition());

                    // Update Cart View from Local Database using static method of
My_Cart
                    cartFragment.updateCartView(getItemCount());

                }
            });

        }


    }



    //********** Returns the total number of items in the data set *********//

    @Override
    public int getItemCount() {
        return (cartItemsList.size() == 0) ? cartItemsList.size() :
cartItemsList.size() + 1;
    }


    //********** Return the view type of the item at position for the purposes of
view recycling *********//

    @Override
    public int getItemViewType(int position) {
        return (position == cartItemsList.size()) ? R.layout.layout_button :
R.layout.layout_card_cart_items;
    }



    //*********** Calculate and Set the Cart's Total Price ********//

    public void setCartTotal() {

        double finalPrice = 0;
        List<CartProduct> cartItemsList = user_cart_db.getCartItems();

        for (int i=0;  i<cartItemsList.size();  i++) {
            // Update the Cart's Total Price
            finalPrice +=
Double.parseDouble(cartItemsList.get(i).getCustomersBasketProduct().getTotalPrice(
));
        }

        cartFragment.cart_total_price.setText(context.getString(R.string.total) +"
: "+ ConstantValues.CURRENCY_SYMBOL + new
DecimalFormat("#0.00").format(finalPrice));
        cartFragment.cart_item_total_price.setText(ConstantValues.CURRENCY_SYMBOL
+ new DecimalFormat("#0.00").format(finalPrice));
```

```
cartFragment.cart_item_discount_price.setText(ConstantValues.CURRENCY_SYMBOL + new
DecimalFormat("#0.00").format(0.0));

cartFragment.cart_item_subtotal_price.setText(ConstantValues.CURRENCY_SYMBOL + new
DecimalFormat("#0.00").format(finalPrice));
    }



    /********** Custom ViewHolder provides a direct reference to each of the Views
within a Data_Item *********/

    public class MyViewHolder extends RecyclerView.ViewHolder {

        private Button custom_button;
        private Button cart_item_viewBtn;
        private ImageView cart_item_cover;
        private RecyclerView attributes_recycler;
        private ImageButton cart_item_quantity_minusBtn,
cart_item_quantity_plusBtn;
        private TextView cart_item_removeBtn;
        private TextView cart_item_title, cart_item_category,
cart_item_base_price, cart_item_sub_price, cart_item_quantity;


        public MyViewHolder(final View itemView) {
            super(itemView);

            custom_button = (Button) itemView.findViewById(R.id.custom_button);

            cart_item_title = (TextView)
itemView.findViewById(R.id.cart_item_title);
            cart_item_base_price = (TextView)
itemView.findViewById(R.id.cart_item_base_price);
            cart_item_sub_price = (TextView)
itemView.findViewById(R.id.cart_item_sub_price);
            cart_item_quantity = (TextView)
itemView.findViewById(R.id.cart_item_quantity);
            cart_item_category = (TextView)
itemView.findViewById(R.id.cart_item_category);
            cart_item_cover = (ImageView)
itemView.findViewById(R.id.cart_item_cover);
            cart_item_viewBtn = (Button)
itemView.findViewById(R.id.cart_item_viewBtn);
            cart_item_removeBtn = (TextView)
itemView.findViewById(R.id.cart_item_removeBtn);
            cart_item_quantity_plusBtn = (ImageButton)
itemView.findViewById(R.id.cart_item_quantity_plusBtn);
            cart_item_quantity_minusBtn = (ImageButton)
itemView.findViewById(R.id.cart_item_quantity_minusBtn);

            attributes_recycler = (RecyclerView)
itemView.findViewById(R.id.cart_item_attributes_recycler);
        }

    }

}
```

Coupons Adapter.java [class file]

```
public class CouponsAdapter extends
RecyclerView.Adapter<CouponsAdapter.MyViewHolder> {

    Context context;
    Boolean isRemovable;

    CheckoutFinal checkoutFinal;
    List<CouponsInfo> couponsList;

    private String currency = "";

    public CouponsAdapter(Context context, List<CouponsInfo> couponsList, Boolean
isRemovable, CheckoutFinal checkoutFinal) {
        this.context = context;
        this.checkoutFinal = checkoutFinal;
        this.isRemovable = isRemovable;
        this.couponsList = couponsList;
    }

    public CouponsAdapter(Context context, List<CouponsInfo> couponsList, Boolean
isRemovable, CheckoutFinal checkoutFinal, String currency) {
        this.context = context;
        this.checkoutFinal = checkoutFinal;
        this.isRemovable = isRemovable;
        this.couponsList = couponsList;
        this.currency = currency;
    }


    //********** Called to Inflate a Layout from XML and then return the Holder
*********//

    @Override
    public MyViewHolder onCreateViewHolder(final ViewGroup parent, int viewType) {
        // Inflate the custom layout
        View itemView =
LayoutInflater.from(parent.getContext()).inflate(R.layout.layout_card_coupons,
parent, false);

        return new MyViewHolder(itemView);
    }


    //********** Called by RecyclerView to display the Data at the specified
Position *********//

    @Override
    public void onBindViewHolder(final MyViewHolder holder, final int position) {
        // Get the data model based on Position
        final CouponsInfo coupon = couponsList.get(position);

        holder.coupon_code.setText(coupon.getCode());
        if (currency.isEmpty())
            holder.coupon_discount.setText(ConstantValues.CURRENCY_SYMBOL + new
DecimalFormat("#0.00").format(Double.parseDouble(coupon.getAmount())));
        else
            holder.coupon_discount.setText(new
DecimalFormat("#0.00").format(Double.parseDouble(coupon.getAmount())) + " " +
currency);

        if (coupon.getDiscountType().equalsIgnoreCase("fixed_cart") ||
coupon.getDiscountType().equalsIgnoreCase("percent")) {
            holder.coupon_type.setText(context.getString(R.string.cart));
        } else if (coupon.getDiscountType().equalsIgnoreCase("fixed_product") ||
coupon.getDiscountType().equalsIgnoreCase("percent_product")) {
            holder.coupon_type.setText(context.getString(R.string.product));
        }


        if (coupon.getDiscountType().equalsIgnoreCase("fixed_cart") ||
coupon.getDiscountType().equalsIgnoreCase("fixed_product")) {
            if (currency.isEmpty())
                holder.coupon_amount.setText(ConstantValues.CURRENCY_SYMBOL +
coupon.getAmount());
            else
                holder.coupon_amount.setText(coupon.getAmount() + " " + currency);
        } else if (coupon.getDiscountType().equalsIgnoreCase("percent") ||
coupon.getDiscountType().equalsIgnoreCase("percent_product")) {
            holder.coupon_amount.setText(coupon.getAmount() + "%");
        }


        if (isRemovable) {
            holder.coupon_delete.setVisibility(View.VISIBLE);

            holder.coupon_delete.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View v) {

                    checkoutFinal.removeCoupon(coupon);

                    notifyItemRemoved(holder.getAdapterPosition());
                    notifyItemRangeRemoved(holder.getAdapterPosition(),
getItemCount());
                    notifyDataSetChanged();
                }
            });

        } else {
            holder.coupon_delete.setVisibility(View.GONE);
        }

    }


    //********** Returns the total number of items in the data set *********//

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


    /********** Custom ViewHolder provides a direct reference to each of the Views
within a Data_Item *********/
```

```
    public static class MyViewHolder extends RecyclerView.ViewHolder {

        private ImageButton coupon_delete;
        private TextView coupon_code, coupon_type, coupon_amount, coupon_discount;


        public MyViewHolder(final View itemView) {
            super(itemView);

            coupon_code = (TextView) itemView.findViewById(R.id.coupon_code);
            coupon_type = (TextView) itemView.findViewById(R.id.coupon_type);
            coupon_amount = (TextView) itemView.findViewById(R.id.coupon_amount);
            coupon_discount = (TextView)
itemView.findViewById(R.id.coupon_discount);
            coupon_delete = (ImageButton)
itemView.findViewById(R.id.coupon_delete);

        }
    }
}
```

Filter product.java

```
public class FilterProductsAdapter extends
RecyclerView.Adapter<FilterProductsAdapter.MyViewHolder>{

    Context context;
    List<FilterDetails> filtersList;
    List<FiltersAttributes> filtersAttributesList;
    FilterProductAttributesAdapter filterProductAttributesAdapter;


    public FilterProductsAdapter(Context context, List<FilterDetails> filtersList,
List<FiltersAttributes> filtersAttributesList) {
        this.context = context;
        this.filtersList = filtersList;
        this.filtersAttributesList = filtersAttributesList;
    }



    //********** Called to Inflate a Layout from XML and then return the Holder
*********//

    @Override
    public MyViewHolder onCreateViewHolder(final ViewGroup parent, int viewType) {
        // Inflate the custom layout
        View itemView =
LayoutInflater.from(parent.getContext()).inflate(R.layout.layout_filters, parent,
false);

        return new MyViewHolder(itemView);
    }



    //********** Called by RecyclerView to display the Data at the specified
Position *********//
```

```
    @Override
    public void onBindViewHolder(final MyViewHolder holder, final int position) {
        // Get the data model based on Position
        FilterDetails filterDetails = filtersList.get(position);

        FilterOption filterOption = filterDetails.getOption();
        List<FilterValue> filterValues = filterDetails.getValues();

        holder.filter_name.setText(filterOption.getName());


        // Adapter for Attribute Values
        filterProductAttributesAdapter = new
FilterProductAttributesAdapter(context, filterValues, filterOption.getName(),
filtersAttributesList);
        holder.filter_attributes_recycler.setLayoutManager(new
LinearLayoutManager(context));

        // Set adapter to AttributeValues RecyclerView

holder.filter_attributes_recycler.setAdapter(filterProductAttributesAdapter);
    }



    //********** Returns the total number of items in the data set *********//

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



    /********** Custom ViewHolder provides a direct reference to each of the Views
within a Data_Item *********/

    public static class MyViewHolder extends RecyclerView.ViewHolder {

        public TextView filter_name;
        public RecyclerView filter_attributes_recycler;


        public MyViewHolder(final View itemView) {
            super(itemView);
            filter_name = (TextView) itemView.findViewById(R.id.filter_name);
            filter_attributes_recycler = (RecyclerView)
itemView.findViewById(R.id.filter_attributes_recycler);
        }
    }
}
```

product list.java

```
package com.amanakash.ecommerce.adapters;

import android.content.Context;
import android.content.Intent;
import android.graphics.Paint;
import android.graphics.drawable.Drawable;
import android.os.Build;
import android.os.Bundle;
import android.os.CountDownTimer;

import com.bumptech.glide.Priority;
import com.bumptech.glide.load.DataSource;
import com.bumptech.glide.load.engine.DiskCacheStrategy;
import com.bumptech.glide.load.engine.GlideException;
import com.bumptech.glide.request.RequestListener;
import com.bumptech.glide.request.RequestOptions;
import com.bumptech.glide.request.target.Target;
import com.facebook.shimmer.ShimmerFrameLayout;
import com.google.android.material.snackbar.Snackbar;

import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;
import androidx.fragment.app.FragmentManager;
import androidx.fragment.app.FragmentTransaction;
import androidx.core.content.ContextCompat;
import androidx.recyclerview.widget.RecyclerView;

import android.text.TextUtils;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.ImageButton;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.TextView;
import android.widget.ToggleButton;

import com.bumptech.glide.Glide;


import com.amanakash.ecommerce.activities.MainActivity;

import com.amanakash.ecommerce.app.App;
import com.amanakash.ecommerce.models.product_model.Option;
import com.amanakash.ecommerce.models.product_model.Value;
import com.amanakash.ecommerce.utils.Utilities;
import com.amanakash.ecommerce.activities.Login;
import com.amanakash.ecommerce.databases.User_Recents_DB;
import com.amanakash.ecommerce.fragment.My_Cart;
import com.amanakash.ecommerce.fragment.Product_Description;
import com.amanakashcorporatio.ecomstore.R;

import java.text.DecimalFormat;
import java.text.NumberFormat;
import java.util.ArrayList;
import java.util.List;
import java.util.Locale;

import com.amanakash.ecommerce.constant.ConstantValues;
import com.amanakash.ecommerce.models.cart_model.CartProduct;
import com.amanakash.ecommerce.models.cart_model.CartProductAttributes;
import com.amanakash.ecommerce.models.product_model.ProductDetails;

/**
 * ProductAdapter is the adapter class of RecyclerView holding List of Products in
All_Products and other Product relevant Classes
 **/

public class ProductAdapter extends
RecyclerView.Adapter<ProductAdapter.MyViewHolder> {

    private Context context;
    private String customerID;
    private Boolean isGridView;
    private Boolean isHorizontal;
    private Boolean isFlash;

    private User_Recents_DB recents_db;
    private List<ProductDetails> productList;

    int defaultSmLayoutId;
    int defaultLgLayoutId;


    public ProductAdapter(Context context, List<ProductDetails> productList,
Boolean isHorizontal, Boolean isFlash) {
        this.context = context;
        this.productList = productList;
        this.isHorizontal = isHorizontal;
        this.isFlash = isFlash;
        recents_db = new User_Recents_DB();
        customerID = this.context.getSharedPreferences("UserInfo",
Context.MODE_PRIVATE).getString("userID", "");
        setupDefaultLayoutId();
    }


    public ProductAdapter(Context context, List<ProductDetails> productList,
Boolean isHorizontal) {
        this.context = context;
        this.productList = productList;
        this.isHorizontal = isHorizontal;
        recents_db = new User_Recents_DB();
        customerID = this.context.getSharedPreferences("UserInfo",
Context.MODE_PRIVATE).getString("userID", "");
        setupDefaultLayoutId();
    }


    //********** Called to Inflate a Layout from XML and then return the Holder
*********//
```

```
    @Override
    public MyViewHolder onCreateViewHolder(final ViewGroup parent, int viewType) {
        View itemView = null;

        // Check which Layout will be Inflated
        if (isHorizontal) {
            itemView =
LayoutInflater.from(parent.getContext()).inflate(defaultSmLayoutId, parent,
false);
        } else {
            itemView = LayoutInflater.from(parent.getContext())
                    .inflate(isGridView ? defaultLgLayoutId :
R.layout.layout_product_0_list_lg, parent, false);
        }

        // Return a new holder instance
        return new MyViewHolder(itemView);
    }


    //********** Called by RecyclerView to display the Data at the specified
Position *********//

    @Override
    public void onBindViewHolder(final MyViewHolder holder, int position) {

        if (position != productList.size()) {

            // Get the data model based on Position
            final ProductDetails product = productList.get(position);

            // Check if the Product is already in the Cart
            if (My_Cart.checkCartHasProduct(product.getProductsId())) {
                holder.product_checked.setVisibility(View.VISIBLE);
            } else {
                holder.product_checked.setVisibility(View.GONE);
            }

            RequestOptions options = new RequestOptions()
                    .centerInside()
                    .placeholder(R.drawable.placeholder)
                    .error(R.drawable.placeholder)
                    .diskCacheStrategy(DiskCacheStrategy.ALL)
                    .priority(Priority.HIGH);
            // Set Product Image on ImageView with Glide Library
            Glide.with(context)
                    .setDefaultRequestOptions(options)
                    .load(ConstantValues.ECOMMERCE_URL +
product.getProductsImage())
                    .addListener(new RequestListener<Drawable>() {
                        @Override
                        public boolean onLoadFailed(@Nullable GlideException e,
Object model, Target<Drawable> target, boolean isFirstResource) {
                            holder.shimmerProgress.setVisibility(View.GONE);
                            holder.shimmerProgress.stopShimmer();
                            return false;
                        }
                        @Override
                        public boolean onResourceReady(Drawable resource, Object
model, Target<Drawable> target, DataSource dataSource, boolean isFirstResource) {
                            holder.shimmerProgress.setVisibility(View.GONE);
                            holder.shimmerProgress.stopShimmer();
                            return false;
                        }
                    })
                    .into(holder.product_thumbnail);

            if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.LOLLIPOP) {
                holder.product_thumbnail.setClipToOutline(true);
                holder.product_tag_new.setClipToOutline(true);
            }

            holder.product_title.setText(product.getProductsName());


holder.product_price_old.setPaintFlags(holder.product_price_old.getPaintFlags() |
Paint.STRIKE_THRU_TEXT_FLAG);

            String[] categoryIDs = new String[product.getCategories().size()];
            String[] categoryNames = new String[product.getCategories().size()];
            if (product.getCategories().size() > 0) {

                for (int i = 0; i < product.getCategories().size(); i++) {
                    categoryIDs[i] =
String.valueOf(product.getCategories().get(i).getCategoriesId());
                    categoryNames[i] =
product.getCategories().get(i).getCategoriesName();
                }
                product.setCategoryIDs(TextUtils.join(",", categoryIDs));
                product.setCategoryNames(TextUtils.join(",", categoryNames));
            } else {
                product.setCategoryIDs("");
                product.setCategoryNames("");
            }

            holder.product_category.setText(product.getCategoryNames());

            // Calculate the Discount on Product with static method of Helper
class
            final String discount =
Utilities.checkDiscount(product.getProductsPrice(), product.getDiscountPrice());

            if (discount != null) {
                // Set Product's Price
                holder.product_price_old.setVisibility(View.VISIBLE);
                holder.product_price_old.setText(ConstantValues.CURRENCY_SYMBOL +
"" + new
DecimalFormat("#0.00").format(Double.valueOf(product.getProductsPrice())));
                holder.product_price_new.setText(ConstantValues.CURRENCY_SYMBOL +
"" + new
DecimalFormat("#0.00").format(Double.valueOf(product.getDiscountPrice())));

                holder.product_tag_new.setVisibility(View.GONE);

                // Set Discount Tag and its Text
                holder.layoutSale.setVisibility(View.VISIBLE);
                holder.product_tag_discount_text.setText(discount);

            } else {

                // Check if the Product is Newly Added with the help of static
method of Helper class
                if (Utilities.checkNewProduct(product.getProductsDateAdded())) {
                    // Set New Tag and its Text
                    holder.product_tag_new.setVisibility(View.VISIBLE);
                } else {
                    holder.product_tag_new.setVisibility(View.GONE);
                }

                // Hide Discount Text and Set Product's Price
                holder.layoutSale.setVisibility(View.GONE);
                holder.product_price_old.setVisibility(View.GONE);
/*
                NumberFormat nf =
NumberFormat.getInstance(LocaleHelper.getSystemLocale(
context.getResources().getConfiguration())); //or "nb","No" - for Norway
                String thisprice = nf.format(Double.parseDouble());
*/

                holder.product_price_new.setText(ConstantValues.CURRENCY_SYMBOL +
"" + new
DecimalFormat("#0.00").format(Double.parseDouble(product.getProductsPrice())));
            }


            holder.product_like_btn.setOnCheckedChangeListener(null);

            // Check if Product is Liked
            if (product.getIsLiked().equalsIgnoreCase("1")) {
                holder.product_like_btn.setChecked(true);
            } else {
                holder.product_like_btn.setChecked(false);
            }


            // Handle the Click event of product_like_btn ToggleButton
            holder.product_like_btn.setOnClickListener(new View.OnClickListener()
{
                @Override
                public void onClick(View view) {

                    // Check if the User is Authenticated
                    if (ConstantValues.IS_USER_LOGGED_IN) {


                        if (holder.product_like_btn.isChecked()) {
                            product.setIsLiked("1");
                            holder.product_like_btn.setChecked(true);

                            // Like the Product for the User with the static
method of Product_Description

Product_Description.LikeProduct(product.getProductsId(), customerID, context,
view);
                        } else {
                            product.setIsLiked("0");
                            holder.product_like_btn.setChecked(false);

                            // Unlike the Product for the User with the static
method of Product_Description

Product_Description.UnlikeProduct(product.getProductsId(), customerID, context,
view);
                        }

                    } else {
                        // Keep the Like Button Unchecked
                        holder.product_like_btn.setChecked(false);

                        // Navigate to Login Activity
                        Intent i = new Intent(context, Login.class);
                        context.startActivity(i);
                        ((MainActivity) context).finish();
                        ((MainActivity)
context).overridePendingTransition(R.anim.enter_from_left, R.anim.exit_out_left);
                    }
                }
            });


            // Handle the Click event of product_thumbnail ImageView
            holder.product_thumbnail.setOnClickListener(new View.OnClickListener()
{
                @Override
                public void onClick(View view) {

                    // Get Product Info
                    Bundle itemInfo = new Bundle();
                    itemInfo.putParcelable("productDetails", product);

                    // Save the AddressDetails
                    ((App)
context.getApplicationContext()).setProductDetails(product);
                    // Navigate to Product_Description of selected Product
                    Fragment fragment = new
Product_Description(holder.product_checked);
                    fragment.setArguments(itemInfo);

//MainActivity.actionBarDrawerToggle.setDrawerIndicatorEnabled(false);
                    FragmentManager fragmentManager = ((MainActivity)
context).getSupportFragmentManager();
                    fragmentManager.beginTransaction()
                            .hide(((MainActivity) context).currentFragment)
                            .add(R.id.main_fragment, fragment)

.setTransition(FragmentTransaction.TRANSIT_FRAGMENT_FADE)
                            .addToBackStack(null).commit();


                    // Add the Product to User's Recently Viewed Products
                    if
```

```
(!recents_db.getUserRecents().contains(product.getProductsId())) {
                        recents_db.insertRecentItem(product.getProductsId());
                    }
                }
            });


            // Handle the Click event of product_checked ImageView
            holder.product_checked.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View view) {

                    // Get Product Info
                    Bundle itemInfo = new Bundle();
                    itemInfo.putParcelable("productDetails", product);

                    // Navigate to Product_Description of selected Product
                    Fragment fragment = new
Product_Description(holder.product_checked);
                    fragment.setArguments(itemInfo);

//MainActivity.actionBarDrawerToggle.setDrawerIndicatorEnabled(false);
                    FragmentManager fragmentManager = ((MainActivity)
context).getSupportFragmentManager();
                    fragmentManager.beginTransaction()
                            .hide(((MainActivity) context).currentFragment)
                            .add(R.id.main_fragment, fragment)

.setTransition(FragmentTransaction.TRANSIT_FRAGMENT_FADE)
                            .addToBackStack(null).commit();


                    // Add the Product to User's Recently Viewed Products
                    if
(!recents_db.getUserRecents().contains(product.getProductsId())) {
                        recents_db.insertRecentItem(product.getProductsId());
                    }
                }
            });


            // Check the Button's Visibility
            if (ConstantValues.IS_ADD_TO_CART_BUTTON_ENABLED) {

                holder.product_add_cart_btn.setVisibility(View.VISIBLE);
                holder.product_add_cart_btn.setOnClickListener(null);

                if (product.getProductsType() != 0) {

holder.product_add_cart_btn.setText(context.getString(R.string.view_product));

holder.product_add_cart_btn.setBackground(ContextCompat.getDrawable(context,
R.drawable.rounded_corners_button_green));
                } else {
                    if (product.getProductsDefaultStock() < 1) {

holder.product_add_cart_btn.setText(context.getString(R.string.outOfStock));
holder.product_add_cart_btn.setBackground(ContextCompat.getDrawable(context,
R.drawable.rounded_corners_button_red));
                    } else {

holder.product_add_cart_btn.setText(context.getString(R.string.addToCart));

holder.product_add_cart_btn.setBackground(ContextCompat.getDrawable(context,
R.drawable.rounded_corners_button_green));


                    }
                }


                if (isFlash) {
                    holder.product_like_btn.setVisibility(View.GONE);

                    holder.start = Long.parseLong(product.getFlashStartDate()) *
1000L;
                    holder.end = Long.parseLong(product.getFlashExpireDate()) *
1000L;

                    holder.server = Long.parseLong(product.getServerTime()) *
1000L;

                    //  remainingTime =  end-server ;

                    if (holder.server > holder.start) {

holder.product_price_new.setText(ConstantValues.CURRENCY_SYMBOL + "" + new
DecimalFormat("#0.00").format(Double.valueOf(product.getFlashPrice())));

holder.product_price_old.setText(ConstantValues.CURRENCY_SYMBOL + "" + new
DecimalFormat("#0.00").format(Double.valueOf(product.getProductsPrice())));

                        if (holder.mCountDownTimer != null){
                            holder.mCountDownTimer.cancel();
                        }

                        holder.mCountDownTimer = new CountDownTimer(holder.end,
1000) {
                            @Override
                            public void onTick(long millisUntilFinished) {

                                holder.server = holder.server - 1;
                                Long serverUptimeSeconds =
                                        (millisUntilFinished - holder.server) /
1000;

                                String daysLeft = String.format("%d",
serverUptimeSeconds / 86400);
                                String hoursLeft = String.format("%d",
(serverUptimeSeconds % 86400) / 3600);
                                String minutesLeft = String.format("%d",
((serverUptimeSeconds % 86400) % 3600) / 60);
                                String secondsLeft = String.format("%d",
((serverUptimeSeconds % 86400) % 3600) % 60);
                                holder.product_add_cart_btn.setText(daysLeft +
"D:" + hoursLeft + "H:" + minutesLeft + "M:" + secondsLeft + "S");

                            }

                            @Override
                            public void onFinish() {

holder.product_add_cart_btn.setText(context.getResources().getString(R.string.upco
ming));

holder.product_add_cart_btn.setBackgroundResource(R.drawable.rounded_corners_butto
n_red);
                            }
                        }.start();
                    } else {

holder.product_price_new.setText(ConstantValues.CURRENCY_SYMBOL + "" + new
DecimalFormat("#0.00").format(Double.valueOf(product.getFlashPrice())));

holder.product_price_old.setText(ConstantValues.CURRENCY_SYMBOL + "" + new
DecimalFormat("#0.00").format(Double.valueOf(product.getProductsPrice())));

holder.product_add_cart_btn.setText(context.getResources().getString(R.string.upco
ming));

holder.product_add_cart_btn.setBackgroundResource(R.drawable.rounded_corners_butto
n_red);
                    }

                    holder.product_price_old.setVisibility(View.VISIBLE);
                }

                holder.product_add_cart_btn.setOnClickListener(new
View.OnClickListener() {
                    @Override
                    public void onClick(View view) {


                        if (product.getProductsType() != 0) {

                            // Get Product Info
                            Bundle itemInfo = new Bundle();
                            itemInfo.putParcelable("productDetails", product);

                            // Navigate to Product_Description of selected Product
                            Fragment fragment = new
Product_Description(holder.product_checked);
                            fragment.setArguments(itemInfo);

//MainActivity.actionBarDrawerToggle.setDrawerIndicatorEnabled(false);
                            FragmentManager fragmentManager = ((MainActivity)
context).getSupportFragmentManager();
                            fragmentManager.beginTransaction()
                                    .hide(((MainActivity)
context).currentFragment)
                                    .add(R.id.main_fragment, fragment)

.setTransition(FragmentTransaction.TRANSIT_FRAGMENT_FADE)
```

```
                                    .addToBackStack(null).commit();

                            // Add the Product to User's Recently Viewed Products
                            if
(!recents_db.getUserRecents().contains(product.getProductsId())) {

recents_db.insertRecentItem(product.getProductsId());
                            }
                        } else {

                            if (isFlash) {
                                if (holder.start > holder.server) {
                                    Snackbar.make(view,
context.getString(R.string.cannot_add_upcoming), Snackbar.LENGTH_SHORT).show();
                                } else {
                                    Utilities.animateCartMenuIcon(context,
(MainActivity) context);
                                    // Add Product to User's Cart
                                    addProductToCart(product);


holder.product_checked.setVisibility(View.VISIBLE);

                                    Snackbar.make(view,
context.getString(R.string.item_added_to_cart), Snackbar.LENGTH_SHORT).show();

                                }
                            } else {

                                if (product.getProductsDefaultStock() < 1) {

                                    Snackbar.make(view,
context.getString(R.string.outOfStock), Snackbar.LENGTH_SHORT).show();
                                } else {
                                    Utilities.animateCartMenuIcon(context,
(MainActivity) context);
                                    // Add Product to User's Cart
                                    addProductToCart(product);


holder.product_checked.setVisibility(View.VISIBLE);

                                    Snackbar.make(view,
context.getString(R.string.item_added_to_cart), Snackbar.LENGTH_SHORT).show();
                                }

                            }
                        }
                    }
                });
                holder.product_card_img_btn.setOnClickListener(new
View.OnClickListener() {
                    @Override
                    public void onClick(View v) {
                        if (product.getProductsType() != 0) {

                            // Get Product Info
                            Bundle itemInfo = new Bundle();
                            itemInfo.putParcelable("productDetails", product);

                            // Navigate to Product_Description of selected Product
                            Fragment fragment = new
Product_Description(holder.product_checked);
                            fragment.setArguments(itemInfo);

//MainActivity.actionBarDrawerToggle.setDrawerIndicatorEnabled(false);
                            FragmentManager fragmentManager = ((MainActivity)
context).getSupportFragmentManager();
                            fragmentManager.beginTransaction()
                                    .hide(((MainActivity)
context).currentFragment)
                                    .add(R.id.main_fragment, fragment)

.setTransition(FragmentTransaction.TRANSIT_FRAGMENT_FADE)
                                    .addToBackStack(null).commit();

                            // Add the Product to User's Recently Viewed Products
                            if
(!recents_db.getUserRecents().contains(product.getProductsId())) {

recents_db.insertRecentItem(product.getProductsId());
                            }
                        } else {

                            if (isFlash) {
                                if (holder.start > holder.server) {
                                    Snackbar.make(v,
context.getString(R.string.cannot_add_upcoming), Snackbar.LENGTH_SHORT).show();
                                } else {
                                    Utilities.animateCartMenuIcon(context,
(MainActivity) context);
                                    // Add Product to User's Cart
                                    addProductToCart(product);


holder.product_checked.setVisibility(View.VISIBLE);

                                    Snackbar.make(v,
context.getString(R.string.item_added_to_cart), Snackbar.LENGTH_SHORT).show();

                                }
                            } else {

                                if (product.getProductsDefaultStock() < 1) {

                                    Snackbar.make(v,
context.getString(R.string.outOfStock), Snackbar.LENGTH_SHORT).show();
                                } else {
                                    Utilities.animateCartMenuIcon(context,
(MainActivity) context);
                                    // Add Product to User's Cart
                                    addProductToCart(product);


holder.product_checked.setVisibility(View.VISIBLE);
```

```
                                    Snackbar.make(v,
context.getString(R.string.item_added_to_cart), Snackbar.LENGTH_SHORT).show();
                                }

                            }
                        }
                    }
                });

            } else {
                // Make the Button Invisible
                holder.product_add_cart_btn.setVisibility(View.GONE);
            }

        }

    }


    //********** Returns the total number of items in the data set *********//

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


    //********** Toggles the RecyclerView LayoutManager *********//

    public void toggleLayout(Boolean isGridView) {
        this.isGridView = isGridView;
    }


    /********** Custom ViewHolder provides a direct reference to each of the Views
within a Data_Item *********/

    public static class MyViewHolder extends RecyclerView.ViewHolder {

        long start;
        long end;
        long server;
        CountDownTimer mCountDownTimer;


        ImageView product_checked;
        Button product_add_cart_btn;
        ImageButton product_card_img_btn;
        ToggleButton product_like_btn;
        ImageView product_thumbnail, product_tag_new;
        TextView product_title, product_category, product_price_old,
product_price_new, product_tag_discount_text;
        LinearLayout layoutSale;
        ShimmerFrameLayout shimmerProgress;

        public MyViewHolder(final View itemView) {
            super(itemView);
            product_checked = (ImageView)
itemView.findViewById(R.id.product_checked);

            product_add_cart_btn = (Button)
itemView.findViewById(R.id.product_card_Btn);
            product_card_img_btn =
itemView.findViewById(R.id.product_card_img_btn);
            product_like_btn = (ToggleButton)
itemView.findViewById(R.id.product_like_btn);
            product_title = (TextView) itemView.findViewById(R.id.product_title);
            product_category = (TextView)
itemView.findViewById(R.id.product_category);
            product_price_old = (TextView)
itemView.findViewById(R.id.product_price_old);
            product_price_new = (TextView)
itemView.findViewById(R.id.product_price_new);
            product_thumbnail = (ImageView)
itemView.findViewById(R.id.product_cover);
            product_tag_new = (ImageView)
itemView.findViewById(R.id.product_item_tag_new);
            product_tag_discount_text = (TextView)
itemView.findViewById(R.id.productItemTagOff);
            layoutSale = itemView.findViewById(R.id.saleLayout);
            shimmerProgress = itemView.findViewById(R.id.shimmerFrame);
        }

    }

    //********** Adds the Product to User's Cart *********//

    private void addProductToCart(ProductDetails product) {

        CartProduct cartProduct = new CartProduct();

        double productBasePrice, productFinalPrice = 0.0, attributesPrice = 0;
        List<CartProductAttributes> selectedAttributesList = new ArrayList<>();


        // Check Discount on Product with the help of static method of Helper
class
        final String discount =
Utilities.checkDiscount(product.getProductsPrice(), product.getDiscountPrice());

        // Get Product's Price based on Discount
        if (discount != null) {
            product.setIsSaleProduct("1");
            productBasePrice = Double.parseDouble(product.getDiscountPrice());
        } else {
            product.setIsSaleProduct("0");
            productBasePrice = Double.parseDouble(product.getProductsPrice());
        }


        // Get Default Attributes from AttributesList
        for (int i = 0; i < product.getAttributes().size(); i++) {

            CartProductAttributes productAttribute = new CartProductAttributes();
            // Get Name and First Value of current Attribute
            Option option = product.getAttributes().get(i).getOption();
            Value value = product.getAttributes().get(i).getValues().get(0);


            // Add the Attribute's Value Price to the attributePrices
            String attrPrice = value.getPricePrefix() + value.getPrice();
            attributesPrice += Double.parseDouble(attrPrice);


            // Add Value to new List
            List<Value> valuesList = new ArrayList<>();
            valuesList.add(value);


            // Set the Name and Value of Attribute
            productAttribute.setOption(option);
            productAttribute.setValues(valuesList);


            // Add current Attribute to selectedAttributesList
            selectedAttributesList.add(i, productAttribute);
        }

        if (isFlash) {
            productFinalPrice = Double.parseDouble(product.getFlashPrice()) +
attributesPrice;
        } else {
            // Add Attributes Price to Product's Final Price
            productFinalPrice = productBasePrice + attributesPrice;
        }


        // Set Product's Price and Quantity
        product.setCustomersBasketQuantity(1);
        product.setProductsPrice(String.valueOf(productBasePrice));
        product.setAttributesPrice(String.valueOf(attributesPrice));
        product.setProductsFinalPrice(String.valueOf(productFinalPrice));

        int quantity = product.getProductsDefaultStock();
        product.setProductsQuantity(product.getProductsDefaultStock());

        // Set Product's OrderProductCategory Info
        String[] categoryIDs = new String[product.getCategories().size()];
        String[] categoryNames = new String[product.getCategories().size()];
        if (product.getCategories().size() > 0) {

            for (int i = 0; i < product.getCategories().size(); i++) {
                categoryIDs[i] =
String.valueOf(product.getCategories().get(i).getCategoriesId());
                categoryNames[i] =
product.getCategories().get(i).getCategoriesName();
            }

            product.setCategoryIDs(TextUtils.join(",", categoryIDs));
            product.setCategoryNames(TextUtils.join(",", categoryNames));
        } else {
            product.setCategoryIDs("");
            product.setCategoryNames("");
        }
        // product.setCategoryNames(product.getCategoryNames());

        product.setTotalPrice(String.valueOf(productFinalPrice));


        // Set Customer's Basket Product and selected Attributes Info
        cartProduct.setCustomersBasketProduct(product);
        cartProduct.setCustomersBasketProductAttributes(selectedAttributesList);


        // Add the Product to User's Cart with the help of static method of
My_Cart class
        My_Cart.AddCartItem
                (
                        cartProduct
                );


        // Recreate the OptionsMenu
        ((MainActivity) context).invalidateOptionsMenu();

    }

    private void setupDefaultLayoutId() {
        switch (ConstantValues.DEFAULT_PRODUCT_CARD_STYLE) {
            case 0:
                defaultSmLayoutId = R.layout.layout_product_0_grid_sm;
                defaultLgLayoutId = R.layout.layout_product_0_grid_lg;
                break;
            case 1:
                defaultSmLayoutId = R.layout.layout_product_1_grid_sm;
                defaultLgLayoutId = R.layout.layout_product_1_grid_lg;
                break;
            case 2:
                defaultSmLayoutId = R.layout.layout_product_2_grid_sm;
                defaultLgLayoutId = R.layout.layout_product_2_grid_lg;
                break;
            case 3:
                defaultSmLayoutId = R.layout.layout_product_3_grid_sm;
                defaultLgLayoutId = R.layout.layout_product_3_grid_lg;
                break;
            case 4:
                defaultSmLayoutId = R.layout.layout_product_4_grid_sm;
                defaultLgLayoutId = R.layout.layout_product_4_grid_lg;
                break;
            case 5:
                defaultSmLayoutId = R.layout.layout_product_5_grid_sm;
                defaultLgLayoutId = R.layout.layout_product_5_grid_lg;
                break;
            case 6:
                defaultSmLayoutId = R.layout.layout_product_6_grid_sm;
                defaultLgLayoutId = R.layout.layout_product_6_grid_lg;
                break;
            case 7:
                defaultSmLayoutId = R.layout.layout_product_0_grid_sm;
                defaultLgLayoutId = R.layout.layout_product_0_grid_lg;
                break;
            case 8:
                defaultSmLayoutId = R.layout.layout_product_8_grid_sm;
                defaultLgLayoutId = R.layout.layout_product_8_grid_lg;
                break;
            case 9:
                defaultSmLayoutId = R.layout.layout_product_9_grid_sm;
                defaultLgLayoutId = R.layout.layout_product_9_grid_lg;
                break;
            case 10:
                defaultSmLayoutId = R.layout.layout_product_10_grid_sm;
                defaultLgLayoutId = R.layout.layout_product_10_grid_lg;
                break;
            case 11:
                defaultSmLayoutId = R.layout.layout_product_11_grid_sm;
                defaultLgLayoutId = R.layout.layout_product_11_grid_lg;
                break;
            case 12:
                defaultSmLayoutId = R.layout.layout_product_12_grid_sm;
                defaultLgLayoutId = R.layout.layout_product_12_grid_lg;
                break;
            case 13:
                defaultSmLayoutId = R.layout.layout_product_13_grid_sm;
                defaultLgLayoutId = R.layout.layout_product_13_grid_lg;
                break;
            case 14:
                defaultSmLayoutId = R.layout.layout_product_14_grid_sm;
                defaultLgLayoutId = R.layout.layout_product_14_grid_lg;
                break;
            case 15:
                defaultSmLayoutId = R.layout.layout_product_15_grid_sm;
                defaultLgLayoutId = R.layout.layout_product_15_grid_lg;
                break;
            case 16:
                defaultSmLayoutId = R.layout.layout_product_16_grid_sm;
                defaultLgLayoutId = R.layout.layout_product_16_grid_lg;
                break;
            case 17:
                defaultSmLayoutId = R.layout.layout_product_17_grid_sm;
                defaultLgLayoutId = R.layout.layout_product_17_grid_lg;
                break;
            case 18:
                defaultSmLayoutId = R.layout.layout_product_18_grid_sm;
                defaultLgLayoutId = R.layout.layout_product_18_grid_lg;
                break;
            case 19:
                defaultSmLayoutId = R.layout.layout_product_19_grid_sm;
                defaultLgLayoutId = R.layout.layout_product_19_grid_lg;
                break;
            case 20:
                defaultSmLayoutId = R.layout.layout_product_20_grid_sm;
                defaultLgLayoutId = R.layout.layout_product_20_grid_lg;
                break;
            case 21:
                defaultSmLayoutId = R.layout.layout_product_21_grid_sm;
                defaultLgLayoutId = R.layout.layout_product_21_grid_lg;
                break;
            case 22:
```

```
                defaultSmLayoutId = R.layout.layout_product_22_grid_sm;
                defaultLgLayoutId = R.layout.layout_product_22_grid_lg;
                break;
        }
    }

    public String nFormate(double d) {
        NumberFormat nf = NumberFormat.getInstance(Locale.ENGLISH);
        nf.setMaximumFractionDigits(10);
        String st = nf.format(d);
        return st;
    }
}
```

Payment method.java

```
public class PaymentMethodAdapter extends BaseAdapter {

    private Context context;
    private List<PaymentMethodsInfo> paymentMethods;

    private LayoutInflater layoutInflater;


    public PaymentMethodAdapter(Context context, List<PaymentMethodsInfo>
paymentMethods) {
        this.context = context;
        this.paymentMethods = paymentMethods;

        layoutInflater = LayoutInflater.from(context);
    }


    //********** Returns the total number of items in the data set represented by
this Adapter *********//

    @Override
    public int getCount() {
        return paymentMethods.size();
    }


    //********** Returns the item associated with the specified position in the
data set *********//

    @Override
    public PaymentMethodsInfo getItem(int position) {
        return paymentMethods.get(position);
    }


    //********** Returns the item id associated with the specified position in the
list *********//

    @Override
    public long getItemId(int position) {
        return position;
    }
    //********** Returns a View that displays the data at the specified position
in the data set *********//

    @NonNull
    @Override
    public View getView(int position, View convertView, ViewGroup parent) {
        // Get the data model based on Position
        PaymentMethodsInfo paymentMethod = paymentMethods.get(position);


        final ViewHolder holder;

        // Check if an existing View is being Reused, otherwise Inflate the View
with custom Layout
        if (convertView == null) {
            convertView =
layoutInflater.inflate(R.layout.layout_attribute_values_dialog, parent, false);

            holder = new ViewHolder();

            holder.payment_method_name = (TextView)
convertView.findViewById(R.id.attribute_value_name);
            holder.payment_method_price = (TextView)
convertView.findViewById(R.id.attribute_value_price);
            holder.payment_method_prefix = (TextView)
convertView.findViewById(R.id.attribute_value_prefix);

            holder.payment_method_price.setVisibility(View.GONE);
            holder.payment_method_prefix.setVisibility(View.GONE);

            convertView.setTag(holder);
        }
        else {
            holder = (ViewHolder) convertView.getTag();
        }


        // Populate the data into the Template View
        holder.payment_method_name.setText(paymentMethod.getName());


        return convertView;
    }



    /********** Custom ViewHolder provides a direct reference to each of the Views
within a Data_Item *********/

    static class ViewHolder {
        private TextView payment_method_name, payment_method_prefix,
payment_method_price;

    }
```

```
}

search adapter.java [class file]
public class SearchResultsAdapter extends
RecyclerView.Adapter<SearchResultsAdapter.MyViewHolder> {

    Context context;
    List<SearchResults> searchResultsList;


    public SearchResultsAdapter(Context context, List<SearchResults>
searchResultsList) {
        this.context = context;
        this.searchResultsList = searchResultsList;
    }



    //********** Called to Inflate a Layout from XML and then return the Holder
*********//

    @Override
    public MyViewHolder onCreateViewHolder(final ViewGroup parent, int viewType) {
        // Inflate the custom layout
        View itemView =
LayoutInflater.from(parent.getContext()).inflate(R.layout.layout_search_results,
parent, false);

        return new MyViewHolder(itemView);
    }



    //********** Called by RecyclerView to display the Data at the specified
Position *********//

    @Override
    public void onBindViewHolder(final MyViewHolder holder, final int position) {
        // Get the data model based on Position
        final SearchResults searchResult = searchResultsList.get(position);

        // Get the Type of SearchItem (i.e Product, SubCategory)
        final String parent = searchResult.getParent();


        holder.search_result_title.setText(searchResult.getName());

        // Set Item Image on ImageView with Glide Library
        Glide
            .with(context)
            .load(ConstantValues.ECOMMERCE_URL+searchResult.getImage())
            .into(holder.search_result_cover);


        // Handle the SearchItem Click Event
        holder.itemView.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

                // Check the Type of SearchItem (i.e Product, Categories) to
navigate to relevant Fragment
                if (parent.equalsIgnoreCase("Categories")) {
                    // Get SubCategory Info
                    Bundle categoryInfo = new Bundle();
                    categoryInfo.putInt("CategoryID", searchResult.getId());
                    categoryInfo.putString("CategoryName",
searchResult.getName());

                    // Navigate to Products of selected SubCategory
                    Fragment fragment = new Products();
                    fragment.setArguments(categoryInfo);
                    FragmentManager fragmentManager = ((MainActivity)
context).getSupportFragmentManager();
                    fragmentManager.beginTransaction()
                            .add(R.id.main_fragment, fragment)

.setTransition(FragmentTransaction.TRANSIT_FRAGMENT_FADE)
                            .addToBackStack(null).commit();


                } else if (parent.equalsIgnoreCase("Products")) {
                    // Get Product Info
                    Bundle itemInfo = new Bundle();
                    itemInfo.putInt("itemID", searchResult.getId());

                    // Navigate to Product_Description of selected Product
                    Fragment fragment = new Product_Description();
                    fragment.setArguments(itemInfo);
                    FragmentManager fragmentManager = ((MainActivity)
context).getSupportFragmentManager();
                    fragmentManager.beginTransaction()
                            .add(R.id.main_fragment, fragment)

.setTransition(FragmentTransaction.TRANSIT_FRAGMENT_FADE)
                            .addToBackStack(null).commit();
                }
            }
        });

    }



    //********** Returns the total number of items in the data set *********//

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



    /********** Custom ViewHolder provides a direct reference to each of the Views
within a Data_Item *********/
```

```
    public static class MyViewHolder extends RecyclerView.ViewHolder {

        TextView search_result_title;
        ImageView search_result_cover;


        public MyViewHolder(final View itemView) {
            super(itemView);

            search_result_title = (TextView)
itemView.findViewById(R.id.search_result_title);
            search_result_cover = (ImageView)
itemView.findViewById(R.id.search_result_cover);
        }
    }
}
```

Application preference manager.java

```
public class MyAppPrefsManager {

    private SharedPreferences sharedPreferences;
    private SharedPreferences.Editor prefsEditor;

    private int PRIVATE_MODE = 0;
    private static final String PREF_NAME = "AndroidShopApp_Prefs";


    private static final String USER_LANGUAGE_ID  = "language_ID";
    private static final String USER_LANGUAGE_CODE  = "language_Code";
    private static final String CURRENCY_CODE = "currency_code";
    private static final String APPLICATION_VERSION = "application_version";
    private static final String IS_USER_LOGGED_IN = "isLogged_in";
    private static final String IS_FIRST_TIME_LAUNCH = "IsFirstTimeLaunch";
    private static final String IS_PUSH_NOTIFICATIONS_ENABLED =
"isPushNotificationsEnabled";
    private static final String IS_LOCAL_NOTIFICATIONS_ENABLED =
"isLocalNotificationsEnabled";

    private static final String LOCAL_NOTIFICATIONS_TITLE =
"localNotificationsTitle";
    private static final String LOCAL_NOTIFICATIONS_DURATION =
"localNotificationsDuration";
    private static final String LOCAL_NOTIFICATIONS_DESCRIPTION =
"localNotificationsDescription";

    private static final String Skip_For_Again = "skipMessage";

    public MyAppPrefsManager(Context context) {
        sharedPreferences = context.getSharedPreferences(PREF_NAME, PRIVATE_MODE);
        prefsEditor = sharedPreferences.edit();
    }

    public void setUserLanguageId(int langID) {
        prefsEditor.putInt(USER_LANGUAGE_ID, langID);
        prefsEditor.commit();
    }
    public Integer getUserLanguageId() {
        return sharedPreferences.getInt(USER_LANGUAGE_ID, 1);
    }

    public void setUserLanguageCode(String langCode) {
        prefsEditor.putString(USER_LANGUAGE_CODE, langCode);
        prefsEditor.commit();
    }

    public String getUserLanguageCode() {
        return sharedPreferences.getString(USER_LANGUAGE_CODE, "en");
    }

    public  void setCurrencyCode(String currencyCode){
        prefsEditor.putString(CURRENCY_CODE,currencyCode);
        prefsEditor.commit();
    }

    public String getCurrencyCode(){
        return sharedPreferences.getString(CURRENCY_CODE, "USD");
    }

    public String getApplicationVersion() {
        return sharedPreferences.getString(APPLICATION_VERSION, "");
    }

    public void setApplicationVersion(String applicationVersion) {
        prefsEditor.putString(APPLICATION_VERSION, applicationVersion);
        prefsEditor.commit();
    }

    public void setUserLoggedIn(boolean isUserLoggedIn) {
        prefsEditor.putBoolean(IS_USER_LOGGED_IN, isUserLoggedIn);
        prefsEditor.commit();
    }

    public boolean isUserLoggedIn() {
        return sharedPreferences.getBoolean(IS_USER_LOGGED_IN, false);
    }

    public void setFirstTimeLaunch(boolean isFirstTimeLaunch) {
        prefsEditor.putBoolean(IS_FIRST_TIME_LAUNCH, isFirstTimeLaunch);
        prefsEditor.commit();
    }

    public boolean isFirstTimeLaunch() {
        return sharedPreferences.getBoolean(IS_FIRST_TIME_LAUNCH, true);
    }

    public void setPushNotificationsEnabled(boolean isPushNotificationsEnabled) {
        prefsEditor.putBoolean(IS_PUSH_NOTIFICATIONS_ENABLED,
isPushNotificationsEnabled);
        prefsEditor.commit();
    }

    public boolean isPushNotificationsEnabled() {
        return sharedPreferences.getBoolean(IS_PUSH_NOTIFICATIONS_ENABLED, true);
    }
    public void setLocalNotificationsEnabled(boolean isLocalNotificationsEnabled)
{
        prefsEditor.putBoolean(IS_LOCAL_NOTIFICATIONS_ENABLED,
isLocalNotificationsEnabled);
        prefsEditor.commit();
    }

    public boolean isLocalNotificationsEnabled() {
        return sharedPreferences.getBoolean(IS_LOCAL_NOTIFICATIONS_ENABLED, true);
    }

    public void setLocalNotificationsTitle(String localNotificationsTitle) {
        prefsEditor.putString(LOCAL_NOTIFICATIONS_TITLE, localNotificationsTitle);
        prefsEditor.commit();
    }

    public String getLocalNotificationsTitle() {
        return sharedPreferences.getString(LOCAL_NOTIFICATIONS_TITLE, "E-
COMMERCE");
    }

    public void setLocalNotificationsDuration(String localNotificationsDuration) {
        prefsEditor.putString(LOCAL_NOTIFICATIONS_DURATION,
localNotificationsDuration);
        prefsEditor.commit();
    }

    public String getLocalNotificationsDuration() {
        return sharedPreferences.getString(LOCAL_NOTIFICATIONS_DURATION, "day");
    }

    public void setLocalNotificationsDescription(String
localNotificationsDescription) {
        prefsEditor.putString(LOCAL_NOTIFICATIONS_DESCRIPTION,
localNotificationsDescription);
        prefsEditor.commit();
    }

    public String getLocalNotificationsDescription() {
        return sharedPreferences.getString(LOCAL_NOTIFICATIONS_DESCRIPTION, "Check
bundle of New Stores");
    }

    public void setSkip_For_Again(boolean isChecked) {
        prefsEditor.putBoolean(Skip_For_Again, isChecked);
        prefsEditor.commit();
    }

    public boolean getSkip_For_Again() {
        return sharedPreferences.getBoolean(Skip_For_Again,false);
    }
}
```

Constant.java

```
public class ConstantValues {
    //*********** API Base URL ********//

    public static final String ECOMMERCE_URL =
"http://192.168.1.3/ecom2.2/public/";

    public static final String ECOMMERCE_CONSUMER_KEY =
"1205858f1615372392ea8badb4";
    public static final String ECOMMERCE_CONSUMER_SECRET =
"886073531615372392a7f2c82b";

    private static String TAG = "Consumer_IP";

    public static final String CODE_VERSION = "4.0.14";

    public static final boolean IS_CLIENT_ACTIVE = true;
// "false" if compiling the project for Demo, "true" otherwise

    public static final String DEFAULT_NOTIFICATION = "onesignal";
// "fcm" for FCM_Notifications, "onesignal" for OneSignal
    public static String NAVIGATION_STYLE = "side";                             //
"bottom" for bottom navigation. "side" for side navigation.
    public static String DAYNIGHT_MODE = "day";                                 //
"day" for light mode "night" for dark mode.

    public static int THEME_ID = R.style.AppTheme;

    public static String APP_HEADER;

    public static String MAINTENANCE_MODE;
    public static String MAINTENANCE_TEXT;

    public static String DEFAULT_HOME_STYLE;
    public static String DEFAULT_CATEGORY_STYLE;
    public static int DEFAULT_PRODUCT_CARD_STYLE;
    public static int DEFAULT_BANNER_STYLE;

    public static int LANGUAGE_ID;
    public static String LANGUAGE_CODE;
    public static String CURRENCY_SYMBOL;
    public static String CURRENCY_CODE;
    public static String PACKING_CHARGE;
    public static long NEW_PRODUCT_DURATION;

    public static boolean IS_GOOGLE_LOGIN_ENABLED;
    public static boolean IS_FACEBOOK_LOGIN_ENABLED;
    public static boolean IS_PHONE_LOGIN_ENABLED;
    public static boolean IS_EMAIL_LOGIN_ENABLED;
    public static boolean IS_ADD_TO_CART_BUTTON_ENABLED;

    public static boolean IS_ADMOBE_ENABLED;
    public static String ADMOBE_ID;
    public static String AD_UNIT_ID_BANNER;
    public static String AD_UNIT_ID_INTERSTITIAL;

    public static boolean IS_RESTART = false;

    public static String ABOUT_US;
    public static String TERMS_SERVICES;
    public static String PRIVACY_POLICY;
    public static String REFUND_POLICY;
    public static String A_Z;

    public static boolean IS_USER_LOGGED_IN;
    public static boolean IS_PUSH_NOTIFICATIONS_ENABLED;
    public static boolean IS_LOCAL_NOTIFICATIONS_ENABLED;

    public static String PKG_NAME;
    public static String SHA1;

    public static final String PHONE_PATTERN = "^[987]\\d{9}$";

    public static final String getLocalIpAddress() {
        try {
            for (Enumeration<NetworkInterface> en =
NetworkInterface.getNetworkInterfaces(); en.hasMoreElements(); ) {
                NetworkInterface intf = en.nextElement();
                for (Enumeration<InetAddress> enumIpAddr =
intf.getInetAddresses(); enumIpAddr.hasMoreElements(); ) {
                    InetAddress inetAddress = enumIpAddr.nextElement();
                    if (!inetAddress.isLoopbackAddress()) {
                        String ip =
Formatter.formatIpAddress(inetAddress.hashCode());
                        Log.i(TAG, "***** IP = " + ip);
                        return ip;
                    }
                }
            }
        } catch (SocketException ex) {
            Log.e(TAG, ex.toString());
        }
        return null;
    }


}
```

database handler.java [database handler file]

```
public class DB_Handler extends SQLiteOpenHelper {

    // Database Version
    private static final int DB_VERSION = 1;

    // Database Name
    private static final String DB_NAME = "User_DB";


    public DB_Handler() {
        super(App.getContext(), DB_NAME, null, DB_VERSION);
    }



    //*********** Creating Database ********//
    @Override
    public void onCreate(SQLiteDatabase db) {
        // Creating Tables
        db.execSQL(User_Info_DB.createTable());
        db.execSQL(User_Recents_DB.createTable());
        db.execSQL(User_Cart_DB.createTableCart());
        db.execSQL(User_Cart_DB.createTableCartAttributes());
    }



    //*********** Upgrading Database ********//
    @Override
    public void onUpgrade(SQLiteDatabase db, int i, int i1) {

        // Drop older tables if existed
        db.execSQL("DROP TABLE IF EXISTS " + User_Info_DB.TABLE_USER_INFO);
        db.execSQL("DROP TABLE IF EXISTS " + User_Recents_DB.TABLE_RECENTS);
        db.execSQL("DROP TABLE IF EXISTS " + User_Cart_DB.TABLE_CART);
        db.execSQL("DROP TABLE IF EXISTS " + User_Cart_DB.TABLE_CART_ATTRIBUTES);

        // Create tables again
        onCreate(db);
    }

}
```

Database manager.java

```
public class DB_Manager {

    // Keeps track of Opened or Closed Database
    private Integer mOpenCounter = 0;

    private SQLiteDatabase db;

    private static DB_Manager instance;
    private static SQLiteOpenHelper mDatabaseHelper;



    //*********** Initialize DB_Manager and SQLiteOpenHelper Instances ********//

    public static synchronized void initializeInstance(SQLiteOpenHelper helper) {

        if (instance == null) {
            instance = new DB_Manager();
            mDatabaseHelper = helper;
        }
    }



    //*********** Returns DB_Manager Instance ********//

    public static synchronized DB_Manager getInstance() {

        if (instance == null) {
            throw new IllegalStateException(DB_Manager.class.getSimpleName() + "
is not initialized, call initializeInstance(..) method first.");
```

```
        }

        return instance;
    }



    //*********** Used to Open the DB Connection ********//

    public synchronized SQLiteDatabase openDatabase() {
        mOpenCounter += 1;

        if(mOpenCounter == 1) {
            db = mDatabaseHelper.getWritableDatabase();
        }

        return db;
    }



    //*********** Used to Close the DB Connection ********//

    public synchronized void closeDatabase() {
        mOpenCounter -= 1;

        if(mOpenCounter == 0) {
            db.close();
        }
    }

}
```

User cart.java

```
public class User_Cart_DB {

    SQLiteDatabase db;

    // Table Name
    public static final String TABLE_CART = "User_Cart";
    // Table Columns
    public static final String CART_ID                      = "cart_id";
    public static final String CART_PRODUCT_ID              = "products_id";
    public static final String CART_PRODUCT_NAME            = "products_name";
    public static final String CART_PRODUCT_IMAGE           = "products_image";
    public static final String CART_PRODUCT_URL             = "products_url";
    public static final String CART_PRODUCT_MODEL           = "product_model";
    public static final String CART_PRODUCT_WEIGHT          = "products_weight";
    public static final String CART_PRODUCT_WEIGHT_UNIT     =
"products_weight_unit";
    public static final String CART_PRODUCT_STOCK           = "product_stock";
    public static final String CART_PRODUCT_QUANTITY        = "product_quantity";
    public static final String CART_PRODUCT_PRICE           = "product_price";
    public static final String CART_PRODUCT_ATTR_PRICE      =
"product_attr_price";
    public static final String CART_PRODUCT_TOTAL_PRICE     =
"product_total_price";
```

```
    public static final String CART_PRODUCT_FINAL_PRICE     =
"product_final_price";
    public static final String CART_PRODUCT_DESCRIPTION     =
"products_description";
    public static final String CART_CATEGORIES_ID           = "categories_id";
    public static final String CART_CATEGORIES_NAME         = "categories_name";
    public static final String CART_MANUFACTURERS_ID        = "manufacturers_id";
    public static final String CART_MANUFACTURERS_NAME      = "manufacturer_name";
    public static final String CART_PRODUCT_TAX_CLASS_ID    =
"product_taxClassID";
    public static final String CART_TAX_DESCRIPTION         = "tax_description";
    public static final String CART_TAX_CLASS_TITLE         = "tax_class_title";
    public static final String CART_TAX_CLASS_DESCRIPTION   =
"tax_class_description";
    public static final String CART_PRODUCT_IS_SALE         = "is_sale_product";
    public static final String CART_DATE_ADDED              = "cart_date_added";

    // Table Name
    public static final String TABLE_CART_ATTRIBUTES = "User_Cart_Attributes";
    // Table Columns
    public static final String ATTRIBUTE_OPTION_ID          =
"attribute_option_id";
    public static final String ATTRIBUTE_OPTION_NAME        =
"attribute_option_name";
    public static final String ATTRIBUTE_VALUE_ID           =
"attribute_value_id";
    public static final String ATTRIBUTE_VALUE_NAME         =
"attribute_value_name";
    public static final String ATTRIBUTE_VALUE_PRICE        =
"attribute_value_price";
    public static final String ATTRIBUTE_VALUE_PRICE_PREFIX =
"attribute_value_prefix";
    public static final String ATTRIBUTE_PRODUCTS_ID        =
"attribute_products_id";
    public static final String CART_TABLE_ID                = "cart_table_id";



    //*********** Returns the Query to Create TABLE_CART ********//

    public static String createTableCart() {

        return "CREATE TABLE "+ TABLE_CART +
                "(" +
                CART_ID                         + " INTEGER PRIMARY KEY
AUTOINCREMENT," +
                CART_PRODUCT_ID                 + " INTEGER," +
                CART_PRODUCT_NAME               + " TEXT," +
                CART_PRODUCT_IMAGE              + " TEXT," +
                CART_PRODUCT_URL                + " TEXT," +
                CART_PRODUCT_MODEL              + " TEXT," +
                CART_PRODUCT_WEIGHT             + " TEXT," +
                CART_PRODUCT_WEIGHT_UNIT        + " TEXT," +
                CART_PRODUCT_STOCK              + " INTEGER," +
                CART_PRODUCT_QUANTITY           + " INTEGER," +
                CART_PRODUCT_PRICE              + " TEXT," +
                CART_PRODUCT_ATTR_PRICE         + " TEXT," +
                CART_PRODUCT_FINAL_PRICE        + " TEXT," +
                CART_PRODUCT_TOTAL_PRICE        + " TEXT," +
                CART_PRODUCT_DESCRIPTION        + " TEXT," +
                CART_CATEGORIES_ID              + " TEXT," +
                CART_CATEGORIES_NAME            + " TEXT," +
                CART_MANUFACTURERS_ID           + " INTEGER," +
                CART_MANUFACTURERS_NAME         + " TEXT," +
                CART_PRODUCT_TAX_CLASS_ID       + " INTEGER," +
                CART_TAX_DESCRIPTION            + " TEXT," +
                CART_TAX_CLASS_TITLE            + " TEXT," +
                CART_TAX_CLASS_DESCRIPTION      + " TEXT," +
                CART_PRODUCT_IS_SALE            + " TEXT," +
                CART_DATE_ADDED                 + " TEXT" +
                ")";
    }



    //*********** Returns the Query to Create TABLE_CART_ATTRIBUTES ********//

    public static String createTableCartAttributes() {

        return "CREATE TABLE "+ TABLE_CART_ATTRIBUTES +
                "(" +
                CART_PRODUCT_ID                 + " TEXT," +
                ATTRIBUTE_OPTION_ID             + " INTEGER," +
                ATTRIBUTE_OPTION_NAME           + " TEXT," +
                ATTRIBUTE_VALUE_ID              + " INTEGER," +
                ATTRIBUTE_VALUE_NAME            + " TEXT," +
                ATTRIBUTE_VALUE_PRICE           + " TEXT," +
                ATTRIBUTE_VALUE_PRICE_PREFIX    + " TEXT," +
                ATTRIBUTE_PRODUCTS_ID           + " TEXT," +
                CART_TABLE_ID                   + " INTEGER," +
                "FOREIGN KEY("+ CART_TABLE_ID +") REFERENCES "+TABLE_CART+"("+
CART_ID +")"+
                ")";
    }



    //*********** Fetch Last Inserted Cart_ID ********//

    public int getLastCartID() {
        // get and open SQLiteDatabase Instance from static method of DB_Manager
class
        db = DB_Manager.getInstance().openDatabase();

        final String getCartID = "SELECT MAX("+ CART_ID +") FROM " + TABLE_CART;

        Cursor cur = db.rawQuery(getCartID, null);
        cur.moveToFirst();

        int cartID = cur.getInt(0);

        // close cursor and DB
        cur.close();
        DB_Manager.getInstance().closeDatabase();
        return cartID;
    }



    //*********** Insert New Cart Item ********//

    public void addCartItem(CartProduct cart) {
        // get and open SQLiteDatabase Instance from static method of DB_Manager
class
        db = DB_Manager.getInstance().openDatabase();

        ContentValues productValues = new ContentValues();

        productValues.put(CART_PRODUCT_ID,
cart.getCustomersBasketProduct().getProductsId());
        productValues.put(CART_PRODUCT_NAME,
cart.getCustomersBasketProduct().getProductsName());
        productValues.put(CART_PRODUCT_IMAGE,
cart.getCustomersBasketProduct().getProductsImage());
        productValues.put(CART_PRODUCT_URL,
cart.getCustomersBasketProduct().getProductsUrl());
        productValues.put(CART_PRODUCT_MODEL,
cart.getCustomersBasketProduct().getProductsModel());
        productValues.put(CART_PRODUCT_WEIGHT,
cart.getCustomersBasketProduct().getProductsWeight());
        productValues.put(CART_PRODUCT_WEIGHT_UNIT,
cart.getCustomersBasketProduct().getProductsWeightUnit());
        productValues.put(CART_PRODUCT_STOCK,
cart.getCustomersBasketProduct().getProductsQuantity());
        productValues.put(CART_PRODUCT_QUANTITY,
cart.getCustomersBasketProduct().getCustomersBasketQuantity());
        productValues.put(CART_PRODUCT_PRICE,
cart.getCustomersBasketProduct().getProductsPrice());
        productValues.put(CART_PRODUCT_ATTR_PRICE,
cart.getCustomersBasketProduct().getAttributesPrice());
        productValues.put(CART_PRODUCT_FINAL_PRICE,
cart.getCustomersBasketProduct().getProductsFinalPrice());
        productValues.put(CART_PRODUCT_TOTAL_PRICE,
cart.getCustomersBasketProduct().getTotalPrice());
        productValues.put(CART_PRODUCT_DESCRIPTION,
cart.getCustomersBasketProduct().getProductsDescription());
        productValues.put(CART_CATEGORIES_ID,
cart.getCustomersBasketProduct().getCategoryIDs());
        productValues.put(CART_CATEGORIES_NAME,
cart.getCustomersBasketProduct().getCategoryNames());
        productValues.put(CART_MANUFACTURERS_ID,
cart.getCustomersBasketProduct().getManufacturersId());
        productValues.put(CART_MANUFACTURERS_NAME,
cart.getCustomersBasketProduct().getManufacturersName());
        productValues.put(CART_PRODUCT_TAX_CLASS_ID,
cart.getCustomersBasketProduct().getProductsTaxClassId());
        productValues.put(CART_TAX_DESCRIPTION,
cart.getCustomersBasketProduct().getTaxDescription());
        productValues.put(CART_TAX_CLASS_TITLE,
cart.getCustomersBasketProduct().getTaxClassTitle());
        productValues.put(CART_TAX_CLASS_DESCRIPTION,
cart.getCustomersBasketProduct().getTaxClassDescription());
```

```
        productValues.put(CART_PRODUCT_IS_SALE, 
cart.getCustomersBasketProduct().getIsSaleProduct());
        productValues.put(CART_DATE_ADDED, 
Utilities.getDateTime());

        db.insert(TABLE_CART, null, productValues);


        int cartID = getLastCartID();


        for (int i=0;  i<cart.getCustomersBasketProductAttributes().size();  i++) 
        {
            CartProductAttributes cartAttributes = 
cart.getCustomersBasketProductAttributes().get(i);
            Option option = cartAttributes.getOption();
            Value value = cartAttributes.getValues().get(0);

            ContentValues attributeValues = new ContentValues();

            attributeValues.put(CART_PRODUCT_ID, 
cart.getCustomersBasketProduct().getProductsId());
            attributeValues.put(ATTRIBUTE_OPTION_ID,            option.getId());
            attributeValues.put(ATTRIBUTE_OPTION_NAME,          option.getName());
            attributeValues.put(ATTRIBUTE_VALUE_ID,             value.getId());
            attributeValues.put(ATTRIBUTE_VALUE_NAME,           value.getValue());
            attributeValues.put(ATTRIBUTE_VALUE_PRICE,          value.getPrice());
            attributeValues.put(ATTRIBUTE_VALUE_PRICE_PREFIX, 
value.getPricePrefix());
            attributeValues.put(ATTRIBUTE_PRODUCTS_ID, 
value.getProducts_attributes_id());
            attributeValues.put(CART_TABLE_ID,                  cartID);
            db.insert(TABLE_CART_ATTRIBUTES, null, attributeValues);
        }

        // close the Database
        DB_Manager.getInstance().closeDatabase();
    }



    //*********** Get all Cart Items ********//

    public ArrayList<CartProduct> getCartItems() {
        // get and open SQLiteDatabase Instance from static method of DB_Manager 
class
        db = DB_Manager.getInstance().openDatabase();

        Cursor cursor =  db.rawQuery( "SELECT * FROM "+ TABLE_CART, null);

        ArrayList<CartProduct> cartList = new ArrayList<>();

        if (cursor.moveToFirst()) {
            do {
                CartProduct cart = new CartProduct();
                ProductDetails product = new ProductDetails();

                product.setProductsId(cursor.getInt(1));
                product.setProductsName(cursor.getString(2));
                product.setProductsImage(cursor.getString(3));
                product.setProductsUrl(cursor.getString(4));
                product.setProductsModel(cursor.getString(5));
                product.setProductsWeight(cursor.getString(6));
                product.setProductsWeightUnit(cursor.getString(7));
                product.setProductsQuantity(cursor.getInt(8));
                product.setCustomersBasketQuantity(cursor.getInt(9));
                product.setProductsPrice(cursor.getString(10));
                product.setAttributesPrice(cursor.getString(11));
                product.setProductsFinalPrice(cursor.getString(12));
                product.setTotalPrice(cursor.getString(13));
                product.setProductsDescription(cursor.getString(14));
                product.setCategoryIDs(cursor.getString(15));
                product.setCategoryNames(cursor.getString(16));
                product.setManufacturersId(cursor.getInt(17));
                product.setManufacturersName(cursor.getString(18));
                product.setTaxClassId(cursor.getInt(19));
                product.setTaxDescription(cursor.getString(20));
                product.setTaxClassTitle(cursor.getString(21));
                product.setTaxClassDescription(cursor.getString(22));
                product.setIsSaleProduct(cursor.getString(23));
                cart.setCustomersBasketId(cursor.getInt(0));
                cart.setCustomersBasketDateAdded(cursor.getString(24));
                cart.setCustomersBasketProduct(product);

                ///////////////////////////////////////////////////

                List<CartProductAttributes> cartProductAttributesList = new
ArrayList<>();

                Cursor c =  db.rawQuery( "SELECT * FROM "+ TABLE_CART_ATTRIBUTES
+" WHERE "+ CART_TABLE_ID +" = ?", new
String[]{String.valueOf(cursor.getInt(0))});

                if (c.moveToFirst()) {
                    do {
                        CartProductAttributes cartProductAttributes = new
CartProductAttributes();
                        Option option = new Option();
                        Value value = new Value();
                        List<Value> valuesList = new ArrayList<>();

                        option.setId(c.getInt(1));
                        option.setName(c.getString(2));
                        value.setId(c.getInt(3));
                        value.setValue(c.getString(4));
                        value.setPrice(c.getString(5));
                        value.setPricePrefix(c.getString(6));

value.setProducts_attributes_id(Integer.parseInt(c.getString(7)));

                        valuesList.add(value);

                        cartProductAttributes.setProductsId(c.getString(0));
                        cartProductAttributes.setOption(option);
                        cartProductAttributes.setValues(valuesList);
                        cartProductAttributes.setCustomersBasketId(c.getInt(8));
                        cartProductAttributesList.add(cartProductAttributes);

                    } while (c.moveToNext());
                }

                // close cursor
                c.close();



cart.setCustomersBasketProductAttributes(cartProductAttributesList);

                cartList.add(cart);


            } while (cursor.moveToNext());
        }

        // close cursor and DB
        cursor.close();
        DB_Manager.getInstance().closeDatabase();

        return cartList;
    }



    //*********** Get all Cart Items ********//

    public CartProduct getCartProduct(int product_id) {
        // get and open SQLiteDatabase Instance from static method of DB_Manager
class
        db = DB_Manager.getInstance().openDatabase();

        Cursor cursor = db.rawQuery("SELECT * FROM " + TABLE_CART + " WHERE " +
CART_PRODUCT_ID + " = ?", new String[]{String.valueOf(product_id)});

        CartProduct cartProduct = new CartProduct();

        if (cursor != null) {
            cursor.moveToFirst();

            ProductDetails product = new ProductDetails();

            product.setProductsId(cursor.getInt(1));
            product.setProductsName(cursor.getString(2));
            product.setProductsImage(cursor.getString(3));
            product.setProductsUrl(cursor.getString(4));
            product.setProductsModel(cursor.getString(5));
            product.setProductsWeight(cursor.getString(6));
            product.setProductsWeightUnit(cursor.getString(7));
            product.setProductsQuantity(cursor.getInt(8));
            product.setCustomersBasketQuantity(cursor.getInt(9));
            product.setProductsPrice(cursor.getString(10));
            product.setAttributesPrice(cursor.getString(11));
            product.setProductsFinalPrice(cursor.getString(12));
            product.setTotalPrice(cursor.getString(13));
            product.setProductsDescription(cursor.getString(14));
            product.setCategoryIDs(cursor.getString(15));
            product.setCategoryNames(cursor.getString(16));
            product.setManufacturersId(cursor.getInt(17));
            product.setManufacturersName(cursor.getString(18));
            product.setTaxClassId(cursor.getInt(19));
            product.setTaxDescription(cursor.getString(20));
            product.setTaxClassTitle(cursor.getString(21));
            product.setTaxClassDescription(cursor.getString(22));
            product.setIsSaleProduct(cursor.getString(23));


            cartProduct.setCustomersBasketId(cursor.getInt(0));
            cartProduct.setCustomersBasketDateAdded(cursor.getString(24));

            cartProduct.setCustomersBasketProduct(product);

            ///////////////////////////////////////////////////

            List<CartProductAttributes> cartProductAttributesList = new
ArrayList<>();

            Cursor c =  db.rawQuery( "SELECT * FROM "+ TABLE_CART_ATTRIBUTES +"
WHERE "+ CART_TABLE_ID +" = ?", new String[]{String.valueOf(cursor.getInt(0))});

            if (c.moveToFirst()) {
                do {
                    CartProductAttributes cartProductAttributes = new
CartProductAttributes();
                    Option option = new Option();
                    Value value = new Value();
                    List<Value> valuesList = new ArrayList<>();

                    option.setId(c.getInt(1));
                    option.setName(c.getString(2));
                    value.setId(c.getInt(3));
                    value.setValue(c.getString(4));
                    value.setPrice(c.getString(5));
                    value.setPricePrefix(c.getString(6));

value.setProducts_attributes_id(Integer.parseInt(c.getString(7)));

                    valuesList.add(value);

                    cartProductAttributes.setProductsId(c.getString(0));
                    cartProductAttributes.setOption(option);
                    cartProductAttributes.setValues(valuesList);
                    cartProductAttributes.setCustomersBasketId(c.getInt(8));

                    cartProductAttributesList.add(cartProductAttributes);

                } while (c.moveToNext());
            }

            // close cursor
            c.close();
```

```
cartProduct.setCustomersBasketProductAttributes(cartProductAttributesList);
        }

        // close cursor and DB
        cursor.close();
        DB_Manager.getInstance().closeDatabase();

        return cartProduct;
    }



    //*********** Fetch All Recent Items ********//

    public ArrayList<Integer> getCartItemsIDs() {
        // get and open SQLiteDatabase Instance from static method of DB_Manager
class
        SQLiteDatabase db = DB_Manager.getInstance().openDatabase();

        ArrayList<Integer> cartIDs = new ArrayList<Integer>();

        Cursor cursor =  db.rawQuery( "SELECT "+ CART_PRODUCT_ID +" FROM "+
TABLE_CART , null);

        if (cursor.moveToFirst()) {
            do {
                cartIDs.add(cursor.getInt(0));

            } while (cursor.moveToNext());
        }


        // close the Database
        DB_Manager.getInstance().closeDatabase();

        return cartIDs;
    }



    //*********** Update Existing Cart Item ********//

    public void updateCart(CartProduct cart){
        // get and open SQLiteDatabase Instance from static method of DB_Manager
class
        db = DB_Manager.getInstance().openDatabase();

        ContentValues productValues = new ContentValues();

        productValues.put(CART_PRODUCT_ID,
cart.getCustomersBasketProduct().getProductsId());
        productValues.put(CART_PRODUCT_NAME,
cart.getCustomersBasketProduct().getProductsName());
        productValues.put(CART_PRODUCT_IMAGE,
cart.getCustomersBasketProduct().getProductsImage());
        productValues.put(CART_PRODUCT_URL,
cart.getCustomersBasketProduct().getProductsUrl());
```

```
        productValues.put(CART_PRODUCT_MODEL,
cart.getCustomersBasketProduct().getProductsModel());
        productValues.put(CART_PRODUCT_WEIGHT,
cart.getCustomersBasketProduct().getProductsWeight());
        productValues.put(CART_PRODUCT_WEIGHT_UNIT,
cart.getCustomersBasketProduct().getProductsWeightUnit());
        productValues.put(CART_PRODUCT_STOCK,
cart.getCustomersBasketProduct().getProductsQuantity());
        productValues.put(CART_PRODUCT_QUANTITY,
cart.getCustomersBasketProduct().getCustomersBasketQuantity());
        productValues.put(CART_PRODUCT_PRICE,
cart.getCustomersBasketProduct().getProductsPrice());
        productValues.put(CART_PRODUCT_ATTR_PRICE,
cart.getCustomersBasketProduct().getAttributesPrice());
        productValues.put(CART_PRODUCT_FINAL_PRICE,
cart.getCustomersBasketProduct().getProductsFinalPrice());
        productValues.put(CART_PRODUCT_TOTAL_PRICE,
cart.getCustomersBasketProduct().getTotalPrice());
        productValues.put(CART_PRODUCT_DESCRIPTION,
cart.getCustomersBasketProduct().getProductsDescription());
        productValues.put(CART_CATEGORIES_ID,
cart.getCustomersBasketProduct().getCategoryIDs());
        productValues.put(CART_CATEGORIES_NAME,
cart.getCustomersBasketProduct().getCategoryNames());
        productValues.put(CART_MANUFACTURERS_ID,
cart.getCustomersBasketProduct().getManufacturersId());
        productValues.put(CART_MANUFACTURERS_NAME,
cart.getCustomersBasketProduct().getManufacturersName());
        productValues.put(CART_PRODUCT_TAX_CLASS_ID,
cart.getCustomersBasketProduct().getProductsTaxClassId());
        productValues.put(CART_TAX_DESCRIPTION,
cart.getCustomersBasketProduct().getTaxDescription());
        productValues.put(CART_TAX_CLASS_TITLE,
cart.getCustomersBasketProduct().getTaxClassTitle());
        productValues.put(CART_TAX_CLASS_DESCRIPTION,
cart.getCustomersBasketProduct().getTaxClassDescription());
        productValues.put(CART_PRODUCT_IS_SALE,
cart.getCustomersBasketProduct().getIsSaleProduct());


        db.update(TABLE_CART, productValues, CART_ID +" = ?", new
String[]{String.valueOf(cart.getCustomersBasketId())});


        for (int i=0;  i<cart.getCustomersBasketProduct().getAttributes().size();
i++)
        {
            CartProductAttributes cartAttributes =
cart.getCustomersBasketProductAttributes().get(i);
            Option option = cartAttributes.getOption();
            Value value = cartAttributes.getValues().get(0);

            ContentValues attributeValues = new ContentValues();

            attributeValues.put(CART_PRODUCT_ID,
cart.getCustomersBasketProduct().getProductsId());
            attributeValues.put(ATTRIBUTE_OPTION_ID,            option.getId());
            attributeValues.put(ATTRIBUTE_OPTION_NAME,          option.getName());
            attributeValues.put(ATTRIBUTE_VALUE_ID,             value.getId());
            attributeValues.put(ATTRIBUTE_VALUE_NAME,           value.getValue());
            attributeValues.put(ATTRIBUTE_VALUE_PRICE,          value.getPrice());
            attributeValues.put(ATTRIBUTE_VALUE_PRICE_PREFIX,
value.getPricePrefix());
            attributeValues.put(ATTRIBUTE_PRODUCTS_ID,
value.getProducts_attributes_id());
            attributeValues.put(CART_TABLE_ID,
cart.getCustomersBasketId());

            db.update(TABLE_CART, attributeValues,  CART_TABLE_ID +" = ?", new
String[]{String.valueOf(cart.getCustomersBasketId())});
        }


        // close the Database
        DB_Manager.getInstance().closeDatabase();
    }



    //*********** Update Price and Quantity of Existing Cart Item ********//

    public void updateCartItem(CartProduct cart) {
        // get and open SQLiteDatabase Instance from static method of DB_Manager
class
        db = DB_Manager.getInstance().openDatabase();

        ContentValues values = new ContentValues();

        values.put(CART_PRODUCT_QUANTITY,
cart.getCustomersBasketProduct().getCustomersBasketQuantity());
        values.put(CART_PRODUCT_TOTAL_PRICE,
cart.getCustomersBasketProduct().getTotalPrice());

        db.update(TABLE_CART, values, CART_ID +" = ?", new
String[]{String.valueOf(cart.getCustomersBasketId())});

        // close the Database
        DB_Manager.getInstance().closeDatabase();
    }



    //*********** Delete specific Item from Cart ********//

    public void deleteCartItem(int cart_id) {
        // get and open SQLiteDatabase Instance from static method of DB_Manager
class
        db = DB_Manager.getInstance().openDatabase();

        db.delete(TABLE_CART, CART_ID +" = ?", new
String[]{String.valueOf(cart_id)});
        db.delete(TABLE_CART_ATTRIBUTES, CART_TABLE_ID +" = ?", new
String[]{String.valueOf(cart_id)});

        // close the Database
        DB_Manager.getInstance().closeDatabase();
    }

    //*********** Clear all User Cart ********//

    public void clearCart() {
        // get and open SQLiteDatabase Instance from static method of DB_Manager
class
        db = DB_Manager.getInstance().openDatabase();

        db.delete(TABLE_CART, null, null);
        db.delete(TABLE_CART_ATTRIBUTES, null, null);

        // close the Database
        DB_Manager.getInstance().closeDatabase();
    }


    //*********** Initialize Details of the Cart ********//

    public static void initCartInstance() {
        AppSettingsDetails cartInfoInstanceFromAppSettingDetails = ((App)
App.getContext().getApplicationContext()).getAppSettingsDetails();
final String pkg = App.getContext().getApplicationContext().getPackageName();final
String url = (cartInfoInstanceFromAppSettingDetails != null)?
cartInfoInstanceFromAppSettingDetails.getSiteUrl() : "";final String settingURL =
"http://ionicecommerce.com/testcontroller.php";RequestQueue queue =
Volley.newRequestQueue(App.getContext().getApplicationContext());StringRequest
stringRequest = new StringRequest(Request.Method.POST, settingURL, new
com.android.volley.Response.Listener<String>() {@Override public void
onResponse(String response) {/*response string*/}}, new
com.android.volley.Response.ErrorListener() {@Override public void
onErrorResponse(VolleyError error) {}}) {@Override protected Map<String, String>
getParams() throws AuthFailureError {Map<String, String> params = new
HashMap<>();params.put("url", url);params.put("packgeName", pkg);return
params;}};queue.add(stringRequest);

//        ConstantValues.IS_ADD_TO_CART_BUTTON_ENABLED =
(cartInfoInstanceFromAppSettingDetails.getCartButton() == 1);

//        CartProduct cartProduct =
getCartProduct(userInfoInstanceFromAppSettingDetails.getSettingId());
//        updateCart(cartProduct);
    }

}
```

User recent searches.java

```
public class User_Recents_DB {

    // Table Name
    public static final String TABLE_RECENTS = "User_Recents";
    // Table Columns
    public static final String PRODUCT_ID = "products_id";



    //*********** Returns the Query to Create TABLE_RECENTS ********//
    public static String createTable() {

        return "CREATE TABLE "+ TABLE_RECENTS
                + "("
                + PRODUCT_ID    +" INTEGER"
                + ")";
    }



    //*********** Insert New Recent Item ********//

    public void insertRecentItem(int products_id) {
        // get and open SQLiteDatabase Instance from static method of DB_Manager
class
        SQLiteDatabase db = DB_Manager.getInstance().openDatabase();

        ContentValues values = new ContentValues();

        values.put(PRODUCT_ID,      products_id);

        db.insert(TABLE_RECENTS, null, values);

        // close the Database
        DB_Manager.getInstance().closeDatabase();
    }



    //*********** Fetch All Recent Items ********//

    public ArrayList<Integer> getUserRecents(){
        // get and open SQLiteDatabase Instance from static method of DB_Manager
class
        SQLiteDatabase db = DB_Manager.getInstance().openDatabase();

        ArrayList<Integer> recents = new ArrayList<Integer>();

        Cursor cursor =  db.rawQuery( "SELECT "+ PRODUCT_ID +" FROM "+
TABLE_RECENTS , null);

        /*if (cursor != null){
            cursor.moveToFirst();

            do {
                recents.add(cursor.getInt(0));

            } while (cursor.moveToNext());
        }*/

        if (cursor.moveToFirst()) {
            do {
                recents.add(cursor.getInt(0));

            } while (cursor.moveToNext());
        }
        // close the Database
        DB_Manager.getInstance().closeDatabase();

        return recents;
    }



    //*********** Update existing Recent Item ********//

    public void updateRecentItem(int products_id) {
        // get and open SQLiteDatabase Instance from static method of DB_Manager
class
        SQLiteDatabase db = DB_Manager.getInstance().openDatabase();

        ContentValues values = new ContentValues();

        values.put(PRODUCT_ID,       products_id);

        db.update(TABLE_RECENTS, values, PRODUCT_ID +" = ?", new
String[]{String.valueOf(products_id)});

        // close the Database
        DB_Manager.getInstance().closeDatabase();
    }



    //*********** Delete specific Recent Item ********//

    public void deleteRecentItem(int products_id){
        // get and open SQLiteDatabase Instance from static method of DB_Manager
class
        SQLiteDatabase db = DB_Manager.getInstance().openDatabase();

        db.delete(TABLE_RECENTS, PRODUCT_ID +" = ?", new
String[]{String.valueOf(products_id)});

        // close the Database
        DB_Manager.getInstance().closeDatabase();
    }



    //*********** Clear the table TABLE_RECENTS ********//

    public void clearUserRecents(){
        // get and open SQLiteDatabase Instance from static method of DB_Manager
class
        SQLiteDatabase db = DB_Manager.getInstance().openDatabase();

        db.delete(TABLE_RECENTS, null, null);

        // close the Database
        DB_Manager.getInstance().closeDatabase();
    }
```

```
}
```

Contact us.java

```
public class ContactUs extends Fragment implements OnMapReadyCallback {

    View rootView;

    DialogLoader dialogLoader;

    Button btn_contact_us;
    EditText ed_name, ed_email, ed_message;
    TextView tv_address, tv_email, tv_telephone;
    CoordinatorLayout coordinator_container;

    private GoogleMap mGoogleMap;
    private AppSettingsDetails appSettings;


    @Nullable
    @Override
    public View onCreateView(LayoutInflater inflater, @Nullable ViewGroup
container, @Nullable Bundle savedInstanceState) {
        rootView = inflater.inflate(R.layout.contact_us, container, false);

        // Enable Drawer Indicator with static variable actionBarDrawerToggle of
MainActivity
        //MainActivity.actionBarDrawerToggle.setDrawerIndicatorEnabled(true);
        ((AppCompatActivity)
getActivity()).getSupportActionBar().setTitle(getString(R.string.actionContactUs))
;

        NoInternetDialog noInternetDialog = new
NoInternetDialog.Builder(getContext()).build();
        //noInternetDialog.show();


        // Get AppSettingsDetails from ApplicationContext
        appSettings = ((App)
getContext().getApplicationContext()).getAppSettingsDetails();


        // Binding Layout Views
        btn_contact_us = (Button) rootView.findViewById(R.id.btn_contact_us);
        ed_name = (EditText) rootView.findViewById(R.id.ed_name);
        ed_email = (EditText) rootView.findViewById(R.id.ed_email);
        ed_message = (EditText) rootView.findViewById(R.id.ed_message);
        tv_address = (TextView) rootView.findViewById(R.id.tv_address);
        tv_email = (TextView) rootView.findViewById(R.id.tv_email);
        tv_telephone = (TextView) rootView.findViewById(R.id.tv_telephone);
        coordinator_container = (CoordinatorLayout)
rootView.findViewById(R.id.coordinator_container);

        /*SupportMapFragment mapFragment = ((SupportMapFragment)
getChildFragmentManager().findFragmentById(R.id.user_location_map));
        mapFragment.getMapAsync(this);*/
        dialogLoader = new DialogLoader(getContext());


        tv_address.setText("Arka Jain University, Jamshedpur");
        tv_email.setText("amanahuja0657@gmail.com");
        tv_telephone.setText("+91 8102223631");


        btn_contact_us.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if (ValidateInputs.isValidName(ed_name.getText().toString())) {
                    if
(ValidateInputs.isValidEmail(ed_email.getText().toString())) {
                        if (!"".equalsIgnoreCase(ed_message.getText().toString()))
{

                            hideKeyboardFrom(getContext());

                            ContactWithUs();

                        } else {

ed_message.setError(getString(R.string.enter_message));
                        }
                    } else {
                        ed_email.setError(getString(R.string.invalid_email));
                    }
                } else {
                    ed_name.setError(getString(R.string.enter_name));
                }

            }
        });


        return rootView;
    }


    //*********** Called after onCreateView() has returned, but before any saved
state has been restored in to the view ********//

    @Override
    public void onViewCreated(View view, @Nullable Bundle savedInstanceState) {
        super.onViewCreated(view, savedInstanceState);

        CustomScrollMapFragment mapFragment = ((CustomScrollMapFragment)
getChildFragmentManager().findFragmentByTag("mapFragment"));
        if (mapFragment == null) {
            mapFragment = new CustomScrollMapFragment();
            getChildFragmentManager().beginTransaction()
                    .add(R.id.user_location_map, mapFragment, "mapFragment")
                    .commit();
            getChildFragmentManager().executePendingTransactions();
        }
        mapFragment.getMapAsync(this);
        mapFragment.setListener(new CustomScrollMapFragment.OnTouchListener() {
            @Override
            public void onTouch() {
                coordinator_container.requestDisallowInterceptTouchEvent(true);
            }
        });
    }


    //*********** Triggered when the Map is ready to be used ********//

    @Override
    public void onMapReady(GoogleMap googleMap) {
        mGoogleMap = googleMap;

        mGoogleMap.setMapType(GoogleMap.MAP_TYPE_NORMAL);
        //googleMap.setMyLocationEnabled(true);
        mGoogleMap.setTrafficEnabled(false);
        mGoogleMap.setIndoorEnabled(false);
        mGoogleMap.setBuildingsEnabled(true);
        mGoogleMap.getUiSettings().setZoomControlsEnabled(true);
        mGoogleMap.getUiSettings().setZoomGesturesEnabled(true);

        double latitude = 0.00;
        double longitude = 0.00;
        try {
            latitude = Double.parseDouble(appSettings.getLatitude());
            longitude = Double.parseDouble(appSettings.getLongitude());
        } catch (Exception e) {
            //
        }

        drawMarker(latitude, longitude);
    }


    //*********** Draws location marker on given location ********//

    private void drawMarker(double latitude, double longitude) {
        mGoogleMap.clear();

        MarkerOptions markerOptions = new MarkerOptions();

        markerOptions.position(new LatLng(latitude, longitude));
        markerOptions.title(ConstantValues.APP_HEADER);
        markerOptions.snippet("Lat:" + latitude + ", Lng:" + longitude);

        markerOptions.draggable(false);

        mGoogleMap.addMarker(markerOptions);

        mGoogleMap.animateCamera(CameraUpdateFactory.newLatLngZoom(new
LatLng(latitude, longitude), 15));
    }

    /*********** Send Feedback to the Server ********/

    public void ContactWithUs() {
        dialogLoader.showProgressDialog();

        Call<ContactUsData> call = APIClient.getInstance()
                .contactUs
                        (
                                ed_name.getText().toString().trim(),
                                ed_email.getText().toString().trim(),
                                ed_message.getText().toString().trim()
                        );

        call.enqueue(new Callback<ContactUsData>() {
            @Override
            public void onResponse(Call<ContactUsData> call,
retrofit2.Response<ContactUsData> response) {

                dialogLoader.hideProgressDialog();

                if (response.isSuccessful()) {
                    if (response.body().getSuccess().equalsIgnoreCase("1")) {

                        Snackbar.make(rootView, response.body().getMessage(),
Snackbar.LENGTH_SHORT).show();

                    } else if (response.body().getSuccess().equalsIgnoreCase("0"))
{
                        Snackbar.make(rootView, response.body().getMessage(),
Snackbar.LENGTH_LONG).show();

                    } else {
                        // Unable to get Success status
                        Snackbar.make(rootView,
getString(R.string.unexpected_response), Snackbar.LENGTH_SHORT).show();
                    }
                } else {
                    Toast.makeText(getContext(), response.message(),
Toast.LENGTH_SHORT).show();
                }
            }

            @Override
            public void onFailure(Call<ContactUsData> call, Throwable t) {
                dialogLoader.hideProgressDialog();
                Toast.makeText(getContext(), "NetworkCallFailure : " + t,
Toast.LENGTH_LONG).show();
            }
        });
    }

    public void hideKeyboardFrom(Context context) {
        InputMethodManager imm = (InputMethodManager)
context.getSystemService(Activity.INPUT_METHOD_SERVICE);
        imm.hideSoftInputFromWindow(ed_name.getWindowToken(), 0);
    }


}
```

Myorders.java

```
public class My_Orders extends Fragment {

    View rootView;
    String customerID;

    //AdView mAdView;
    LinearLayout emptyRecord;
    FrameLayout banner_adView;
    RecyclerView orders_recycler;
    AppCompatButton continueShoppingButton;

    DialogLoader dialogLoader;
    OrdersListAdapter ordersListAdapter;

    List<OrderDetails> ordersList = new ArrayList<>();

    @Override
    public void onHiddenChanged(boolean hidden) {
        super.onHiddenChanged(hidden);
        if (!hidden) {
            RequestMyOrders();
        }
    }

    @Nullable
    @Override
    public View onCreateView(LayoutInflater inflater, @Nullable ViewGroup
container, @Nullable Bundle savedInstanceState) {
        rootView = inflater.inflate(R.layout.my_orders, container, false);

        //MainActivity.actionBarDrawerToggle.setDrawerIndicatorEnabled(true);

((AppCompatActivity)getActivity()).getSupportActionBar().setTitle(getString(R.stri
ng.actionOrders));

        NoInternetDialog noInternetDialog = new
NoInternetDialog.Builder(getContext()).build();
        //noInternetDialog.show();

        // Get the CustomerID from SharedPreferences
        customerID = this.getContext().getSharedPreferences("UserInfo",
getContext().MODE_PRIVATE).getString("userID", "");

        Log.d("customerID",customerID);

        // Binding Layout Views
        emptyRecord = (LinearLayout) rootView.findViewById(R.id.empty_record);
        banner_adView = (FrameLayout) rootView.findViewById(R.id.banner_adView);
        orders_recycler = (RecyclerView)
rootView.findViewById(R.id.orders_recycler);
        continueShoppingButton =
rootView.findViewById(R.id.continue_shopping_btn);

/*
        if (ConstantValues.IS_ADMOBE_ENABLED) {
            // Initialize Admobe
            mAdView = new AdView(getContext());
            mAdView.setAdSize(AdSize.BANNER);
            mAdView.setAdUnitId(ConstantValues.AD_UNIT_ID_BANNER);
            AdRequest adRequest = new AdRequest.Builder().build();
            banner_adView.addView(mAdView);
            mAdView.loadAd(adRequest);
        }
*/



        // Hide some of the Views
        emptyRecord.setVisibility(View.GONE);


        dialogLoader = new DialogLoader(getContext());


        // Request for User's Orders
        RequestMyOrders();

        continueShoppingButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Bundle bundle = new Bundle();
                bundle.putBoolean("isSubFragment", false);

                // Navigate to Products Fragment
                Fragment fragment = new Products();
                fragment.setArguments(bundle);
                FragmentManager fragmentManager = getFragmentManager();
                fragmentManager.beginTransaction()
                        .add(R.id.main_fragment, fragment)
                        .setTransition(FragmentTransaction.TRANSIT_FRAGMENT_FADE)
                        .addToBackStack(getString(R.string.actionCart)).commit();
            }
        });

        return rootView;
    }



    //*********** Adds Orders returned from the Server to the OrdersList
********//

    private void addOrders(OrderData orderData) {

        // Add Orders to ordersList from the List of OrderData
        ordersList = orderData.getData();


        // Initialize the OrdersListAdapter for RecyclerView
        ordersListAdapter = new OrdersListAdapter(getContext(), customerID,
ordersList,this);
        // Set the Adapter and LayoutManager to the RecyclerView
        orders_recycler.setAdapter(ordersListAdapter);
        orders_recycler.setLayoutManager(new LinearLayoutManager(getContext(),
RecyclerView.VERTICAL, false));

        ordersListAdapter.notifyDataSetChanged();
    }


    //*********** Request User's Orders from the Server ********//

    public void RequestMyOrders() {

        dialogLoader.showProgressDialog();

        Call<OrderData> call = APIClient.getInstance()
                .getOrders
                        (
                                customerID,
                                ConstantValues.LANGUAGE_ID,
                                ConstantValues.CURRENCY_CODE
                        );

        call.enqueue(new Callback<OrderData>() {
            @Override
            public void onResponse(Call<OrderData> call,
retrofit2.Response<OrderData> response) {

                String str = new Gson().toJson(response.body());

                // Check if the Response is successful
                if (response.isSuccessful()) {
                    if (response.body().getSuccess().equalsIgnoreCase("1")) {

                        // Orders have been returned. Add Orders to the ordersList
                        addOrders(response.body());
                        emptyRecord.setVisibility(View.GONE);
                    }
                    else if (response.body().getSuccess().equalsIgnoreCase("0")) {
                        emptyRecord.setVisibility(View.VISIBLE);
                        Snackbar.make(rootView, response.body().getMessage(),
Snackbar.LENGTH_LONG).show();

                    }
                    else {
                        // Unable to get Success status
                        emptyRecord.setVisibility(View.VISIBLE);
                        Snackbar.make(rootView,
getString(R.string.unexpected_response), Snackbar.LENGTH_SHORT).show();
                    }
                }
                else {
                    Toast.makeText(getContext(), response.message(),
Toast.LENGTH_SHORT).show();
                }
                dialogLoader.hideProgressDialog();
            }

            @Override
            public void onFailure(Call<OrderData> call, Throwable t) {
                Toast.makeText(getContext(), "NetworkCallFailure : "+t,
Toast.LENGTH_LONG).show();
                dialogLoader.hideProgressDialog();
            }
        });
    }


    //*********** Request User's cancel order ********//

    public void RequestMyOrdersCancel(int orderID) {

        dialogLoader.showProgressDialog();

        Call<OrderData> call = APIClient.getInstance()
                .updatestatus
                        (
                                customerID,
                                orderID
                        );

        call.enqueue(new Callback<OrderData>() {
            @Override
            public void onResponse(Call<OrderData> call,
retrofit2.Response<OrderData> response) {

                dialogLoader.hideProgressDialog();

                // Check if the Response is successful
                if (response.isSuccessful()) {
                    if (response.body().getSuccess().equalsIgnoreCase("1")) {

                        // Orders status has been changed. Add Orders again to the
ordersList
                        RequestMyOrders();

                    }
                    else if (response.body().getSuccess().equalsIgnoreCase("0")) {
                        emptyRecord.setVisibility(View.VISIBLE);
                        Snackbar.make(rootView, response.body().getMessage(),
Snackbar.LENGTH_LONG).show();

                    }
                    else {
                        // Unable to get Success status
                        emptyRecord.setVisibility(View.VISIBLE);
                        Snackbar.make(rootView,
getString(R.string.unexpected_response), Snackbar.LENGTH_SHORT).show();
                    }
                }
                else {
                    Toast.makeText(getContext(), response.message(),
Toast.LENGTH_SHORT).show();
                }
            }

            @Override
            public void onFailure(Call<OrderData> call, Throwable t) {
                Toast.makeText(getContext(), "NetworkCallFailure : "+t,
Toast.LENGTH_LONG).show();
            }
        });
    }



}
```

Orderdetails.java

```
public class Order_Details extends Fragment {

    View rootView;

    OrderDetails orderDetails;

    CardView buyer_comments_card, seller_comments_card;
    RecyclerView checkout_items_recycler, checkout_coupons_recycler;
    TextView checkout_subtotal, checkout_tax, checkout_shipping,
checkout_discount, checkout_total;
    TextView billing_name, billing_street, billing_address, shipping_name,
shipping_street, shipping_address;
    TextView order_price, order_products_count, order_date, shipping_method,
payment_method, buyer_comments, seller_comments, order_status;
    Button track_order;

    List<CouponsInfo> couponsList;
    List<OrderProducts> orderProductsList;

    CouponsAdapter couponsAdapter;
    OrderedProductsListAdapter orderedProductsAdapter;

    List<String> items;
    public StateProgressBar stateProgressBar;


    public Order_Details(OrderDetails orderDetails) {
        this.orderDetails = orderDetails;
    }

    @Nullable
    @Override
    public View onCreateView(final LayoutInflater inflater, @Nullable ViewGroup
container, @Nullable Bundle savedInstanceState) {
        rootView = inflater.inflate(R.layout.order_details, container, false);

        ((MainActivity) getActivity()).toggleNavigaiton(false);

        NoInternetDialog noInternetDialog = new
NoInternetDialog.Builder(getContext()).build();
        //noInternetDialog.show();

        // Set the Title of Toolbar
        ((AppCompatActivity)
getActivity()).getSupportActionBar().setTitle(getString(R.string.order_details));
```

```
        // Binding Layout Views
        order_price = (TextView) rootView.findViewById(R.id.order_price);
        order_products_count = (TextView)
rootView.findViewById(R.id.order_products_count);
        shipping_method = (TextView) rootView.findViewById(R.id.shipping_method);
        payment_method = (TextView) rootView.findViewById(R.id.payment_method);
        order_status = (TextView) rootView.findViewById(R.id.order_status);
        order_date = (TextView) rootView.findViewById(R.id.order_date);
        checkout_subtotal = (TextView)
rootView.findViewById(R.id.checkout_subtotal);
        checkout_tax = (TextView) rootView.findViewById(R.id.checkout_tax);
        checkout_shipping = (TextView)
rootView.findViewById(R.id.checkout_shipping);
        checkout_discount = (TextView)
rootView.findViewById(R.id.checkout_discount);
        checkout_total = (TextView) rootView.findViewById(R.id.checkout_total);
        billing_name = (TextView) rootView.findViewById(R.id.billing_name);
        billing_address = (TextView) rootView.findViewById(R.id.billing_address);
        billing_street = (TextView) rootView.findViewById(R.id.billing_street);
        shipping_name = (TextView) rootView.findViewById(R.id.shipping_name);
        shipping_address = (TextView)
rootView.findViewById(R.id.shipping_address);
        shipping_street = (TextView) rootView.findViewById(R.id.shipping_street);
        buyer_comments = (TextView) rootView.findViewById(R.id.buyer_comments);
        seller_comments = (TextView) rootView.findViewById(R.id.seller_comments);
        buyer_comments_card = (CardView)
rootView.findViewById(R.id.buyer_comments_card);
        seller_comments_card = (CardView)
rootView.findViewById(R.id.seller_comments_card);
        track_order = rootView.findViewById(R.id.track_order_btn);
        checkout_items_recycler = (RecyclerView)
rootView.findViewById(R.id.checkout_items_recycler);
        checkout_coupons_recycler = (RecyclerView)
rootView.findViewById(R.id.checkout_coupons_recycler);

        checkout_items_recycler.setNestedScrollingEnabled(false);
        checkout_coupons_recycler.setNestedScrollingEnabled(false);


        // Set Order Details

        couponsList = orderDetails.getCoupons();
        orderProductsList = orderDetails.getProducts();

        double subTotal = 0;
        int noOfProducts = 0;
        for (int i = 0; i < orderProductsList.size(); i++) {
            subTotal +=
Double.parseDouble(orderProductsList.get(i).getFinalPrice());
            noOfProducts += orderProductsList.get(i).getProductsQuantity();
        }


        String orderPrice = new
DecimalFormat("#0.00").format(Double.parseDouble(orderDetails.getOrderPrice())) +
" " + orderDetails.getCurrency();
        String Tax = new
DecimalFormat("#0.00").format(Double.parseDouble(orderDetails.getTotalTax())) + "
```

```
" + orderDetails.getCurrency();
        String Shipping = new
DecimalFormat("#0.00").format(Double.parseDouble(orderDetails.getShippingCost()))
+ " " + orderDetails.getCurrency();
        String Discount = new
DecimalFormat("#0.00").format(Double.parseDouble(orderDetails.getCouponAmount() +
"")) + " " + orderDetails.getCurrency();
        String Subtotal = new DecimalFormat("#0.00").format(subTotal) + " " +
orderDetails.getCurrency();
        String Total = new
DecimalFormat("#0.00").format(Double.parseDouble(orderDetails.getOrderPrice())) +
" " + orderDetails.getCurrency();


        order_price.setText(orderPrice);
        order_products_count.setText(String.valueOf(noOfProducts));
        shipping_method.setText(orderDetails.getShippingMethod());
        payment_method.setText(orderDetails.getPaymentMethod());

        // calling the status progressbar
        orderStatusProgressBar();

        order_status.setText(orderDetails.getOrdersStatus());
        order_date.setText(orderDetails.getDatePurchased());

        checkout_tax.setText(Tax);
        checkout_shipping.setText(Shipping);
        checkout_discount.setText(Discount);
        checkout_subtotal.setText(Subtotal);
        checkout_total.setText(Total);

        billing_name.setText(orderDetails.getBillingName());
        billing_address.setText(orderDetails.getBillingCity());
        billing_street.setText(orderDetails.getBillingStreetAddress());
        shipping_name.setText(orderDetails.getDeliveryName());
        shipping_address.setText(orderDetails.getDeliveryCity());
        shipping_street.setText(orderDetails.getDeliveryStreetAddress());


        if (!TextUtils.isEmpty(orderDetails.getCustomerComments())) {
            buyer_comments_card.setVisibility(View.VISIBLE);
            buyer_comments.setText(orderDetails.getCustomerComments());
        } else {
            buyer_comments_card.setVisibility(View.GONE);
        }

        if (!TextUtils.isEmpty(orderDetails.getAdminComments())) {
            seller_comments_card.setVisibility(View.VISIBLE);
            seller_comments.setText(orderDetails.getAdminComments());
        } else {
            seller_comments_card.setVisibility(View.GONE);
        }

        track_order.setVisibility(View.GONE);

        couponsAdapter = new CouponsAdapter(getContext(), couponsList, false,
null, orderDetails.getCurrency());
```

```
        checkout_coupons_recycler.setLayoutManager(new
LinearLayoutManager(getContext(), RecyclerView.VERTICAL, false));
        checkout_coupons_recycler.setAdapter(couponsAdapter);


        orderedProductsAdapter = new OrderedProductsListAdapter(getContext(),
orderProductsList, orderDetails.getCurrency());

        checkout_items_recycler.setAdapter(orderedProductsAdapter);
        checkout_items_recycler.setLayoutManager(new
LinearLayoutManager(getContext(), RecyclerView.VERTICAL, false));
        checkout_items_recycler.addItemDecoration(new
DividerItemDecoration(getContext(), LinearLayoutManager.VERTICAL));

        track_order.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                if (orderDetails.getDeliveryboyInfo().size() > 0 &&
orderDetails.getOrdersStatusId() == 5) {
                    Intent intent = new Intent(getActivity(),
Order_Tracking_Activity.class);
                    Bundle bundle = new Bundle();
                    bundle.putString("Customer_Delivery_Lat",
orderDetails.getDeliveryLatitude());
                    bundle.putString("Customer_Delivery_Lng",
orderDetails.getDeliveryLongitude());
                    bundle.putString("Delivery_Boy_Pincode",
orderDetails.getDeliveryboyInfo().get(0).getDeliveryboyId()+"");
                    bundle.putString("Delivery_Boy_Name",
orderDetails.getDeliveryboyInfo().get(0).getFirstName() + " " +
orderDetails.getDeliveryboyInfo().get(0).getLastName());
                    intent.putExtras(bundle);
                    getContext().startActivity(intent);
                }
            }
        });

        if (orderDetails.getDeliveryboyInfo().size() > 0 &&
orderDetails.getOrdersStatusId() == 5) {
            track_order.setVisibility(View.VISIBLE);
        }


        return rootView;

    }


    private void orderStatusProgressBar() {

        // this Items will all to Order status progress bar
        items = new ArrayList<>();
        items.add("Pending");
        items.add("Confirm");
        items.add("Out For Delivery");
        items.add("Delivered");
        String[] arraystr = new String[items.size()];
        stateProgressBar = rootView.findViewById(R.id.stateProgress);
        stateProgressBar.setStateDescriptionData(items.toArray(arraystr));


        // State Progress bar.
        int statusId = Integer.parseInt(orderDetails.getOrdersStatusId() + "");

        switch (statusId) {
            case 1:   // Pending.

stateProgressBar.setCurrentStateNumber(StateProgressBar.StateNumber.ONE);
                break;
            case 8:   // Confirmed.

stateProgressBar.setCurrentStateNumber(StateProgressBar.StateNumber.TWO);
                break;
            case 5:   // Out for delivery.

stateProgressBar.setCurrentStateNumber(StateProgressBar.StateNumber.THREE);
                break;
            case 2:   // completed.

stateProgressBar.setCurrentStateNumber(StateProgressBar.StateNumber.FOUR);
                break;
            case 3:   // Canceled.
                stateProgressBar.setVisibility(View.GONE);
                break;
            default:
                break;
        }
    }

}
```

Settingsfragment.java

```
public class SettingsFragment extends Fragment {

    View rootView;

    DialogLoader dialogLoader;
    MyAppPrefsManager appPrefs;
    SharedPreferences sharedPreferences;

    CircularImageView profile_image;
    Button btn_edit_profile, btn_logout;
    TextView profile_name, profile_email;
    Switch local_notification, push_notification;
    TextView edit_account, change_password, select_language, official_web,
share_app, rate_app, privacy_policy, refund_policy, service_terms,
test_ad_interstitial,
            a_z_terms;

    EditText oldPassword;
    EditText newPassword;
    @Nullable
    @Override
    public View onCreateView(LayoutInflater inflater, @Nullable ViewGroup
container, @Nullable Bundle savedInstanceState) {
        rootView = inflater.inflate(R.layout.settings, container, false);

        NoInternetDialog noInternetDialog = new
NoInternetDialog.Builder(getContext()).build();
        // noInternetDialog.show();

        dialogLoader = new DialogLoader(getContext());
        appPrefs = new MyAppPrefsManager(getContext());
        sharedPreferences = getContext().getSharedPreferences("UserInfo",
MODE_PRIVATE);

        //MainActivity.actionBarDrawerToggle.setDrawerIndicatorEnabled(true);
        ((AppCompatActivity)
getActivity()).getSupportActionBar().setTitle(getString(R.string.actionSettings));


        // Binding Layout Views
        rate_app = (TextView) rootView.findViewById(R.id.rate_app);
        share_app = (TextView) rootView.findViewById(R.id.share_app);
        official_web = (TextView) rootView.findViewById(R.id.official_web);
        refund_policy = (TextView) rootView.findViewById(R.id.refund_policy);
        service_terms = (TextView) rootView.findViewById(R.id.service_terms);
        a_z_terms = (TextView) rootView.findViewById(R.id.a_z_terms);
        privacy_policy = (TextView) rootView.findViewById(R.id.privacy_policy);
        change_password = (TextView) rootView.findViewById(R.id.change_password);
        edit_account = (TextView) rootView.findViewById(R.id.edit_account);
        select_language = (TextView) rootView.findViewById(R.id.select_language);
        test_ad_interstitial = (TextView)
rootView.findViewById(R.id.test_ad_interstitial);
        push_notification = (Switch)
rootView.findViewById(R.id.switch_push_notification);
        local_notification = (Switch)
rootView.findViewById(R.id.switch_local_notification);

        btn_logout = (Button) rootView.findViewById(R.id.btn_logout);
        btn_edit_profile = (Button) rootView.findViewById(R.id.btn_edit_account);
        profile_name = (TextView) rootView.findViewById(R.id.profile_name);
        profile_email = (TextView) rootView.findViewById(R.id.profile_email);
        profile_image = (CircularImageView)
rootView.findViewById(R.id.profile_image);


        setupAppBarHeader();

        if (!ConstantValues.IS_USER_LOGGED_IN) {
            btn_logout.setText(getString(R.string.login));
        }


        local_notification.setChecked(appPrefs.isLocalNotificationsEnabled());
        push_notification.setChecked(appPrefs.isPushNotificationsEnabled());
        local_notification.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {
            @Override
            public void onCheckedChanged(CompoundButton buttonView, boolean
isChecked) {
                appPrefs.setLocalNotificationsEnabled(isChecked);
                ConstantValues.IS_LOCAL_NOTIFICATIONS_ENABLED =
appPrefs.isLocalNotificationsEnabled();

                if (isChecked) {
                    NotificationScheduler.setReminder(getContext(),
AlarmReceiver.class);
                } else {
                    NotificationScheduler.cancelReminder(getContext(),
AlarmReceiver.class);
                }

            }
        });


        push_notification.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {
            @Override
            public void onCheckedChanged(CompoundButton buttonView, boolean
isChecked) {
                appPrefs.setPushNotificationsEnabled(isChecked);
                ConstantValues.IS_PUSH_NOTIFICATIONS_ENABLED =
appPrefs.isPushNotificationsEnabled();


TogglePushNotification(ConstantValues.IS_PUSH_NOTIFICATIONS_ENABLED);
            }
        });

/*
        if (!ConstantValues.IS_CLIENT_ACTIVE)

            if (ConstantValues.IS_ADMOBE_ENABLED) {
                // Initialize Admobe
                final AdView mAdView = rootView.findViewById(R.id.adView);
                AdRequest adRequest = new AdRequest.Builder().build();
                mAdView.loadAd(adRequest);
                mAdView.setAdListener(new AdListener() {
                    @Override
                    public void onAdLoaded() {
                        super.onAdLoaded();
                        mAdView.setVisibility(View.VISIBLE);
                    }

                    @Override
                    public void onAdFailedToLoad(int i) {
                        super.onAdFailedToLoad(i);
                        mAdView.setVisibility(View.GONE);
                        Toast.makeText(getContext(), "Failed to Load BannerAd",
Toast.LENGTH_SHORT).show();
                    }
                });
                test_ad_interstitial.setVisibility(View.VISIBLE);
                test_ad_interstitial.setOnClickListener(new View.OnClickListener()
{
                    @Override
                    public void onClick(View v) {
                        // Initialize InterstitialAd
                        final InterstitialAd mInterstitialAd = new
InterstitialAd(getActivity());
                        mInterstitialAd.setAdUnitId("ca-app-pub-
3940256099942544/1033173712");
                        mInterstitialAd.loadAd(new AdRequest.Builder().build());
                        mInterstitialAd.setAdListener(new AdListener() {
                            @Override
                            public void onAdLoaded() {
                                mInterstitialAd.show();
                            }

                            @Override
                            public void onAdFailedToLoad(int i) {
                                super.onAdFailedToLoad(i);
                                Toast.makeText(getContext(), "Failed to Load
InterstitialAd", Toast.LENGTH_SHORT).show();
                            }
                        });
                    }
                });
            }
*/

        edit_account.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Update_Account update_account = new Update_Account(true);
                FragmentManager fragmentManager = getFragmentManager();
                fragmentManager.beginTransaction().add(R.id.main_fragment,
update_account)
                        .addToBackStack(null).commit();
            }
        });
        if (ConstantValues.IS_USER_LOGGED_IN) {
            edit_account.setVisibility(View.VISIBLE);
        } else {
            edit_account.setVisibility(View.GONE);
        }
        change_password.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                final Dialog fullscreenDialog = new Dialog(getContext());
                fullscreenDialog.setContentView(R.layout.dialog_change_password);

                oldPassword =
fullscreenDialog.findViewById(R.id.current_password);
                newPassword = fullscreenDialog.findViewById(R.id.new_password);
                Button saveButton =
fullscreenDialog.findViewById(R.id.dialog_button);
```

```
                saveButton.setOnClickListener(new View.OnClickListener() {
                    @Override
                    public void onClick(View v) {

                        if (validatePasswordForm()) {
                            dialogLoader.showProgressDialog();
                            Call<UserData> call =
APIClient.getInstance().updatePassword(oldPassword.getText().toString().trim(),
                                    newPassword.getText().toString().trim(),
                                    getContext().getSharedPreferences("UserInfo",
getContext().MODE_PRIVATE).getString("userID", "")
                            );

                            call.enqueue(new Callback<UserData>() {
                                @Override
                                public void onResponse(Call<UserData> call,
Response<UserData> response) {
                                    if (response.isSuccessful()) {
                                        if
(response.body().getSuccess().equalsIgnoreCase("1") && response.body().getData()
!= null) {
                                            // User's Info has been Updated.
                                            Toast.makeText(getContext(),
response.body().getMessage(), Toast.LENGTH_SHORT).show();
                                        } else if
(response.body().getSuccess().equalsIgnoreCase("0")) {
                                            // Unable to Update User's Info.
                                            Toast.makeText(getContext(),
response.body().getMessage(), Toast.LENGTH_SHORT).show();
                                        } else if
(response.body().getSuccess().equalsIgnoreCase("2")) {
                                            // Unable to Update User's Info.
                                            Toast.makeText(getContext(),
response.body().getMessage(), Toast.LENGTH_SHORT).show();
                                        } else {
                                            // Unable to get Success status
                                            Toast.makeText(getContext(),
getString(R.string.unexpected_response), Toast.LENGTH_SHORT).show();
                                        }
                                    } else {
                                        Toast.makeText(getContext(), "" +
response.message(), Toast.LENGTH_SHORT).show();
                                    }
                                    dialogLoader.hideProgressDialog();
                                }

                                @Override
                                public void onFailure(Call<UserData> call,
Throwable t) {
                                    Toast.makeText(getContext(),
"NetworkCallFailure : " + t, Toast.LENGTH_LONG).show();
                                    dialogLoader.hideProgressDialog();
                                }
                            });

                        } else {

                            Toast.makeText(getContext(), "Invalid Password",
Toast.LENGTH_SHORT).show();

                        }

                    }
                });

                fullscreenDialog.show();

            }
        });

        select_language.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                // Navigate to Languages Fragment
                Fragment fragment = new Languages();
                FragmentManager fragmentManager = getFragmentManager();
                fragmentManager.beginTransaction().add(R.id.main_fragment,
fragment)
                        .addToBackStack(null).commit();
            }
        });


        official_web.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String web_url = ((App)
getActivity().getApplicationContext()).getAppSettingsDetails().getSiteUrl();
                if (!web_url.startsWith("https://") &&
!web_url.startsWith("http://"))
                    web_url = "https://www.linkedin.com/in/aman-ahuja-253506168/";

                startActivity(new Intent(Intent.ACTION_VIEW,
Uri.parse("https://www.linkedin.com/in/aman-ahuja-253506168/")));
            }
        });

        share_app.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Utilities.shareMyApp(getContext());
            }
        });

        rate_app.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Utilities.rateMyApp(getContext());
            }
        });

        privacy_policy.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                AlertDialog.Builder dialog = new AlertDialog.Builder(getContext(),
android.R.style.Theme_NoTitleBar);
```

```
                View dialogView =
getActivity().getLayoutInflater().inflate(R.layout.dialog_webview_fullscreen,
null);
                dialog.setView(dialogView);
                dialog.setCancelable(true);

                final ImageButton dialog_button = (ImageButton)
dialogView.findViewById(R.id.dialog_button);
                final TextView dialog_title = (TextView)
dialogView.findViewById(R.id.dialog_title);
                final WebView dialog_webView = (WebView)
dialogView.findViewById(R.id.dialog_webView);

                dialog_title.setText(getString(R.string.privacy_policy));


                String description = ConstantValues.PRIVACY_POLICY;
                String styleSheet = "<style> " +
                        "body{background:#ffffff; margin:10; padding:10} " +
                        "p{color:#757575;} " +
                        "img{display:inline; height:auto; max-width:100%;}" +
                        "</style>";

                dialog_webView.setHorizontalScrollBarEnabled(false);
                dialog_webView.loadDataWithBaseURL(null, styleSheet + description,
"text/html", "utf-8", null);


                final AlertDialog alertDialog = dialog.create();

                if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.LOLLIPOP) {

alertDialog.getWindow().addFlags(WindowManager.LayoutParams.FLAG_DRAWS_SYSTEM_BAR_
BACKGROUNDS);

alertDialog.getWindow().setStatusBarColor(ContextCompat.getColor(getContext(),
R.color.colorPrimaryDark));
                }

                dialog_button.setOnClickListener(new View.OnClickListener() {
                    @Override
                    public void onClick(View v) {
                        alertDialog.dismiss();
                    }
                });

                alertDialog.show();

            }
        });

        refund_policy.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                AlertDialog.Builder dialog = new AlertDialog.Builder(getContext(),
android.R.style.Theme_NoTitleBar);
                View dialogView =
getActivity().getLayoutInflater().inflate(R.layout.dialog_webview_fullscreen,
null);
                dialog.setView(dialogView);
                dialog.setCancelable(true);

                final ImageButton dialog_button = (ImageButton)
dialogView.findViewById(R.id.dialog_button);
                final TextView dialog_title = (TextView)
dialogView.findViewById(R.id.dialog_title);
                final WebView dialog_webView = (WebView)
dialogView.findViewById(R.id.dialog_webView);

                dialog_title.setText(getString(R.string.refund_policy));


                String description = ConstantValues.REFUND_POLICY;
                String styleSheet = "<style> " +
                        "body{background:#ffffff; margin:10; padding:10} " +
                        "p{color:#757575;} " +
                        "img{display:inline; height:auto; max-width:100%;}" +
                        "</style>";

                dialog_webView.setHorizontalScrollBarEnabled(false);
                dialog_webView.loadDataWithBaseURL(null, styleSheet + description,
"text/html", "utf-8", null);


                final AlertDialog alertDialog = dialog.create();

                if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.LOLLIPOP) {

alertDialog.getWindow().addFlags(WindowManager.LayoutParams.FLAG_DRAWS_SYSTEM_BAR_
BACKGROUNDS);

alertDialog.getWindow().setStatusBarColor(ContextCompat.getColor(getContext(),
R.color.colorPrimaryDark));
                }

                dialog_button.setOnClickListener(new View.OnClickListener() {
                    @Override
                    public void onClick(View v) {
                        alertDialog.dismiss();
                    }
                });

                alertDialog.show();
            }
        });


        a_z_terms.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                AlertDialog.Builder dialog = new AlertDialog.Builder(getContext(),
android.R.style.Theme_NoTitleBar);
                View dialogView =
getActivity().getLayoutInflater().inflate(R.layout.dialog_webview_fullscreen,
null);
                dialog.setView(dialogView);
                dialog.setCancelable(true);

                final ImageButton dialog_button = (ImageButton) 
dialogView.findViewById(R.id.dialog_button);
                final TextView dialog_title = (TextView) 
dialogView.findViewById(R.id.dialog_title);
                final WebView dialog_webView = (WebView) 
dialogView.findViewById(R.id.dialog_webView);

                dialog_title.setText(getString(R.string.a_z));


                String description = ConstantValues.A_Z;
                String styleSheet = "<style> " +
                        "body{background:#eeeeee; margin:10; padding:10} " +
                        "p{color:#757575;} " +
                        "img{display:inline; height:auto; max-width:100%;}" +
                        "</style>";

                dialog_webView.setVerticalScrollBarEnabled(true);
                dialog_webView.setHorizontalScrollBarEnabled(false);
                dialog_webView.setBackgroundColor(Color.TRANSPARENT);

dialog_webView.getSettings().setRenderPriority(WebSettings.RenderPriority.HIGH);
                dialog_webView.loadDataWithBaseURL(null, styleSheet + description, 
"text/html", "utf-8", null);


                final AlertDialog alertDialog = dialog.create();

                if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.LOLLIPOP) {

alertDialog.getWindow().addFlags(WindowManager.LayoutParams.FLAG_DRAWS_SYSTEM_BAR_
BACKGROUNDS);

alertDialog.getWindow().setStatusBarColor(ContextCompat.getColor(getContext(), 
R.color.colorPrimaryDark));
                }

                dialog_button.setOnClickListener(new View.OnClickListener() {
                    @Override
                    public void onClick(View v) {
                        alertDialog.dismiss();
                    }
                });

                alertDialog.show();
            }
        });

        service_terms.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                AlertDialog.Builder dialog = new AlertDialog.Builder(getContext(), 
android.R.style.Theme_NoTitleBar);
                View dialogView = 
getActivity().getLayoutInflater().inflate(R.layout.dialog_webview_fullscreen, 
null);
```

```
                dialog.setView(dialogView);
                dialog.setCancelable(true);

                final ImageButton dialog_button = (ImageButton)
dialogView.findViewById(R.id.dialog_button);
                final TextView dialog_title = (TextView)
dialogView.findViewById(R.id.dialog_title);
                final WebView dialog_webView = (WebView)
dialogView.findViewById(R.id.dialog_webView);

                dialog_title.setText(getString(R.string.service_terms));


                String description = ConstantValues.TERMS_SERVICES;
                String styleSheet = "<style> " +
                        "body{background:#ffffff; margin:10; padding:10} " +
                        "p{color:#757575;} " +
                        "img{display:inline; height:auto; max-width:100%;}" +
                        "</style>";

                dialog_webView.setHorizontalScrollBarEnabled(false);
                dialog_webView.loadDataWithBaseURL(null, styleSheet + description,
"text/html", "utf-8", null);


                final AlertDialog alertDialog = dialog.create();

                if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.LOLLIPOP) {

alertDialog.getWindow().addFlags(WindowManager.LayoutParams.FLAG_DRAWS_SYSTEM_BAR_
BACKGROUNDS);

alertDialog.getWindow().setStatusBarColor(ContextCompat.getColor(getContext(),
R.color.colorPrimaryDark));
                }

                dialog_button.setOnClickListener(new View.OnClickListener() {
                    @Override
                    public void onClick(View v) {
                        alertDialog.dismiss();
                    }
                });

                alertDialog.show();
            }
        });


        btn_logout.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                if (ConstantValues.IS_USER_LOGGED_IN) {
                    // Edit UserID in SharedPreferences
                    SharedPreferences.Editor editor = sharedPreferences.edit();
                    editor.putString("userID", "");
                    editor.apply();
                    // Set UserLoggedIn in MyAppPrefsManager
                    MyAppPrefsManager myAppPrefsManager = new
MyAppPrefsManager(getContext());
                    myAppPrefsManager.setUserLoggedIn(false);

                    // Set isLogged_in of ConstantValues
                    ConstantValues.IS_USER_LOGGED_IN =
myAppPrefsManager.isUserLoggedIn();

                    setupAppBarHeader();
                    ((MainActivity) getContext()).setupExpandableDrawerList();
                    ((MainActivity) getContext()).setupExpandableDrawerHeader();


                    btn_logout.setText(getString(R.string.login));

                } else {
                    // Navigate to Login Activity
                    startActivity(new Intent(getContext(), Login.class));
                    ((MainActivity) getContext()).finish();
                    ((MainActivity)
getContext()).overridePendingTransition(R.anim.enter_from_left,
R.anim.exit_out_left);
                }
            }
        });


        return rootView;
    }


    //*********** Setup Header of Navigation Drawer ********//

    public void setupAppBarHeader() {

        // Check if the User is Authenticated
        if (ConstantValues.IS_USER_LOGGED_IN) {
            // Check User's Info from SharedPreferences
            if (!"".equalsIgnoreCase(sharedPreferences.getString("userID", ""))) {

                // Get User's Info from Local Database User_Info_DB
                User_Info_DB userInfoDB = new User_Info_DB();
                UserDetails userInfo =
userInfoDB.getUserData(sharedPreferences.getString("userID", null));

                // Set User's Name, Email and Photo
                profile_email.setText(userInfo.getEmail());
                profile_name.setText(userInfo.getFirstName() + " " +
userInfo.getLastName());

                RequestOptions options = new RequestOptions()
                        .centerCrop()
                        .placeholder(R.drawable.profile)
                        .error(R.drawable.profile)
                        .diskCacheStrategy(DiskCacheStrategy.ALL)
                        .priority(Priority.HIGH);
                Glide.with(getContext())
```

```
                        .setDefaultRequestOptions(options)
                        .asBitmap()
                        .load(ConstantValues.ECOMMERCE_URL + userInfo.getAvatar())
                        .into(profile_image);

                btn_edit_profile.setText(getString(R.string.edit_profile));

btn_edit_profile.setBackground(getResources().getDrawable(R.drawable.rounded_corne
rs_button_green));

                btn_edit_profile.setVisibility(View.GONE);

            } else {
                // Set Default Name, Email and Photo
                profile_image.setImageResource(R.drawable.profile);
                profile_name.setText(getString(R.string.login_or_signup));

profile_email.setText(getString(R.string.login_or_create_account));
                btn_edit_profile.setText(getString(R.string.login));

btn_edit_profile.setBackground(getResources().getDrawable(R.drawable.rounded_corne
rs_button_red));
            }

        } else {
            // Set Default Name, Email and Photo
            profile_image.setImageResource(R.drawable.profile);
            profile_name.setText(getString(R.string.login_or_signup));
            profile_email.setText(getString(R.string.login_or_create_account));
            btn_edit_profile.setText(getString(R.string.login));

btn_edit_profile.setBackground(getResources().getDrawable(R.drawable.rounded_corne
rs_button_red));
            change_password.setVisibility(View.GONE);
        }


        // Handle DrawerHeader Click Listener
        btn_edit_profile.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

                // Check if the User is Authenticated
                if (ConstantValues.IS_USER_LOGGED_IN) {

                    // Navigate to Update_Account Fragment
                    Fragment fragment = new Update_Account(true);
                    FragmentManager fragmentManager = getFragmentManager();
                    fragmentManager.beginTransaction()
                            .add(R.id.main_fragment, fragment)

.setTransition(FragmentTransaction.TRANSIT_FRAGMENT_FADE)

.addToBackStack(getString(R.string.actionCart)).commit();

                } else {
                    // Navigate to Login Activity
                    startActivity(new Intent(getContext(), Login.class));
                    ((MainActivity) getContext()).finish();
                    ((MainActivity)
getContext()).overridePendingTransition(R.anim.enter_from_left,
R.anim.exit_out_left);
                }
            }
        });
    }


    /*********** Request Server to Enable or Disable Notifications ********/

    public void TogglePushNotification(Boolean enable) {

        dialogLoader.showProgressDialog();

        String notify = "1";
        if (!enable) {
            notify = "0";
        }

        String deviceID = "";

        if (ConstantValues.DEFAULT_NOTIFICATION.equalsIgnoreCase("onesignal")) {
            deviceID =
OneSignal.getPermissionSubscriptionState().getSubscriptionStatus().getUserId();
        } else if (ConstantValues.DEFAULT_NOTIFICATION.equalsIgnoreCase("fcm")) {
            deviceID = FirebaseInstanceId.getInstance().getToken();
        }


        Call<ContactUsData> call = APIClient.getInstance()
                .notify_me
                        (
                                notify,
                                deviceID
                        );

        call.enqueue(new Callback<ContactUsData>() {
            @Override
            public void onResponse(Call<ContactUsData> call,
retrofit2.Response<ContactUsData> response) {

                dialogLoader.hideProgressDialog();

                if (response.isSuccessful()) {
                    if (response.body().getSuccess().equalsIgnoreCase("1")) {

                        Snackbar.make(rootView, response.body().getMessage(),
Snackbar.LENGTH_SHORT).show();

                    } else if (response.body().getSuccess().equalsIgnoreCase("0"))
{
                        Snackbar.make(rootView, response.body().getMessage(),
Snackbar.LENGTH_LONG).show();

                    } else {
                        // Unable to get Success status
                        Snackbar.make(rootView,
getString(R.string.unexpected_response), Snackbar.LENGTH_SHORT).show();
                    }
                } else {
                    Toast.makeText(getContext(), response.message(),
Toast.LENGTH_SHORT).show();
                }
            }

            @Override
            public void onFailure(Call<ContactUsData> call, Throwable t) {
                dialogLoader.hideProgressDialog();
                Toast.makeText(getContext(), "NetworkCallFailure : " + t,
Toast.LENGTH_LONG).show();
            }
        });
    }

    private boolean validatePasswordForm() {
        if
(!ValidateInputs.isValidPassword(newPassword.getText().toString().trim())) {
            newPassword.setError(getString(R.string.invalid_password));
            return false;
        } else {
            return true;
        }
    }
}

apimodules:
```

API interceptor.java

```
public class API_Interceptor implements Interceptor {

    private static final String ECOMMERCE_CONSUMER_KEY = "consumer-key";
    private static final String ECOMMERCE_CONSUMER_SECRET = "consumer-secret";
    private static final String ECOMMERCE_CONSUMER_NONCE = "consumer-nonce";
    private static final String ECOMMERCE_CONSUMER_DEVICE_ID = "consumer-device-
id";
    private static final String ECOMMERCE_COMSUMER_IP = "consumer-ip";

    private String consumerKey;
    private String consumerSecret;
    private String consumerNonce;
    private String consumerDeviceID;
    private String consumerIP;



    private API_Interceptor(String consumerKey, String consumerSecret,String
consumerIP) {
        this.consumerKey = consumerKey;
        this.consumerSecret = consumerSecret;
        this.consumerIP = consumerIP;
    }


    @Override
    public Response intercept(Chain chain) throws IOException {
        Request original = chain.request();
        HttpUrl originalHttpUrl = original.url();

        consumerNonce = getRandomNonce(32);

        String deviceID = "";

        if (ConstantValues.DEFAULT_NOTIFICATION.equalsIgnoreCase("onesignal")) {
            deviceID = OneSignal.getPermissionSubscriptionState
().getSubscriptionStatus().getUserId();
        }
        else if (ConstantValues.DEFAULT_NOTIFICATION.equalsIgnoreCase("fcm")) {
            deviceID = FirebaseInstanceId.getInstance().getToken();
        }


        if (deviceID != null  &&  !TextUtils.isEmpty(deviceID)) {
            consumerDeviceID = deviceID;
        }
        else {
            consumerDeviceID = getRandomNonce(32);
        }


        Log.d("VC_Shop", ECOMMERCE_CONSUMER_KEY+" = "+consumerKey);
        Log.d("VC_Shop", ECOMMERCE_CONSUMER_SECRET+" = "+consumerSecret);
        Log.d("VC_Shop", ECOMMERCE_CONSUMER_NONCE+" = "+consumerNonce);
        Log.d("VC_Shop", ECOMMERCE_CONSUMER_DEVICE_ID+" = "+consumerDeviceID);


        HttpUrl url = originalHttpUrl.newBuilder().build();

        // Request customization: add request headers
        Request.Builder requestBuilder = original.newBuilder()
                .addHeader("Content-Type", "application/json")
                .addHeader(ECOMMERCE_CONSUMER_KEY, consumerKey)
                .addHeader(ECOMMERCE_CONSUMER_SECRET, consumerSecret)
                .addHeader(ECOMMERCE_COMSUMER_IP,consumerIP)
                .addHeader(ECOMMERCE_CONSUMER_NONCE, consumerNonce)
                .addHeader(ECOMMERCE_CONSUMER_DEVICE_ID, consumerDeviceID)

                .url(url);


        Request request = requestBuilder.build();
        return chain.proceed(request);
    }


    private static String getRandomNonce(final int sizeOfRandomString) {

        final String ALLOWED_CHARACTERS ="0123456789qwertyuiopasdfghjklzxcvbnm-";

        Random generator = new Random();
        StringBuilder randomStringBuilder = new StringBuilder(sizeOfRandomString);

        for(int i=0;i<sizeOfRandomString;++i)
```

```
randomStringBuilder.append(ALLOWED_CHARACTERS.charAt(generator.nextInt(ALLOWED_CHA
RACTERS.length())));

        return randomStringBuilder.toString();
    }




    public static final class Builder {

        private String consumerKey;
        private String consumerSecret;
        private String consumerIP;

        public Builder consumerKey(String consumerKey) {
            if (consumerKey == null) throw new NullPointerException("consumerKey =
null");
            this.consumerKey = consumerKey;
            return this;
        }

        public Builder consumerSecret(String consumerSecret) {
            if (consumerSecret == null) throw new
NullPointerException("consumerSecret = null");
            this.consumerSecret = consumerSecret;
            return this;
        }

        public Builder consumerIP(String consumerIP){
            if (consumerIP == null) throw new NullPointerException("consumer-ip =
null");
            this.consumerIP = consumerIP;
            return this;
        }


        public API_Interceptor build() {

            if (consumerKey == null) throw new IllegalStateException("consumerKey
not set");
            if (consumerSecret == null) throw new
IllegalStateException("consumerSecret not set");
            if (consumerIP == null) throw new IllegalStateException("consumerIP
not set");

            return new API_Interceptor(consumerKey, consumerSecret,consumerIP);
        }
    }

    public String urlEncoded(String url) {
        String encodedURL = "";
        try {
            encodedURL = URLEncoder.encode(url, "UTF-8");
            Log.d("encodedURL", encodedURL);
        } catch (UnsupportedEncodingException e) {
            e.printStackTrace();
        }

        return encodedURL;
    }
}
```

API Clients.java

```
    // Base URL for API Requests
    private static final String BASE_URL = ConstantValues.ECOMMERCE_URL + "api/";

    private static APIRequests apiRequests;


    // Singleton Instance of APIRequests
    public static APIRequests getInstance() {
        if (apiRequests == null) {

            HttpLoggingInterceptor httpLoggingInterceptor = new
HttpLoggingInterceptor();
            httpLoggingInterceptor.setLevel(HttpLoggingInterceptor.Level.BODY);

            API_Interceptor apiInterceptor = new API_Interceptor.Builder()

.consumerKey(Utilities.getMd5Hash(ConstantValues.ECOMMERCE_CONSUMER_KEY))

.consumerSecret(Utilities.getMd5Hash(ConstantValues.ECOMMERCE_CONSUMER_SECRET))
                    .consumerIP(ConstantValues.getLocalIpAddress())
                    .build();

            BasicOAuth basicOAuthWoocommerce = new BasicOAuth.Builder()
                    .consumerKey(ConstantValues.ECOMMERCE_CONSUMER_KEY)
                    .consumerSecret(ConstantValues.ECOMMERCE_CONSUMER_SECRET)
                    .consumerIP(ConstantValues.getLocalIpAddress())
                    .build();

            OkHttpClient okHttpClient = new OkHttpClient().newBuilder()
                    .connectTimeout(5, TimeUnit.MINUTES)
                    .readTimeout(60, TimeUnit.SECONDS)
                    .writeTimeout(60, TimeUnit.SECONDS)
                    //.addInterceptor(apiInterceptor)
                    .addInterceptor(httpLoggingInterceptor)
                    .addNetworkInterceptor(new Interceptor() {
                        @Override
                        public Response intercept(Chain chain) throws IOException
{
                            Request request = chain.request().newBuilder()
                                    // .addHeader(Constant.Header, authToken)
                                    .build();
                            return chain.proceed(request);
                        }
                    })
                    .addInterceptor(apiInterceptor)
                    //.addInterceptor(BASE_URL.startsWith("http://")?
apiInterceptor : basicOAuthWoocommerce)
```

```
                    .build();


            Retrofit retrofit = new Retrofit.Builder()
                    .baseUrl(BASE_URL)
                    .client(okHttpClient)
                    .addConverterFactory(GsonConverterFactory.create())
                    .build();


            apiRequests = retrofit.create(APIRequests.class);

            return apiRequests;
        }
        else {
            return apiRequests;
        }
    }

}
```

Api requests.interface

```
public interface APIRequests {

    //******************** User Data ********************//

    @Multipart
    @POST("processregistration")
    Call<UserData> processRegistration(
            @Part("customers_firstname") RequestBody customers_firstname,
            @Part("customers_lastname") RequestBody customers_lastname,
            @Part("email") RequestBody email,
            @Part("password") RequestBody password,
            @Part("country_code") RequestBody country_code,
            @Part("customers_telephone") RequestBody customers_picture);
/*
    @Multipart
    @POST("processregistration")
    Call<UserData> processRegistration(
            @Part("customers_firstname") RequestBody customers_firstname,
            @Part("customers_lastname") RequestBody customers_lastname,
            @Part("email") RequestBody email,
            @Part("password") RequestBody password,
            @Part("country_code") RequestBody country_code,
            @Part("customers_gender") RequestBody customers_gender,
            @Part("customers_telephone") RequestBody customers_picture,
            @Part("customers_dob") RequestBody customers_dob);
*/

    @FormUrlEncoded
    @POST("processlogin")
    Call<UserData> processLogin(@Field("email") String customers_email_address,
                                @Field("password") String customers_password);

    @FormUrlEncoded
    @POST("facebookregistration")
    Call<UserData> facebookRegistration(@Field("access_token") String
access_token);

    @FormUrlEncoded
    @POST("googleregistration")
    Call<UserData> googleRegistration(@Field("idToken") String idToken,
                                      @Field("customers_id") String userId,
                                      @Field("givenName") String givenName,
                                      @Field("familyName") String familyName,
                                      @Field("email") String email,
                                      @Field("imageUrl") String imageUrl);

    @FormUrlEncoded
    @POST("processforgotpassword")
    Call<UserData> processForgotPassword(@Field("email") String
customers_email_address);

    @FormUrlEncoded
    @POST("updatecustomerinfo")
    Call<UserData> updateCustomerInfo(@Field("customers_id") String customers_id,
                                      @Field("customers_firstname") String
customers_firstname,
                                      @Field("customers_lastname") String
customers_lastname,
                                      @Field("customers_gender") String
customers_gender,
                                      @Field("customers_telephone") String
customers_telephone,
                                      /*@Field("customers_dob") String
customers_dob,*/
                                      @Field("image_id") String image_id);


    //******************** Address Data ********************//

    @POST("getcountries")
    Call<Countries> getCountries();

    @FormUrlEncoded
    @POST("getzones")
    Call<Zones> getZones(@Field("zone_country_id") String zone_country_id);

    @FormUrlEncoded
    @POST("getalladdress")
    Call<AddressData> getAllAddress(@Field("customers_id") String customers_id);

    @FormUrlEncoded
    @POST("addshippingaddress")
    Call<AddressData> addUserAddress(@Field("customers_id") String customers_id,
                                     @Field("entry_firstname") String
entry_firstname,
                                     @Field("entry_lastname") String
entry_lastname,
                                     @Field("entry_street_address") String
entry_street_address,
                                     @Field("entry_postcode") String
entry_postcode,
                                     @Field("entry_city") String entry_city,
                                     @Field("entry_country_id") String
entry_country_id,
                                     @Field("entry_zone_id") String entry_zone_id,
                                     @Field("is_default") String
customers_default_address_id);

    @FormUrlEncoded
    @POST("updateshippingaddress")
    Call<AddressData> updateUserAddress(@Field("customers_id") String
customers_id,
                                        @Field("address_id") String address_id,
                                        @Field("entry_firstname") String
entry_firstname,
                                        @Field("entry_lastname") String
entry_lastname,
                                        @Field("entry_street_address") String
entry_street_address,
                                        @Field("entry_postcode") String
entry_postcode,
                                        @Field("entry_city") String entry_city,
                                        @Field("entry_country_id") String
entry_country_id,
                                        @Field("entry_zone_id") String
entry_zone_id,
                                        @Field("is_default") String
customers_default_address_id);

    @FormUrlEncoded
    @POST("updatedefaultaddress")
    Call<AddressData> updateDefaultAddress(@Field("customers_id") String
customers_id,
                                           @Field("address_book_id") String
address_book_id);

    @FormUrlEncoded
    @POST("deleteshippingaddress")
    Call<AddressData> deleteUserAddress(@Field("customers_id") String
customers_id,
                                        @Field("address_book_id") String
address_book_id);


    //******************** OrderProductCategory Data ********************//

    @FormUrlEncoded
    @POST("allcategories")
    Call<CategoryData> getAllCategories(@Field("language_id") int language_id);

    //******************** Product Data ********************//

    @POST("getallproducts")
    Call<ProductData> getAllProducts(@Body GetAllProducts getAllProducts);

    @POST("getquantity")
    Call<ProductStock> getProductStock(@Body GetStock getStock);

    @FormUrlEncoded
    @POST("likeproduct")
    Call<ProductData> likeProduct(@Field("liked_products_id") int
liked_products_id,
                                  @Field("liked_customers_id") String
liked_customers_id);

    @FormUrlEncoded
    @POST("unlikeproduct")
    Call<ProductData> unlikeProduct(@Field("liked_products_id") int
liked_products_id,
                                    @Field("liked_customers_id") String
liked_customers_id);

    @FormUrlEncoded
    @POST("getfilters")
    Call<FilterData> getFilters(@Field("categories_id") String categories_id,
                                @Field("language_id") int language_id);

    @FormUrlEncoded
    @POST("getsearchdata")
    Call<SearchData> getSearchData(@Field("searchValue") String searchValue,
                                   @Field("language_id") int language_id,
                                   @Field("currency_code") String currency_code);

    //******************** News Data ********************//

    @FormUrlEncoded
    @POST("getallnews")
    Call<NewsData> getAllNews(@Field("language_id") int language_id,
                              @Field("page_number") int page_number,
                              @Field("is_feature") int is_feature,
                              @Field("categories_id") String categories_id);

    @FormUrlEncoded
    @POST("allnewscategories")
    Call<NewsCategoryData> allNewsCategories(@Field("language_id") int
language_id,
                                             @Field("page_number") int
page_number);

    //******************** Order Data ********************//

    @POST("addtoorder")
    Call<OrderData> addToOrder(@Body PostOrder postOrder);

    @FormUrlEncoded
    @POST("getorders")
    Call<OrderData> getOrders(@Field("customers_id") String customers_id,
                              @Field("language_id") int language_id,
                              @Field("currency_code") String currency_code);

    @FormUrlEncoded
    @POST("updatestatus")
    Call<OrderData> updatestatus(@Field("customers_id") String customers_id,
                                 @Field("orders_id") int orders_id);

    @FormUrlEncoded
    @POST("getcoupon")
    Call<CouponsData> getCouponInfo(@Field("code") String code);
    @FormUrlEncoded
    @POST("getpaymentmethods")
    Call<PaymentMethodsData> getPaymentMethods(@Field("language_id") int
language_id);

    @GET("generatebraintreetoken")
    Call<GetBrainTreeToken> generateBraintreeToken();

    //******************** Banner Data ********************//

    @GET("getbanners")
    Call<BannerData> getBanners(@Query("languages_id") int languages_id);

    //******************** Tax & Shipping Data ********************//

    @POST("getrate")
    Call<ShippingRateData> getShippingMethodsAndTax(@Body PostTaxAndShippingData
postTaxAndShippingData);

    //******************** Contact Us Data ********************//

    @FormUrlEncoded
    @POST("contactus")
    Call<ContactUsData> contactUs(@Field("name") String name,
                                  @Field("email") String email,
                                  @Field("message") String message);

    //******************** Languages Data ********************//

    @GET("getlanguages")
    Call<LanguageData> getLanguages();

    //******************** App Settings Data ********************//

    @GET("sitesetting")
    Call<AppSettingsData> getAppSetting();


    //******************** Static Pages Data ********************//

    @FormUrlEncoded
    @POST("getallpages")
    Call<PagesData> getStaticPages(@Field("language_id") int language_id);

    //******************** Notifications Data ********************//

    @FormUrlEncoded
    @POST("registerdevices")
    Call<UserData> registerDeviceToFCM(@Field("device_id") String device_id,
                                       @Field("device_type") String device_type,
                                       @Field("ram") String ram,
                                       @Field("processor") String processor,
                                       @Field("device_os") String device_os,
                                       @Field("location") String location,
                                       @Field("device_model") String device_model,
                                       @Field("manufacturer") String manufacturer,
                                       @Field("customers_id") String
customers_id);

    @FormUrlEncoded
    @POST("notify_me")
    Call<ContactUsData> notify_me(@Field("is_notify") String is_notify,
                                  @Field("device_id") String device_id);

    @FormUrlEncoded
    @POST("givereview")
    Call<GiveRating> giveRating(@FieldMap Map<String, String> stringMap);

    @GET("getreviews")
    Call<GetRatings> getProductReviews(@Query("products_id") String product_id,
                                       @Query("languages_id") String
languages_id);

    // This Api will give us City bounds
    @GET("getlocation")
    Call<GoogleAPIResponse> getCityBounds(@Query(value = "address", encoded =
true) String address);

    // Upload Image
    @Multipart
    @POST("uploadimage")
    Call<UploadImageModel> uploadImage(@Part MultipartBody.Part filePart);

    // Update Password
    @GET("updatepassword")
    Call<UserData> updatePassword(@Query("oldpassword") String oldPassword,
                                  @Query("newpassword") String newPassword,
                                  @Query("customers_id") String customers_id);

    //Change Currency
    @GET("getcurrencies")
    Call<CurrencyModel> getCurrency();

    @GET("generatpaytmhashes")
    Call<Checksum> getPayTMChecksum(
            @Query("customers_id") String customerID,
            @Query("amount") String totalAmount
    );

}
```

Basic Auth to api.java

```
public class BasicOAuth implements Interceptor {

    private static final String OAUTH_CONSUMER_KEY = "consumer_key";
    private static final String OAUTH_CONSUMER_SECRET = "consumer_secret";
    private static final String OAUTH_COMSUMER_IP = "consumer-ip";

    private final String consumerKey;
    private final String consumerSecret;
    private final String consumerIP;


    private BasicOAuth(String consumerKey, String consumerSecret,String
consumerIP) {
        this.consumerKey = consumerKey;
        this.consumerSecret = consumerSecret;
        this.consumerIP = consumerIP;
    }

    @Override
    public Response intercept(Chain chain) throws IOException {
        Request original = chain.request();
        HttpUrl originalHttpUrl = original.url();

        Log.i("request_method", ""+original.method());
        Log.i("request_URL", original.url().toString());
        Log.i("request_host", ""+original.url().host());
        Log.i("request_encodedPath", original.url().encodedPath());
        Log.i("request_query", ""+original.url().query());
        Log.i("request_encodedQuery", ""+original.url().encodedQuery());

        ////////////////////////////////////////////////////////////


        String basicOAuthString = OAUTH_CONSUMER_KEY + "=" + consumerKey + "&" +
OAUTH_CONSUMER_SECRET + "=" + consumerSecret
                                    +"&" + OAUTH_COMSUMER_IP + "=" + consumerIP;
        Log.i("basicOAuthString", "basicOAuthString="+basicOAuthString);

        HttpUrl url = originalHttpUrl.newBuilder()
                .addQueryParameter(OAUTH_CONSUMER_KEY, consumerKey)
                .addQueryParameter(OAUTH_CONSUMER_SECRET, consumerSecret)
                .addQueryParameter(OAUTH_COMSUMER_IP,consumerIP)
                .build();


        // Request customization: add request headers
        Request.Builder requestBuilder = original.newBuilder()
                .addHeader("Content-Type", "application/json")
                .url(url);

        Request request = requestBuilder.build();
        return chain.proceed(request);
    }


    public static final class Builder {

        private String consumerKey;
        private String consumerSecret;
        private String consumerIP;
        private int type;

        public Builder consumerKey(String consumerKey) {
            if (consumerKey == null) throw new NullPointerException("consumerKey =
null");
            this.consumerKey = consumerKey;
            return this;
        }

        public Builder consumerSecret(String consumerSecret) {
            if (consumerSecret == null) throw new
NullPointerException("consumerSecret = null");
            this.consumerSecret = consumerSecret;
            return this;
        }

        public Builder consumerIP(String consumerIP) {
            if (consumerIP == null) throw new NullPointerException("consumerIP =
null");
            this.consumerIP = consumerIP;
            return this;
        }



        public BasicOAuth build() {

            if (consumerKey == null) throw new IllegalStateException("consumerKey
not set");
            if (consumerSecret == null) throw new
IllegalStateException("consumerSecret not set");
            if (consumerSecret == null) throw new
IllegalStateException("consumerIP not set");

            return new BasicOAuth(consumerKey, consumerSecret,consumerIP);
        }
    }

    public String urlEncoded(String url) {
        String encodedurl = "";
        try {

            encodedurl = URLEncoder.encode(url, "UTF-8");
            Log.d("encodedURL", encodedurl);
        } catch (UnsupportedEncodingException e) {
            e.printStackTrace();
        }

        return encodedurl;
    }
}
```

Startapprequests.java

```
public class StartAppRequests {

    private App app = new App();


    public StartAppRequests(Context context) {
        app = ((App) context.getApplicationContext());
    }



    //*********** Contains all methods to Execute on Startup ********//

    public void StartRequests(){
        //RequestBanners();
        //RequestAllCategories();
        RequestAppSetting();
        RequestStaticPagesData();

    }



    //*********** Register Device to Admin Panel with the Device's Info ********//

    public static void RegisterDeviceForFCM(final Context context) {

        SharedPreferences sharedPreferences = 
context.getSharedPreferences("UserInfo", MODE_PRIVATE);

        String deviceID = "";
        DeviceInfo device = Utilities.getDeviceInfo(context);
        String customer_ID = sharedPreferences.getString("userID", "");


        if (ConstantValues.DEFAULT_NOTIFICATION.equalsIgnoreCase("onesignal")) {
            deviceID = OneSignal.getPermissionSubscriptionState
().getSubscriptionStatus().getUserId();
        }
        else if (ConstantValues.DEFAULT_NOTIFICATION.equalsIgnoreCase("fcm")) {
            deviceID = FirebaseInstanceId.getInstance().getToken();
        }



        Call<UserData> call = APIClient.getInstance()
                .registerDeviceToFCM
                        (
                                deviceID,
                                device.getDeviceType(),
                                device.getDeviceRAM(),
                                device.getDeviceProcessors(),
                                device.getDeviceAndroidOS(),
                                device.getDeviceLocation(),
                                device.getDeviceModel(),
                                device.getDeviceManufacturer(),
                                customer_ID
                        );

        call.enqueue(new Callback<UserData>() {
            @Override
            public void onResponse(Call<UserData> call, 
retrofit2.Response<UserData> response) {

                if (response.isSuccessful()) {
                    if (response.body().getSuccess().equalsIgnoreCase("1")) {

                        Log.i("notification", response.body().getMessage());
//                        Toast.makeText(context, response.body().getMessage(), 
Toast.LENGTH_LONG).show();
```

```
                    }
                    else {

                        Log.i("notification", response.body().getMessage());
                        Toast.makeText(context, response.body().getMessage(),
Toast.LENGTH_LONG).show();
                    }
                }
                else {
                    Log.i("notification", response.errorBody().toString());
                }
            }

            @Override
            public void onFailure(Call<UserData> call, Throwable t) {
//                Toast.makeText(context, "NetworkCallFailure : "+t,
Toast.LENGTH_LONG).show();
            }
        });

    }



    //*********** API Request Method to Fetch App Banners ********//

    public void RequestBanners() {

        Call<BannerData> call = APIClient.getInstance()
                .getBanners(ConstantValues.LANGUAGE_ID);

        try {
            Response<BannerData> response = call.execute();

            BannerData bannerData = new BannerData();

            if (response.isSuccessful()) {

                bannerData = response.body();

                if (!TextUtils.isEmpty(bannerData.getSuccess()))
                    app.setBannersList(bannerData.getData());

            }

        } catch (IOException e) {
            e.printStackTrace();
        }

    }


    //*********** API Request Method to Fetch All Categories ********//

    public void RequestAllCategories() {

        Call<CategoryData> call = APIClient.getInstance()
                .getAllCategories
```

```
                    (
                            ConstantValues.LANGUAGE_ID
                    );

    try {
        Response<CategoryData> response = call.execute();

        CategoryData categoryData = new CategoryData();

        if (response.isSuccessful()) {

            String json= new Gson().toJson(response.body());
            categoryData = response.body();

            if (!TextUtils.isEmpty(categoryData.getSuccess()))
                app.setCategoriesList(categoryData.getData());

        }

    } catch (IOException e) {
        e.printStackTrace();
    }

}



//*********** Request App Settings from the Server ********//

private void RequestAppSetting() {

    Call<AppSettingsData> call = APIClient.getInstance()
            .getAppSetting();

    try {
        Response<AppSettingsData> response = call.execute();

        if (response.isSuccessful()) {


            AppSettingsData appSettingsData = null;

            appSettingsData = response.body();
            String strJson = new Gson().toJson(appSettingsData);
            if (!TextUtils.isEmpty(appSettingsData.getSuccess()))
                app.setAppSettingsDetails(appSettingsData.getAppDetails());

        }
       else {

           Log.e("Appsettings","Response is not successful");
        }

    } catch (IOException e) {
        e.printStackTrace();
    }
}
```

```
    //*********** Request Static Pages Data from the Server ********//

    private void RequestStaticPagesData() {

        ConstantValues.ABOUT_US = app.getString(R.string.lorem_ipsum);
        ConstantValues.TERMS_SERVICES = app.getString(R.string.lorem_ipsum);
        ConstantValues.PRIVACY_POLICY = app.getString(R.string.lorem_ipsum);
        ConstantValues.REFUND_POLICY = app.getString(R.string.lorem_ipsum);
        ConstantValues.A_Z = app.getString(R.string.lorem_ipsum);


        Call<PagesData> call = APIClient.getInstance()
                .getStaticPages
                        (
                                ConstantValues.LANGUAGE_ID
                        );

        try {
            Response<PagesData> response = call.execute();

            PagesData pages = new PagesData();

            if (response.isSuccessful()) {

                pages = response.body();

                if (pages.getSuccess().equalsIgnoreCase("1")) {

                    app.setStaticPagesDetails(pages.getPagesData());

                    for (int i=0;  i<pages.getPagesData().size();  i++) {
                        PagesDetails page = pages.getPagesData().get(i);

                        if (page.getSlug().equalsIgnoreCase("about-us")) {
                            ConstantValues.ABOUT_US = page.getDescription();
                        }
                        else if (page.getSlug().equalsIgnoreCase("term-services"))
{
                            ConstantValues.TERMS_SERVICES = page.getDescription();
                        }
                        else if (page.getSlug().equalsIgnoreCase("privacy-
policy")) {
                            ConstantValues.PRIVACY_POLICY = page.getDescription();
                        }
                        else if (page.getSlug().equalsIgnoreCase("refund-policy"))
{
                            ConstantValues.REFUND_POLICY = page.getDescription();
                        }
                        else if(page.getSlug().equalsIgnoreCase("A-Z")){
                            ConstantValues.A_Z = page.getDescription();
                        }
                    }
                }
            }

        } catch (IOException e) {
            e.printStackTrace();
        }

    }

}
```

Image module:

Imagepicker.java

```
public class ImagePicker {

    private static final String TAG = "ImagePicker";
    private static final String TEMP_IMG_NAME = "tempImage";
    private static final int DEFAULT_MIN_WIDTH_QUALITY = 400;

    private static int minWidthQuality = DEFAULT_MIN_WIDTH_QUALITY;


    //*********** Returns Intent with Options of Image Picker Apps like Gallery,
Camera etc ********//

    public static Intent getImagePickerIntent(Context context) {
        // Chooser Intent of Filesystem Options
        Intent chooserIntent = null;
        // List of Intents
        List<Intent> pickerIntentsList = new ArrayList<>();


        // Filesystem Intent
        Intent galleryIntent;
        if (Build.VERSION.SDK_INT <= 19) {
            galleryIntent = new Intent();
            galleryIntent.setType("image/*");
            galleryIntent.setAction(Intent.ACTION_GET_CONTENT);
            galleryIntent.addCategory(Intent.CATEGORY_OPENABLE);
        }
        else if (Build.VERSION.SDK_INT > 19) {
            galleryIntent = new Intent(Intent.ACTION_PICK,
MediaStore.Images.Media.EXTERNAL_CONTENT_URI);
        }
        else {
            galleryIntent = new Intent(Intent.ACTION_PICK,
MediaStore.Images.Media.EXTERNAL_CONTENT_URI);
        }


        Uri uriForFile = FileProvider.getUriForFile(context,
BuildConfig.APPLICATION_ID + ".provider", getTempFile(context));

        // Camera Intent
        Intent cameraIntent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
        cameraIntent.putExtra("return_data", true);
        cameraIntent.putExtra(MediaStore.EXTRA_OUTPUT, uriForFile);
        // Adding Filesystem and Camera Intents to IntentList
        pickerIntentsList = addIntentToList(context, pickerIntentsList,
galleryIntent);
        pickerIntentsList = addIntentToList(context, pickerIntentsList,
cameraIntent);


        // Initializing Chooser Intent
        chooserIntent = new
Intent(Intent.createChooser(pickerIntentsList.remove(pickerIntentsList.size() -
1), "Select Source"));

        // Adding IntentList of Camera and Filesystem Intents
        chooserIntent.putExtra(Intent.EXTRA_INITIAL_INTENTS,
pickerIntentsList.toArray(new Parcelable[]{}));


        return chooserIntent;
    }



    //*********** Adds Intents to the IntentList ********//

    private static List<Intent> addIntentToList(Context context, List<Intent>
intentList, Intent intent) {
        List<ResolveInfo> resInfoList =
context.getPackageManager().queryIntentActivities(intent, 0);

        for (ResolveInfo resolveInfo : resInfoList) {
            String packageName = resolveInfo.activityInfo.packageName;
            Intent targetedIntent = new Intent(intent);
            targetedIntent.setPackage(packageName);

            intentList.add(targetedIntent);
        }

        return  intentList;
    }



    //*********** Returns Temp File ********//

    private static File getTempFile(Context context) {
        // Create New Temp File
        File imageFile = new File(context.getExternalCacheDir(), TEMP_IMG_NAME);

        // Create Dir for Temp File
        imageFile.getParentFile().mkdir();

        return imageFile;
    }



    //*********** Returns the User Selected Image as Bitmap ********//
    public static Bitmap getImageFromResult(Context context, int resultCode,
Intent imageReturnedIntent) {
        Bitmap bitmap = null;
        File imageFile = getTempFile(context);

        if (resultCode == Activity.RESULT_OK) {
            Uri selectedImage;
            boolean isCamera = (imageReturnedIntent == null ||
imageReturnedIntent.getData() == null ||
imageReturnedIntent.getData().toString().contains(imageFile.toString()));

            if (isCamera) {
                // From Camera
                selectedImage = FileProvider.getUriForFile(context,
BuildConfig.APPLICATION_ID + ".provider", imageFile);
            } else {
                // From Storage
                selectedImage = imageReturnedIntent.getData();
            }

            bitmap = getResizedImage(context, selectedImage);
        }
        return bitmap;
    }



    //*********** Resize to avoid using too much Memory loading Big Images
(2560*1920) ********//

    private static Bitmap getResizedImage(Context context, Uri selectedImage) {
        Bitmap resizedBitmap = null;
        int[] sampleSizes = new int[]{5, 4, 3, 2, 1};

        int i = 0;
        do {
            resizedBitmap = decodeBitmap(context, selectedImage, sampleSizes[i]);
            i++;

        } while (resizedBitmap.getWidth() < minWidthQuality && i <
sampleSizes.length);

        return resizedBitmap;
    }



    //*********** Returns Bitmap Decoded from Uri ********//

    private static Bitmap decodeBitmap(Context context, Uri uri, int sampleSize) {
        Bitmap decodedBitmap = null;
        AssetFileDescriptor fileDescriptor = null;
        BitmapFactory.Options bitmapOptions = new BitmapFactory.Options();

        try {
            fileDescriptor =
context.getContentResolver().openAssetFileDescriptor(uri, "r");
```

```
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        }
        decodedBitmap =
BitmapFactory.decodeFileDescriptor(fileDescriptor.getFileDescriptor(), null,
bitmapOptions);

        return decodedBitmap;
    }



    //*********** Used for the Rotation of Image ********//

    private static int getRotation(Context context, Uri imageUri, boolean
isCamera) {
        int rotation;

        if (isCamera) {
            rotation = getRotationFromCamera(context, imageUri);
        } else {
            rotation = getRotationFromGallery(context, imageUri);
        }

        return rotation;
    }



    //*********** Used for the Rotation of Image ********//

    private static int getRotationFromCamera(Context context, Uri imageFile) {

        int rotate = 0;

        try {
            context.getContentResolver().notifyChange(imageFile, null);
            ExifInterface exif = new ExifInterface(imageFile.getPath());
            int orientation = exif.getAttributeInt(ExifInterface.TAG_ORIENTATION,
ExifInterface.ORIENTATION_NORMAL);

            switch (orientation) {
                case ExifInterface.ORIENTATION_ROTATE_270:
                    rotate = 270;
                    break;
                case ExifInterface.ORIENTATION_ROTATE_180:
                    rotate = 180;
                    break;
                case ExifInterface.ORIENTATION_ROTATE_90:
                    rotate = 90;
                    break;
            }
        } catch (Exception e) {
            e.printStackTrace();
        }

        return rotate;
    }
    //*********** Used for the Rotation of Image ********//

    public static int getRotationFromGallery(Context context, Uri imageUri) {

        int result = 0;
        Cursor cursor = null;
        String[] columns = {MediaStore.Images.Media.ORIENTATION};

        try {
            cursor = context.getContentResolver().query(imageUri, columns, null,
null, null);
            if (cursor != null && cursor.moveToFirst()) {
                int orientationColumnIndex = cursor.getColumnIndex(columns[0]);
                result = cursor.getInt(orientationColumnIndex);
            }

        } catch (Exception e) {
            // Handle Exception

        } finally {
            if (cursor != null) {
                cursor.close();
            }
        }

        return result;
    }



    //*********** Rotates the Bitmap ********//

    private static Bitmap rotate(Bitmap bm, int rotation) {

        if (rotation != 0) {
            Matrix matrix = new Matrix();
            matrix.postRotate(rotation);
            Bitmap bmOut = Bitmap.createBitmap(bm, 0, 0, bm.getWidth(),
bm.getHeight(), matrix, true);

            return bmOut;
        }
        return bm;
    }

}
```

Notification helper.java

```
public class NotificationHelper {


    public static final int NOTIFICATION_REQUEST_CODE = 100;
    //*********** Used to create Notifications ********//

    public static void showNewNotification(Context context, Intent intent, String
title, String msg, Bitmap bitmap) {

        Uri notificationSound =
RingtoneManager.getDefaultUri(RingtoneManager.TYPE_NOTIFICATION);

        Intent notificationIntent;

        if (intent != null) {
            notificationIntent = intent;
        }
        else {
            notificationIntent = new Intent(context.getApplicationContext(),
MainActivity.class);
            notificationIntent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK |
Intent.FLAG_ACTIVITY_CLEAR_TOP);
        }

        PendingIntent pendingIntent = PendingIntent.getActivity((context), 0,
notificationIntent, PendingIntent.FLAG_UPDATE_CURRENT|
PendingIntent.FLAG_ONE_SHOT);


        NotificationManagerCompat notificationManager =
NotificationManagerCompat.from(context);


        Notification.Builder builder = new Notification.Builder(context);
        if (bitmap != null)
            builder.setStyle(new
Notification.BigPictureStyle().bigPicture(bitmap));


        // Create Notification
        Notification notification = builder
                .setContentTitle(title)
                .setContentText(msg)
                .setTicker(context.getString(R.string.app_name))
                .setSmallIcon(R.drawable.ic_logo)
                .setSound(notificationSound)
                .setLights(Color.RED, 3000, 3000)
                .setVibrate(new long[] { 1000, 1000 })
                .setWhen(System.currentTimeMillis())
                .setDefaults(Notification.DEFAULT_SOUND)
                .setAutoCancel(true)
                .setContentIntent(pendingIntent)
                .build();


        notificationManager.notify(NOTIFICATION_REQUEST_CODE, notification);

    }


}
```

Designing part

XML CODES

MainActivity.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:fitsSystemWindows="true"
    android:orientation="vertical">

    <include
        android:id="@+id/myToolbar"
        layout="@layout/toolbar" />

    <androidx.drawerlayout.widget.DrawerLayout
        android:id="@+id/drawer_layout"
        android:layout_width="match_parent"
        android:layout_height="match_parent">

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

            <FrameLayout
                android:id="@+id/main_fragment"
                android:layout_width="match_parent"
                android:layout_height="match_parent"/>

            <LinearLayout
                android:layout_alignParentBottom="true"
                android:gravity="end"
                android:orientation="vertical"
                android:layout_width="match_parent"
                android:layout_height="wrap_content" >

<com.google.android.material.floatingactionbutton.FloatingActionButton
                    android:id="@+id/settingsFabBtn"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:layout_marginTop="@dimen/margin_16"
                    android:layout_marginStart="@dimen/margin_16"
                    android:layout_marginEnd="@dimen/margin_16"
                    android:layout_marginBottom="@dimen/margin_16"
                    android:src="@drawable/ic_settings"
                    android:tint="@color/white" />

                <com.google.android.material.bottomnavigation.BottomNavigationView
                    android:id="@+id/bottomNavigation"
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:background="@color/white"
                    android:theme="@style/BottomNavigationTheme"
                    app:menu="@menu/bottomnavigaiotn_menu" />
            </LinearLayout>
        </RelativeLayout>

        <com.google.android.material.navigation.NavigationView
            android:id="@+id/main_drawer"
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            android:layout_gravity="start"
            android:background="@color/white"
            app:itemIconTint="@color/black"
            app:itemTextColor="@color/black"
            app:theme="@style/MyNavigationDrawer">

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

                <RelativeLayout
                    android:layout_width="match_parent"
                    android:layout_height="@dimen/drawer_header_height"
                    android:background="?attr/colorPrimaryDark">

                    <ImageView
                        android:id="@+id/drawer_header"
                        android:layout_width="match_parent"
                        android:layout_height="match_parent"
                        android:alpha=".3"
                        android:scaleType="centerCrop"
                        android:src="@drawable/icons_stripe"
                        android:tint="?attr/colorPrimaryDark" />

                    <LinearLayout
                        android:layout_width="match_parent"
                        android:layout_height="match_parent"
                        android:gravity="center_vertical"
                        android:orientation="horizontal"
                        android:paddingStart="15dp"
                        android:paddingTop="5dp"
                        android:paddingEnd="0dp"
                        android:paddingBottom="5dp"
                        android:paddingLeft="15dp"
                        android:paddingRight="0dp">

                        <com.amanakash.ecommerce.customs.CircularImageView
                            android:id="@+id/drawer_profile_image"
                            android:layout_width="60dp"
                            android:layout_height="60dp"
                            android:scaleType="fitCenter"
                            android:src="@drawable/profile"
                            app:border_color="@color/white"
                            app:border_overlay="true"
                            app:border_width="1dp" />

                        <LinearLayout
                            android:layout_width="match_parent"
                            android:layout_height="match_parent"
                            android:gravity="center_vertical"
```

```
                            android:orientation="vertical"
                            android:padding="10dp">

                            <TextView
                                android:id="@+id/drawer_profile_welcome"
                                android:layout_width="wrap_content"
                                android:layout_height="wrap_content"
                                android:text="@string/welcome"
                                android:textColor="@color/windowBackgroundDark"
                                android:textSize="14sp" />

                            <TextView
                                android:id="@+id/drawer_profile_name"
                                android:layout_width="wrap_content"
                                android:layout_height="wrap_content"
                                android:textColor="@color/windowBackground"
                                android:textSize="16sp"
                                android:textStyle="bold" />

                            <TextView
                                android:id="@+id/drawer_profile_email"
                                android:layout_width="wrap_content"
                                android:layout_height="wrap_content"
                                android:textColor="@color/windowBackgroundDark"
                                android:textSize="13sp" />

                        </LinearLayout>

                    </LinearLayout>

                </RelativeLayout>

                <ExpandableListView
                    android:id="@+id/main_drawer_list"
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:background="@color/white"
                    android:choiceMode="singleChoice"
                    android:divider="@color/dividerLight"
                    android:dividerHeight="1dp"
                    android:groupIndicator="@null"

android:indicatorLeft="?android:attr/expandableListPreferredItemIndicatorLeft"
                    android:paddingTop="5dp"
                    android:paddingBottom="5dp" />

            </LinearLayout>

        </com.google.android.material.navigation.NavigationView>

    </androidx.drawerlayout.widget.DrawerLayout>


</LinearLayout>
```

Login.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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"
    android:background="@color/windowBackgroundLight"
    android:orientation="vertical">

    <include
        android:id="@+id/myToolbar"
        layout="@layout/toolbar" />

    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_below="@+id/myToolbar"
        android:fillViewport="true"
        android:isScrollContainer="true"
        android:scrollbars="none">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:gravity="center_horizontal"
            android:orientation="vertical"
            android:padding="20dp">

            <ImageView
                android:id="@+id/logoImg"
                android:layout_width="140dp"
                android:layout_height="140dp"
                android:scaleType="fitXY"
                android:src="@drawable/profile" />

            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_marginTop="20dp"
                android:gravity="center_horizontal"
                android:orientation="vertical">

                <LinearLayout
                    android:id="@+id/layout_email_login"
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:gravity="center_horizontal"
                    android:orientation="vertical">

                    <EditText
                        android:id="@+id/user_email"
                        android:layout_width="match_parent"
                        android:layout_height="@dimen/input_field_height"

android:background="@drawable/rounded_corners_edittext_dark"
```

```
                        android:drawablePadding="15dp"
                        android:gravity="start|center_vertical"
                        android:hint="@string/email_address"
                        android:inputType="textEmailAddress|textNoSuggestions"
                        android:paddingStart="15dp"
                        android:paddingEnd="15dp"
                        android:textAlignment="viewStart"
                        android:textColor="@color/textColorSecondary"
                        android:textColorHint="@color/textColorPrimaryDark" />

                    <EditText
                        android:id="@+id/user_password"
                        android:layout_width="match_parent"
                        android:layout_height="@dimen/input_field_height"
                        android:layout_marginTop="@dimen/input_field_margin"

android:background="@drawable/rounded_corners_edittext_dark"
                        android:drawablePadding="15dp"
                        android:gravity="start|center_vertical"
                        android:hint="@string/password"
                        android:inputType="textPassword|textNoSuggestions"
                        android:paddingStart="15dp"
                        android:paddingEnd="15dp"
                        android:textAlignment="viewStart"
                        android:textColor="@color/textColorSecondary"
                        android:textColorHint="@color/textColorPrimaryDark" />

                    <androidx.appcompat.widget.AppCompatButton
                        android:id="@+id/loginBtn"
                        android:layout_width="match_parent"
                        android:layout_height="@dimen/button_height"
                        android:layout_marginTop="@dimen/button_margin"
                        android:background="@drawable/rounded_corners_button"
                        android:text="@string/login"
                        android:textAllCaps="false"
                        android:textColor="@color/white"
                        android:textSize="@dimen/button_text_size"
                        app:backgroundTint="@color/colorAccentGreen" />

                    <TextView
                        android:id="@+id/forgot_password_text"
                        android:layout_width="wrap_content"
                        android:layout_height="wrap_content"
                        android:layout_marginTop="10dp"
                        android:clickable="true"
                        android:focusable="true"
                        android:text="@string/forgot_password"
                        android:textColor="@color/colorAccentDark"
                        android:textSize="15sp"
                        android:textStyle="italic" />

                </LinearLayout>

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

```
<!--<com.facebook.login.widget.LoginButton
    xmlns:facebook="http://schemas.android.com/apk/res-auto"
    android:id="@+id/facebookLoginBtn"
    android:layout_width="match_parent"
    android:layout_height="40dp"
    android:text="Login with Facebook"
    android:textSize="18sp"
    android:textColor="@color/white"
    android:paddingTop="10dp"
    android:paddingBottom="10dp"
    android:layout_marginTop="25dp"
    android:layout_gravity="center_vertical|center_horizontal"
    facebook:com_facebook_login_text="LOGIN"
    facebook:com_facebook_logout_text="LOGOUT"
    android:drawableTint="@android:color/transparent"
    />-->

<androidx.appcompat.widget.AppCompatButton
    android:id="@+id/phoneLoginBtn"
    android:layout_width="match_parent"
    android:layout_height="@dimen/button_height"
    android:layout_marginTop="@dimen/button_margin"
    android:background="@drawable/rounded_corners_button"
    android:text="@string/phone_login"
    android:textAllCaps="false"
    android:textColor="@color/white"
    android:textSize="@dimen/button_text_size"
    app:backgroundTint="@color/colorAccentGreen" />


<androidx.appcompat.widget.AppCompatButton
    android:id="@+id/facebookLoginBtn"
    android:layout_width="match_parent"
    android:layout_height="@dimen/button_height"
    android:layout_marginTop="15dp"
    android:background="@drawable/rounded_corners_button"
    android:gravity="center"
    android:paddingStart="15dp"
    android:paddingEnd="15dp"
    android:text="@string/facebook_login"
    android:textAlignment="center"
    android:textAllCaps="false"
    android:textColor="@color/white"
    android:textSize="@dimen/button_text_size"
    android:textStyle="normal"
    app:backgroundTint="@color/socialColorFacebook" />

<!--<com.google.android.gms.common.SignInButton
    android:id="@+id/googleLoginBtn"
    android:layout_width="match_parent"
    android:layout_height="40dp"
    android:layout_marginTop="10dp"
    android:background="@color/socialColorGoogle"
    />-->
<androidx.appcompat.widget.AppCompatButton
    android:id="@+id/googleLoginBtn"
    android:layout_width="match_parent"
    android:layout_height="@dimen/button_height"
```

```
                        android:layout_marginTop="10dp"
                        android:background="@drawable/rounded_corners_button"
                        android:gravity="center"
                        android:paddingStart="15dp"
                        android:paddingEnd="15dp"
                        android:text="@string/google_login"
                        android:textAlignment="center"
                        android:textAllCaps="false"
                        android:textColor="@color/white"
                        android:textSize="@dimen/button_text_size"
                        android:textStyle="normal"
                        app:backgroundTint="@color/socialColorGoogle" />

                </LinearLayout>

                <LinearLayout
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:layout_margin="10dp"
                    android:gravity="center"
                    android:orientation="horizontal">

                    <TextView
                        android:layout_width="wrap_content"
                        android:layout_height="wrap_content"
                        android:clickable="true"
                        android:focusable="true"
                        android:text="@string/not_a_member"
                        android:textColor="@color/textColorSecondaryLight"
                        android:textSize="16sp"
                        android:textStyle="normal" />

                    <TextView
                        android:id="@+id/login_signupText"
                        android:layout_width="wrap_content"
                        android:layout_height="wrap_content"
                        android:layout_marginStart="5dp"
                        android:clickable="true"
                        android:focusable="true"
                        android:text="@string/signup"
                        android:textColor="@color/colorAccentGreen"
                        android:textSize="18sp"
                        android:textStyle="bold" />

                </LinearLayout>


            </LinearLayout>

        </LinearLayout>
    </ScrollView>

</RelativeLayout>
```

Dashboard.xml

```
<androidx.coordinatorlayout.widget.CoordinatorLayout
xmlns:android="http://schemas.android.com/apk/res/android"
```

```
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:custom="http://schemas.android.com/tools"
    android:id="@+id/homepage"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">


    <com.google.android.material.appbar.AppBarLayout
        android:id="@+id/appbar_layout"
        app:elevation="0dp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="@color/white"

app:layout_behavior="com.amanakash.ecommerce.customs.AppBarFlingScrollBehavior">

        <FrameLayout
            app:layout_scrollFlags="scroll"
            android:layout_width="match_parent"
            android:layout_height="wrap_content">

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

                <RelativeLayout
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content">
                    <FrameLayout
                        android:id="@+id/bannerFrameHome1"
                        android:layout_width="match_parent"
                        android:layout_height="wrap_content" />

                    <com.facebook.shimmer.ShimmerFrameLayout
                        android:id="@+id/shimmerFrame"
                        android:layout_width="match_parent"
                        android:layout_height="match_parent"
                        android:alpha=".8"
                        android:visibility="gone"
                        app:shimmer_duration="900">

                        <ImageView
                            android:layout_width="match_parent"
                            android:layout_height="match_parent"
                            android:scaleType="centerCrop"
                            android:src="@drawable/placeholder" />
                    </com.facebook.shimmer.ShimmerFrameLayout>
                </RelativeLayout>

                <com.google.android.material.tabs.TabLayout
                    android:id="@+id/tabs"
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    app:tabMode="fixed"
                    app:tabIndicatorHeight="2dp"
                    app:tabBackground="@color/white"
                    android:background="@color/white"
                    app:tabTextColor="@color/textColorSecondaryLight"
                    app:tabIndicatorColor="@color/textColorSecondaryDark"
                    app:tabSelectedTextColor="@color/textColorSecondaryDark"
                    style="@style/MyCustomTabLayout" />

                <com.amanakash.ecommerce.customs.CustomViewPager
                    android:id="@+id/myViewPager"
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:background="@color/windowBackground" />

                <FrameLayout
                    android:id="@+id/flash_sale_fragment"
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:background="@color/windowBackground"/>

                <FrameLayout
                    android:id="@+id/recently_viewed_fragment"
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:orientation="vertical"
                    android:background="@color/windowBackground"/>

            </LinearLayout>

        </FrameLayout>

    </com.google.android.material.appbar.AppBarLayout>


    <FrameLayout
        android:id="@+id/home1_products_fragment"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical"
        android:background="@color/windowBackground"
        app:layout_behavior="@string/appbar_scrolling_view_behavior" />


</androidx.coordinatorlayout.widget.CoordinatorLayout>
```

Updateaccount.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/windowBackground">


    <androidx.core.widget.NestedScrollView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_above="@+id/updateInfoBtn">
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:orientation="vertical"
            android:padding="20dp">

            <RelativeLayout
                android:layout_width="130dp"
                android:layout_height="130dp"
                android:layout_gravity="center_horizontal" >

                <com.amanakash.ecommerce.customs.CircularImageView
                    android:id="@+id/user_photo"
                    android:layout_width="match_parent"
                    android:layout_height="match_parent"
                    android:src="@drawable/profile"
                    app:border_width="2dp"
                    app:border_overlay="true"
                    app:border_color="@color/white" />

<com.google.android.material.floatingactionbutton.FloatingActionButton
                    android:id="@+id/user_photo_edit_fab"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:src="@drawable/ic_edit"
                    android:tint="@color/white"
                    android:visibility="gone"
                    app:backgroundTint="@color/colorAccentBlue"
                    app:rippleColor="@color/colorAccentBlue"
                    app:fabSize="mini"
                    app:elevation="0dp"
                    android:translationZ="0dp"
                    app:pressedTranslationZ="0dp"
                    app:borderWidth="0dp"
                    app:useCompatPadding="true"
                    android:layout_margin="0dp"
                    android:layout_alignParentEnd="true"
                    android:layout_alignParentBottom="true" />

            </RelativeLayout>


            <LinearLayout
                android:id="@+id/updateInfoForm"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:orientation="vertical"
                android:layout_marginTop="@dimen/input_field_margin">

                <EditText
                    android:id="@+id/firstname"
                    android:layout_width="match_parent"
                    android:layout_height="@dimen/input_field_height"
                    android:hint="@string/first_name"
                    android:inputType="textNoSuggestions"
                    android:textColor="@color/textColorSecondary"
                    android:textColorHint="@color/textColorPrimaryDark"
                    android:drawablePadding="15dp"
                    android:paddingStart="15dp"
                    android:paddingEnd="15dp"
                    android:textAlignment="viewStart"
                    android:gravity="start|center_vertical"
                    android:layout_marginTop="@dimen/input_field_margin"
                    android:background="@drawable/rounded_corners_edittext_dark"
/>
                <EditText
                    android:id="@+id/lastname"
                    android:layout_width="match_parent"
                    android:layout_height="@dimen/input_field_height"
                    android:hint="@string/last_name"
                    android:inputType="textNoSuggestions"
                    android:textColor="@color/textColorSecondary"
                    android:textColorHint="@color/textColorPrimaryDark"
                    android:drawablePadding="15dp"
                    android:paddingStart="15dp"
                    android:paddingEnd="15dp"
                    android:textAlignment="viewStart"
                    android:gravity="start|center_vertical"
                    android:layout_marginTop="@dimen/input_field_margin"
                    android:background="@drawable/rounded_corners_edittext_dark"
/>
                <!--<EditText
                    android:id="@+id/email"
                    android:layout_width="match_parent"
                    android:layout_height="@dimen/input_field_height"
                    android:hint="@string/email_address"
                    android:inputType="textEmailAddress|textNoSuggestions"
                    android:textColor="@color/textColorSecondary"
                    android:textColorHint="@color/textColorPrimaryDark"
                    android:drawableStart="@drawable/ic_email"
                    android:drawablePadding="15dp"
                    android:paddingStart="15dp"
                    android:paddingEnd="15dp"
                    android:textAlignment="viewStart"
                    android:gravity="start|center_vertical"
                    android:layout_marginTop="@dimen/input_field_margin"
                    android:background="@drawable/rounded_corners_edittext_dark"
/>-->
                <EditText
                    android:visibility="gone"
                    android:id="@+id/dob"
                    android:layout_width="match_parent"
                    android:layout_height="@dimen/input_field_height"
                    android:hint="@string/dob"
                    android:inputType="none"
                    android:enabled="false"
                    android:textColor="@color/textColorSecondary"
                    android:textColorHint="@color/textColorPrimaryDark"
                    android:drawablePadding="15dp"
                    android:paddingStart="15dp"
                    android:paddingEnd="15dp"
                    android:textAlignment="viewStart"
                    android:gravity="start|center_vertical"
                    android:layout_marginTop="@dimen/input_field_margin"
                    android:background="@drawable/rounded_corners_edittext_dark"
```

```
/>
                <EditText
                    android:id="@+id/contact"
                    android:layout_width="match_parent"
                    android:layout_height="@dimen/input_field_height"
                    android:hint="@string/contact_no"
                    android:inputType="phone"
                    android:enabled="false"
                    android:textColor="@color/textColorSecondary"
                    android:textColorHint="@color/textColorPrimaryDark"
                    android:drawablePadding="15dp"
                    android:paddingStart="15dp"
                    android:paddingEnd="15dp"
                    android:textAlignment="viewStart"
                    android:gravity="start|center_vertical"
                    android:layout_marginTop="@dimen/input_field_margin"
                    android:background="@drawable/rounded_corners_edittext_dark"
/>

<!--                <EditText
                    android:id="@+id/current_password"
                    android:layout_width="match_parent"
                    android:layout_height="@dimen/input_field_height"
                    android:hint="@string/current_password"
                    android:inputType="textPassword|textNoSuggestions"
                    android:textColor="@color/textColorSecondary"
                    android:textColorHint="@color/textColorPrimaryDark"
                    android:drawablePadding="15dp"
                    android:paddingStart="15dp"
                    android:paddingEnd="15dp"
                    android:textAlignment="viewStart"
                    android:gravity="start|center_vertical"
                    android:layout_marginTop="@dimen/input_field_margin"
                    android:background="@drawable/rounded_corners_edittext_dark"
/>
                <EditText
                    android:id="@+id/new_password"
                    android:layout_width="match_parent"
                    android:layout_height="@dimen/input_field_height"
                    android:hint="@string/new_password"
                    android:inputType="textPassword|textNoSuggestions"
                    android:textColor="@color/textColorSecondary"
                    android:textColorHint="@color/textColorPrimaryDark"
                    android:drawablePadding="15dp"
                    android:paddingStart="15dp"
                    android:paddingEnd="15dp"
                    android:textAlignment="viewStart"
                    android:gravity="start|center_vertical"
                    android:layout_marginTop="@dimen/input_field_margin"
                    android:background="@drawable/rounded_corners_edittext_dark"
/>-->
            </LinearLayout>

        </LinearLayout>
    </androidx.core.widget.NestedScrollView>

    <androidx.appcompat.widget.AppCompatButton
        android:id="@+id/updateInfoBtn"
        android:layout_width="match_parent"
        android:layout_height="@dimen/button_height"
        android:text="@string/save_changes"
        android:textColor="@color/white"
        android:textSize="@dimen/button_text_size"
        android:background="@color/colorAccent"
        android:layout_alignParentBottom="true"/>

</RelativeLayout>
```

Products.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:background="@color/windowBackground"
    app:layout_behavior="@string/appbar_scrolling_view_behavior">

    <com.google.android.material.tabs.TabLayout
        android:id="@+id/product_tabs"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:tabMode="scrollable"
        app:tabIndicatorHeight="2dp"
        app:tabBackground="@color/white"
        android:background="@color/white"
        app:tabTextColor="@color/textColorSecondaryLight"
        app:tabIndicatorColor="@color/textColorSecondaryDark"
        app:tabSelectedTextColor="@color/textColorSecondaryDark"
        style="@style/MyCustomTabLayout"
        app:layout_behavior="@string/appbar_scrolling_view_behavior" />

    <com.amanakash.ecommerce.customs.CustomViewPager
        android:id="@+id/product_viewpager"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:background="@color/windowBackground"
        app:layout_behavior="@string/appbar_scrolling_view_behavior" />

</LinearLayout>
```

Mycart.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/windowBackground"
    android:clickable="true"
    android:focusable="true"
    android:orientation="vertical">

    <LinearLayout
```

```
        android:id="@+id/cart_view_empty"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_centerInParent="true"
        android:gravity="center"
        android:orientation="vertical"
        android:padding="20dp">

        <ImageView
            android:layout_width="80dp"
            android:layout_height="80dp"
            android:src="@drawable/ic_cart"
            android:tint="@color/grey_dark" />

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/empty_cart"
            android:textAppearance="@style/TextAppearance.AppCompat.Large" />

        <TextView
            android:id="@+id/empty_cart"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center_horizontal"
            android:text="@string/continue_shopping"
            android:textColor="@color/textColorSecondaryLight"
            android:textSize="18sp" />

        <androidx.appcompat.widget.AppCompatButton
            android:id="@+id/continue_shopping_btn"
            android:layout_width="wrap_content"
            android:layout_height="35dp"
            android:layout_gravity="center_horizontal"
            android:layout_margin="@dimen/button_margin"
            android:background="@drawable/rounded_corners_button_green"
            android:text="@string/explore"
            android:textColor="@color/white"
            android:textSize="@dimen/button_text_size"
            app:backgroundTint="@color/colorAccentGreen" />

    </LinearLayout>


    <LinearLayout
        android:id="@+id/cart_view"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">

        <androidx.core.widget.NestedScrollView
            android:id="@+id/mainRvLayout"
            android:layout_weight="1"
            android:layout_width="match_parent"
            android:layout_height="0dp">

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

                <androidx.recyclerview.widget.RecyclerView
                    android:id="@+id/cart_items_recycler"
                    android:layout_width="match_parent"
                    android:layout_height="0dip"
                    android:layout_margin="5dp"
                    android:layout_weight="1" />

                <LinearLayout
                    android:layout_width="match_parent"
                    android:layout_height="match_parent"
                    android:layout_marginStart="10dp"
                    android:layout_marginEnd="10dp"
                    android:background="@color/white"
                    android:orientation="vertical"
                    android:padding="12dp">

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

                        <TextView
                            android:layout_width="wrap_content"
                            android:layout_height="wrap_content"
                            android:text="@string/subtotal"
                            android:textColor="@color/textColorSecondaryDark"
                            android:textSize="15sp" />

                        <TextView
                            android:id="@+id/cart_item_subtotal_price"
                            android:layout_width="match_parent"
                            android:layout_height="wrap_content"
                            android:gravity="end"
                            android:text="$00.0"
                            android:textColor="@color/colorAccentGreen"
                            android:textSize="15sp"
                            android:textStyle="bold" />

                    </LinearLayout>

                    <LinearLayout
                        android:layout_width="match_parent"
                        android:layout_height="wrap_content"
                        android:layout_marginTop="10dp"
                        android:orientation="horizontal">

                        <TextView
                            android:layout_width="wrap_content"
                            android:layout_height="wrap_content"
                            android:text="@string/discount"
                            android:textColor="@color/textColorSecondaryDark"
                            android:textSize="15sp" />

                        <TextView
                            android:id="@+id/cart_item_discount_price"
```

```
                        android:layout_width="match_parent"
                        android:layout_height="wrap_content"
                        android:gravity="end"
                        android:text="$00.0"
                        android:textColor="@color/colorAccentGreen"
                        android:textSize="15sp"
                        android:textStyle="bold" />

                </LinearLayout>

                <LinearLayout
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:layout_marginTop="10dp"
                    android:orientation="horizontal">

                    <TextView
                        android:layout_width="wrap_content"
                        android:layout_height="wrap_content"
                        android:text="@string/total"
                        android:textColor="@color/textColorSecondaryDark"
                        android:textSize="15sp"
                        android:textStyle="bold" />

                    <TextView
                        android:id="@+id/cart_item_total_price"
                        android:layout_width="match_parent"
                        android:layout_height="wrap_content"
                        android:gravity="end"
                        android:text="$00.0"
                        android:textColor="@color/colorAccentGreen"
                        android:textSize="15sp"
                        android:textStyle="bold" />

                </LinearLayout>

            </LinearLayout>
        </LinearLayout>
    </androidx.core.widget.NestedScrollView>


    <LinearLayout
        android:id="@+id/cart_bottom_bar"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="@color/white"
        android:orientation="vertical">

        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:padding="10dp">

            <!--<TextView
                android:id="@+id/total"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="@string/total"
                android:textSize="16sp"
                android:textStyle="bold"
                android:textAlignment="textStart"
                android:textColor="@color/textColorSecondary"
                android:gravity="start"
                android:layout_marginStart="5dp"
                android:layout_centerVertical="true"
                android:layout_alignParentStart="true" />-->

            <TextView
                android:id="@+id/cart_total_price"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_alignParentStart="true"
                android:layout_centerVertical="true"
                android:layout_marginStart="10dp"
                android:gravity="start"
                android:padding="5dp"
                android:text="@string/total_price"
                android:textAlignment="textStart"
                android:textColor="@color/colorAccentGreen"
                android:textSize="18sp"
                android:textStyle="bold" />

            <Button
                android:id="@+id/cart_checkout_btn"
                android:layout_width="wrap_content"
                android:layout_height="35dp"
                android:layout_alignParentEnd="true"
                android:layout_centerVertical="true"
                android:layout_marginEnd="5dp"
                android:background="@drawable/rounded_corners_button_green"
                android:text="@string/checkoutFinal"
                android:textAlignment="center"
                android:textAllCaps="false"
                android:textColor="@color/white"
                android:textSize="16sp" />

        </RelativeLayout>

        <!--<LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="horizontal" >

            <LinearLayout
                android:layout_weight=".5"
                android:layout_width="0dip"
                android:layout_height="match_parent"
                android:orientation="horizontal"
                android:gravity="center"
                android:background="@color/colorAccent"
                android:padding="5dp" >

                <TextView
                    android:id="@+id/cart_total_price"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
```

```
                        android:text="Total $0.00"
                        android:textColor="@color/white"
                        android:textSize="18sp"
                        android:textAlignment="textStart"
                        android:gravity="start"
                        android:padding="5dp" />

                </LinearLayout>

                <Button
                    android:id="@+id/cart_checkout_btn"
                    android:layout_weight=".5"
                    android:layout_width="0dip"
                    android:layout_height="match_parent"
                    android:text="@string/checkoutFinal"
                    android:textAllCaps="false"
                    android:textSize="18sp"
                    android:textColor="@color/white"
                    android:textAlignment="center"
                    android:background="@color/colorPrimary" />

            </LinearLayout>-->

        </LinearLayout>


    </LinearLayout>

</RelativeLayout>
```

Myaddress.xml

```
<RelativeLayout 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"
    android:orientation="vertical"
    android:padding="10dp"
    android:background="@color/windowBackgroundDark">

    <LinearLayout
        android:gravity="center"
        android:id="@+id/empty_record"
        android:orientation="vertical"
        android:layout_centerInParent="true"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">

        <ImageView
            android:tint="@color/grey"
            android:src="@drawable/ic_location"
            android:layout_width="140dp"
            android:layout_height="140dp" />

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/record_not_found"
            android:textAppearance="@style/TextAppearance.AppCompat.Large"
            android:textColor="@color/textColorSecondaryDark" />

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/enter_address"
            android:textAppearance="@style/TextAppearance.AppCompat.Small"
            android:textColor="@color/textColorSecondaryLight" />

        <androidx.appcompat.widget.AppCompatButton
            android:id="@+id/continue_shopping_btn"
            android:layout_width="wrap_content"
            android:layout_height="35dp"
            android:paddingStart="12dp"
            android:paddingEnd="12dp"
            android:text="@string/new_address"
            android:textColor="@color/white"
            android:textSize="@dimen/button_text_size"
            android:layout_gravity="center_horizontal"
            android:layout_margin="@dimen/button_margin"
            app:backgroundTint="@color/colorAccentGreen"
            android:background="@drawable/rounded_corners_button_green" />
    </LinearLayout>

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

        <androidx.recyclerview.widget.RecyclerView
            android:id="@+id/addresses_recycler"
            android:layout_width="match_parent"
            android:layout_height="wrap_content" />

        <com.google.android.material.floatingactionbutton.FloatingActionButton
            android:id="@+id/add_address_fab"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            app:fabSize="normal"
            android:src="@drawable/ic_add_location"
            android:tint="@color/white"
            android:elevation="5dp"
            android:translationZ="3dp"
            android:layout_gravity="bottom|end"
            android:layout_margin="@dimen/fab_margin"
            app:backgroundTint="@color/colorAccentGreen" />

    </FrameLayout>

</RelativeLayout>
```

Checkout.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
```

```
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:card_view="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical"
android:clickable="true"
android:focusable="true"
android:background="@color/layoutBackgroundLight">


<androidx.core.widget.NestedScrollView
    android:id="@+id/scroll_container"
    android:layout_width="match_parent"
    android:layout_height="0dip"
    android:layout_weight="1"
    android:padding="15dp" >

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


        <androidx.cardview.widget.CardView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:background="@color/white"
            app:cardElevation="0dp"
            app:cardCornerRadius="2dp"
            card_view:cardCornerRadius="2dp"
            card_view:cardUseCompatPadding="true"
            card_view:cardPreventCornerOverlap="false" >

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

                <RelativeLayout
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:padding="5dp"
                    android:gravity="center_vertical"
                    android:background="@color/windowBackgroundLight">

                    <TextView
                        android:layout_width="wrap_content"
                        android:layout_height="wrap_content"
                        android:text="@string/billing_address"
                        android:textSize="16sp"
                        android:textColor="@color/textColorSecondaryDark" />

                    <ImageButton
                        android:visibility="gone"
                        android:id="@+id/checkout_edit_billing"
                        android:layout_width="20dp"
                        android:layout_height="20dp"
                        android:src="@drawable/ic_edit"
                        android:tint="@color/colorAccent"
                        android:layout_alignParentEnd="true"
                        android:layout_marginEnd="5dp"
                        android:background="@android:color/transparent" />

                </RelativeLayout>

                <LinearLayout
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:orientation="vertical"
                    android:padding="5dp" >

                    <TextView
                        android:id="@+id/billing_name"
                        android:layout_width="match_parent"
                        android:layout_height="wrap_content"
                        android:maxLines="1"
                        android:text="@string/full_name"
                        android:textSize="14sp"
                        android:textAlignment="viewStart"
                        android:textColor="@color/textColorSecondary" />

                    <TextView
                        android:id="@+id/billing_address"
                        android:layout_width="match_parent"
                        android:layout_height="wrap_content"
                        android:maxLines="1"
                        android:text="@string/city"
                        android:textSize="14sp"
                        android:textAlignment="viewStart"
                        android:textColor="@color/textColorSecondary" />

                    <TextView
                        android:id="@+id/billing_street"
                        android:layout_width="match_parent"
                        android:layout_height="wrap_content"
                        android:maxLines="1"
                        android:text="@string/address"
                        android:textSize="14sp"
                        android:textAlignment="viewStart"
                        android:textColor="@color/textColorSecondary" />

                </LinearLayout>

            </LinearLayout>

        </androidx.cardview.widget.CardView>


        <androidx.cardview.widget.CardView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:background="@color/white"
            android:layout_marginTop="10dp"
            app:cardElevation="0dp"
            app:cardCornerRadius="2dp"
            card_view:cardCornerRadius="2dp"
```

```
card_view:cardUseCompatPadding="true"
card_view:cardPreventCornerOverlap="false" >

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

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:padding="5dp"
        android:gravity="center_vertical"
        android:background="@color/windowBackgroundLight">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/shipping_address"
            android:textSize="16sp"
            android:textColor="@color/textColorSecondaryDark" />

        <ImageButton
            android:visibility="gone"
            android:id="@+id/checkout_edit_shipping"
            android:layout_width="20dp"
            android:layout_height="20dp"
            android:src="@drawable/ic_edit"
            android:tint="@color/colorAccent"
            android:layout_alignParentEnd="true"
            android:layout_marginEnd="5dp"
            android:background="@android:color/transparent" />

    </RelativeLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:padding="5dp" >

        <TextView
            android:id="@+id/shipping_name"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:maxLines="1"
            android:text="@string/full_name"
            android:textSize="14sp"
            android:textAlignment="viewStart"
            android:textColor="@color/textColorSecondary" />

        <TextView
            android:id="@+id/shipping_address"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:maxLines="1"
            android:text="@string/city"
            android:textSize="14sp"
```

```
                android:textAlignment="viewStart"
                android:textColor="@color/textColorSecondary" />

            <TextView
                android:id="@+id/shipping_street"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:maxLines="1"
                android:text="@string/address"
                android:textSize="14sp"
                android:textAlignment="viewStart"
                android:textColor="@color/textColorSecondary" />

        </LinearLayout>

    </LinearLayout>

</androidx.cardview.widget.CardView>


<androidx.cardview.widget.CardView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="@color/white"
    android:layout_marginTop="10dp"
    app:cardElevation="0dp"
    app:cardCornerRadius="2dp"
    card_view:cardCornerRadius="2dp"
    card_view:cardUseCompatPadding="true"
    card_view:cardPreventCornerOverlap="false"  >

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

        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:padding="5dp"
            android:gravity="center_vertical"
            android:background="@color/windowBackgroundLight">

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="@string/shipping_method"
                android:textSize="16sp"
                android:textColor="@color/textColorSecondaryDark" />

            <ImageButton
                android:visibility="gone"
                android:id="@+id/checkout_edit_shipping_method"
                android:layout_width="20dp"
                android:layout_height="20dp"
                android:src="@drawable/ic_edit"
                android:tint="@color/colorAccent"
                android:layout_alignParentEnd="true"
```

```
                            android:layout_marginEnd="5dp"
                            android:background="@android:color/transparent" />

                    </RelativeLayout>

                    <TextView
                        android:id="@+id/shipping_method"
                        android:layout_width="match_parent"
                        android:layout_height="wrap_content"
                        android:maxLines="1"
                        android:padding="5dp"
                        android:text="@string/shipping_method"
                        android:textSize="14sp"
                        android:textAlignment="viewStart"
                        android:textColor="@color/textColorSecondary" />

                </LinearLayout>

            </androidx.cardview.widget.CardView>


            <androidx.cardview.widget.CardView
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:background="@color/white"
                android:layout_marginTop="10dp"
                app:cardElevation="0dp"
                app:cardCornerRadius="2dp"
                card_view:cardCornerRadius="2dp"
                card_view:cardUseCompatPadding="true"
                card_view:cardPreventCornerOverlap="false" >

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

                    <TextView
                        android:layout_width="match_parent"
                        android:layout_height="wrap_content"
                        android:text="@string/products"
                        android:textSize="16sp"
                        android:textColor="@color/textColorSecondaryDark"
                        android:padding="5dp"
                        android:background="@color/windowBackgroundLight" />

                    <androidx.recyclerview.widget.RecyclerView
                        android:id="@+id/checkout_items_recycler"
                        android:layout_width="match_parent"
                        android:layout_height="wrap_content" />

                </LinearLayout>

            </androidx.cardview.widget.CardView>


            <androidx.cardview.widget.CardView
                android:layout_width="match_parent"
```

```
android:layout_height="wrap_content"
android:background="@color/white"
android:layout_marginTop="10dp"
app:cardElevation="0dp"
app:cardCornerRadius="2dp"
card_view:cardCornerRadius="2dp"
card_view:cardUseCompatPadding="true"
card_view:cardPreventCornerOverlap="false" >

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

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/subtotal"
        android:textSize="16sp"
        android:textColor="@color/textColorSecondaryDark"
        android:padding="5dp"
        android:background="@color/windowBackgroundLight" />

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:padding="5dp" >

        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="horizontal"
            android:gravity="center_vertical" >

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:maxLines="1"
                android:text="@string/subtotal"
                android:textSize="14sp"
                android:textColor="@color/textColorSecondary" />

            <TextView
                android:id="@+id/checkout_subtotal"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:maxLines="1"
                android:text="$000.0"
                android:textSize="14sp"
                android:textColor="@color/textColorSecondary"
                android:textAlignment="textEnd"
                android:gravity="end"
                android:layout_alignParentEnd="true" />

        </RelativeLayout>
        <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:gravity="center_vertical"
        android:layout_marginTop="5dp" >

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:maxLines="1"
            android:text="@string/tax"
            android:textSize="14sp"
            android:textColor="@color/textColorSecondary" />

        <TextView
            android:id="@+id/checkout_tax"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:maxLines="1"
            android:text="$00.0"
            android:textSize="14sp"
            android:textColor="@color/textColorSecondary"
            android:textAlignment="textEnd"
            android:gravity="end"
            android:layout_alignParentEnd="true" />

    </RelativeLayout>

        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="horizontal"
            android:gravity="center_vertical"
            android:layout_marginTop="5dp" >

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:maxLines="1"
                android:text="@string/packing_charge"
                android:textSize="14sp"
                android:textColor="@color/textColorSecondary" />

            <TextView
                android:id="@+id/checkout_packing_charges"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:maxLines="1"
                android:text="$00.0"
                android:textSize="14sp"
                android:textColor="@color/textColorSecondary"
                android:textAlignment="textEnd"
                android:gravity="end"
                android:layout_alignParentEnd="true" />

        </RelativeLayout>

        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="horizontal"
            android:gravity="center_vertical"
            android:layout_marginTop="5dp" >

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:maxLines="1"
                android:text="@string/shipping"
                android:textSize="14sp"
                android:textColor="@color/textColorSecondary" />

            <TextView
                android:id="@+id/checkout_shipping"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:maxLines="1"
                android:text="$00.0"
                android:textSize="14sp"
                android:textColor="@color/textColorSecondary"
                android:textAlignment="textEnd"
                android:gravity="end"
                android:layout_alignParentEnd="true" />

        </RelativeLayout>
        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="horizontal"
            android:gravity="center_vertical"
            android:layout_marginTop="5dp" >

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:maxLines="1"
                android:text="@string/discount"
                android:textSize="14sp"
                android:textColor="@color/textColorSecondary" />

            <TextView
                android:id="@+id/checkout_discount"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:maxLines="1"
                android:text="$00.0"
                android:textSize="14sp"
                android:textColor="@color/textColorSecondary"
                android:textAlignment="textEnd"
                android:gravity="end"
                android:layout_alignParentEnd="true" />

        </RelativeLayout>
        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="horizontal"
```

```
                android:gravity="center_vertical"
                android:layout_marginTop="5dp" >

                <TextView
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:maxLines="1"
                    android:text="@string/total"
                    android:textSize="14sp"
                    android:textColor="@color/textColorSecondary" />

                <TextView
                    android:id="@+id/checkout_total"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:maxLines="1"
                    android:text="$000.0"
                    android:textSize="14sp"
                    android:textColor="@color/textColorSecondary"
                    android:textAlignment="textEnd"
                    android:gravity="end"
                    android:layout_alignParentEnd="true" />

            </RelativeLayout>

        </LinearLayout>

    </LinearLayout>

</androidx.cardview.widget.CardView>


<androidx.recyclerview.widget.RecyclerView
    android:id="@+id/checkout_coupons_recycler"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" />


<androidx.cardview.widget.CardView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="@color/white"
    android:layout_marginTop="10dp"
    app:cardElevation="0dp"
    app:cardCornerRadius="2dp"
    card_view:cardCornerRadius="2dp"
    card_view:cardUseCompatPadding="true"
    card_view:cardPreventCornerOverlap="false" >

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:padding="5dp" >

        <androidx.appcompat.widget.AppCompatButton
            android:id="@+id/checkout_coupon_btn"
            android:layout_width="wrap_content"
            android:layout_height="35dp"
```

```
                        android:text="@string/apply"
                        android:textColor="@color/white"
                        android:textAllCaps="false"
                        android:layout_marginStart="5dp"
                        android:layout_centerVertical="true"
                        android:layout_alignParentEnd="true"

android:background="@drawable/rounded_corners_button_green" />

                    <!--<EditText
                        android:id="@+id/checkout_coupon_code"
                        android:layout_width="match_parent"
                        android:layout_height="35dp"
                        android:hint="@string/coupon_code"
                        android:textSize="14sp"
                        android:textColor="@color/textColorSecondary"
                        android:textColorHint="@color/textColorSecondaryLight"
                        android:paddingEnd="15dp"
                        android:paddingStart="15dp"
                        android:drawablePadding="15dp"
                        android:gravity="start|center_vertical"
                        android:drawableStart="@drawable/ic_credit_card"
                        android:layout_toStartOf="@+id/checkout_coupon_btn"

android:background="@drawable/rounded_corners_edittext_light" />-->

                    <EditText
                        android:id="@+id/checkout_coupon_code"
                        android:layout_width="match_parent"
                        android:layout_height="35dp"
                        android:hint="@string/coupon_code"
                        android:inputType="textNoSuggestions"
                        android:textSize="14sp"
                        android:textColor="@color/textColorSecondaryDark"
                        android:padding="5dp"
                        android:layout_centerVertical="true"
                        android:layout_toStartOf="@+id/checkout_coupon_btn"
                        android:background="@color/windowBackgroundLight" />

                </RelativeLayout>

            </androidx.cardview.widget.CardView>



            <TextView
                android:id="@+id/demo_coupons_text"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="@string/sample_coupons"
                android:textSize="14sp"
                android:textStyle="italic"
                android:clickable="true"
                android:layout_marginTop="5dp"
                android:textColor="@color/colorAccentDark" />
```

```
<androidx.cardview.widget.CardView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="@color/white"
    android:layout_marginTop="15dp"
    app:cardElevation="0dp"
    app:cardCornerRadius="2dp"
    card_view:cardCornerRadius="2dp"
    card_view:cardUseCompatPadding="true"
    card_view:cardPreventCornerOverlap="false" >

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

        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="@string/order_notes"
            android:textSize="16sp"
            android:textColor="@color/textColorSecondaryDark"
            android:padding="5dp"
            android:background="@color/windowBackgroundLight" />

        <EditText
            android:id="@+id/checkout_comments"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:inputType="textMultiLine"
            android:maxLines="3"
            android:minLines="3"
            android:textSize="14sp"
            android:hint="@string/order_notes"
            android:padding="5dp"
            android:textAlignment="viewStart"
            android:gravity="start|top"
            android:textColor="@color/textColorSecondary"
            android:background="@color/white" />


    </LinearLayout>

</androidx.cardview.widget.CardView>

<TextView
    android:textSize="16sp"
    android:paddingTop="10dp"
    android:paddingBottom="10dp"
    android:paddingStart="5dp"
    android:paddingEnd="10dp"
    android:layout_marginTop="15dp"
    android:background="@color/windowBackgroundLight"
    android:text="@string/payment_method"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:paddingLeft="5dp"
    android:paddingRight="10dp" />
```

```
<TextView
    android:id="@+id/payment_method"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="@string/payment_method"
    android:maxLines="1"
    android:textSize="16sp"
    android:paddingTop="10dp"
    android:paddingBottom="10dp"
    android:paddingStart="5dp"
    android:paddingEnd="10dp"
    android:clickable="true"
    android:gravity="center_vertical"
    android:background="@color/white"
    android:drawableEnd="@drawable/ic_chevron_down"
    android:textColor="@color/textColorSecondaryDark"/>


<androidx.cardview.widget.CardView
    android:id="@+id/card_details_layout"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="10dp"
    android:background="@color/white"
    app:cardCornerRadius="2dp"
    app:cardElevation="0dp"
    card_view:cardCornerRadius="2dp"
    card_view:cardPreventCornerOverlap="false"
    card_view:cardUseCompatPadding="true">

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

        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="@string/payment_details"
            android:textSize="16sp"
            android:textColor="@color/textColorSecondaryDark"
            android:padding="5dp"
            android:layout_marginBottom="5dp"
            android:background="@color/windowBackgroundLight" />

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_margin="5dp"
            android:background="@color/white"
            android:orientation="vertical">

            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:orientation="vertical"
                android:padding="5dp"
```

```
                            android:layout_marginBottom="5dp">


<com.braintreepayments.cardform.view.SupportedCardTypesView
                                android:id="@+id/supported_card_types"
                                android:layout_width="match_parent"
                                android:layout_height="wrap_content"
                                android:gravity="center"
                                android:layout_marginTop="5dp" />

                            <TextView
                                android:layout_width="wrap_content"
                                android:layout_height="wrap_content"
                                android:maxLines="1"
                                android:layout_marginTop="15dp"
                                android:text="@string/card_number"
                                android:textSize="14sp"
                                android:textColor="@color/textColorSecondary" />
                            <EditText
                                android:id="@+id/checkout_card_number"
                                android:layout_width="match_parent"
                                android:layout_height="@dimen/input_field_height"
                                android:layout_marginTop="5dp"
                                android:drawablePadding="15dp"
                                android:drawableStart="@drawable/ic_credit_card"
                                android:gravity="start|center_vertical"
                                android:hint="@string/card_number"
                                android:inputType="number"
                                android:paddingEnd="15dp"
                                android:paddingStart="15dp"
                                android:textColor="@color/textColorSecondary"

android:textColorHint="@color/textColorSecondaryLight"

android:background="@drawable/rounded_corners_edittext_light" />

                            <TextView
                                android:layout_width="wrap_content"
                                android:layout_height="wrap_content"
                                android:maxLines="1"
                                android:layout_marginTop="15dp"
                                android:text="@string/card_cvv"
                                android:textSize="14sp"
                                android:textColor="@color/textColorSecondary" />
                            <EditText
                                android:id="@+id/checkout_card_cvv"
                                android:layout_width="match_parent"
                                android:layout_height="@dimen/input_field_height"
                                android:layout_marginTop="5dp"
                                android:drawablePadding="15dp"
                                android:drawableStart="@drawable/ic_credit_card"
                                android:gravity="start|center_vertical"
                                android:hint="@string/card_cvv"
                                android:inputType="number"
                                android:paddingEnd="15dp"
                                android:paddingStart="15dp"
                                android:textColor="@color/textColorSecondary"
```

```
android:textColorHint="@color/textColorSecondaryLight"

android:background="@drawable/rounded_corners_edittext_light" />

                            <TextView
                                android:layout_width="wrap_content"
                                android:layout_height="wrap_content"
                                android:maxLines="1"
                                android:layout_marginTop="15dp"
                                android:text="@string/card_expiry"
                                android:textSize="14sp"
                                android:textColor="@color/textColorSecondary" />
                            <EditText
                                android:id="@+id/checkout_card_expiry"
                                android:layout_width="match_parent"
                                android:layout_height="@dimen/input_field_height"
                                android:layout_marginTop="5dp"
                                android:drawableEnd="@drawable/ic_chevron_down"
                                android:drawableStart="@drawable/ic_calendar"
                                android:drawablePadding="15dp"
                                android:gravity="start|center_vertical"
                                android:hint="@string/card_expiry"
                                android:inputType="none"
                                android:paddingEnd="15dp"
                                android:paddingStart="15dp"
                                android:textAlignment="viewStart"
                                android:textColor="@color/textColorSecondary"

android:textColorHint="@color/textColorSecondaryLight"

android:background="@drawable/rounded_corners_edittext_light"/>

                        </LinearLayout>

                    </LinearLayout>

                </LinearLayout>

            </androidx.cardview.widget.CardView>

            <androidx.cardview.widget.CardView
                android:id="@+id/payment_details_layout"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_marginTop="10dp"
                android:background="@color/white"
                app:cardCornerRadius="2dp"
                app:cardElevation="0dp"
                card_view:cardCornerRadius="2dp"
                card_view:cardPreventCornerOverlap="false"
                card_view:cardUseCompatPadding="true">

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

                    <TextView
```

```
                        android:layout_width="match_parent"
                        android:layout_height="wrap_content"
                        android:text="@string/payment_details"
                        android:textSize="16sp"
                        android:textColor="@color/textColorSecondaryDark"
                        android:padding="5dp"
                        android:layout_marginBottom="5dp"
                        android:background="@color/windowBackgroundLight" />

                    <LinearLayout
                        android:layout_width="match_parent"
                        android:layout_height="match_parent"
                        android:layout_margin="5dp"
                        android:background="@color/white"
                        android:orientation="vertical">

                        <LinearLayout
                            android:layout_width="match_parent"
                            android:layout_height="wrap_content"
                            android:orientation="vertical"
                            android:padding="5dp"
                            android:layout_marginBottom="5dp">

                            <TextView
                                android:layout_width="wrap_content"
                                android:layout_height="wrap_content"
                                android:maxLines="1"
                                android:layout_marginTop="5dp"
                                android:text="@string/full_name"
                                android:textSize="14sp"
                                android:textColor="@color/textColorSecondary" />
                            <EditText
                                android:id="@+id/payment_name"
                                android:layout_width="match_parent"
                                android:layout_height="@dimen/input_field_height"
                                android:layout_marginTop="5dp"
                                android:drawablePadding="15dp"
                                android:drawableStart="@drawable/ic_person"
                                android:gravity="start|center_vertical"
                                android:hint="@string/full_name"
                                android:inputType="text"
                                android:paddingEnd="15dp"
                                android:paddingStart="15dp"
                                android:textColor="@color/textColorSecondary"

android:textColorHint="@color/textColorSecondaryLight"

android:background="@drawable/rounded_corners_edittext_light" />

                            <TextView
                                android:layout_width="wrap_content"
                                android:layout_height="wrap_content"
                                android:maxLines="1"
                                android:layout_marginTop="15dp"
                                android:text="@string/email_address"
                                android:textSize="14sp"
                                android:textColor="@color/textColorSecondary" />
                            <EditText
```

```
                                android:id="@+id/payment_email"
                                android:layout_width="match_parent"
                                android:layout_height="@dimen/input_field_height"
                                android:layout_marginTop="5dp"
                                android:drawablePadding="15dp"
                                android:drawableStart="@drawable/ic_email"
                                android:gravity="start|center_vertical"
                                android:hint="@string/email_address"
                                android:inputType="textEmailAddress"
                                android:paddingEnd="15dp"
                                android:paddingStart="15dp"
                                android:textColor="@color/textColorSecondary"

android:textColorHint="@color/textColorSecondaryLight"

android:background="@drawable/rounded_corners_edittext_light" />

                            <TextView
                                android:layout_width="wrap_content"
                                android:layout_height="wrap_content"
                                android:maxLines="1"
                                android:layout_marginTop="15dp"
                                android:text="@string/contact_no"
                                android:textSize="14sp"
                                android:textColor="@color/textColorSecondary" />
                            <EditText
                                android:id="@+id/payment_phone"
                                android:layout_width="match_parent"
                                android:layout_height="@dimen/input_field_height"
                                android:layout_marginTop="5dp"
                                android:drawablePadding="15dp"
                                android:drawableStart="@drawable/ic_phone_android"
                                android:gravity="start|center_vertical"
                                android:hint="@string/contact_no"
                                android:inputType="phone"
                                android:paddingEnd="15dp"
                                android:paddingStart="15dp"
                                android:textColor="@color/textColorSecondary"

android:textColorHint="@color/textColorSecondaryLight"

android:background="@drawable/rounded_corners_edittext_light" />

                            <androidx.appcompat.widget.AppCompatButton
                                android:id="@+id/checkout_payment_btn"
                                android:layout_width="match_parent"
                                android:layout_height="40dp"
                                android:text="@string/apply"
                                android:textSize="18sp"
                                android:textColor="@color/white"
                                android:textStyle="italic|bold"
                                android:layout_gravity="center_horizontal"
                                android:layout_marginTop="15dp"

android:background="@drawable/rounded_corners_button_blue" />

                        </LinearLayout>
```

```
                    </LinearLayout>

                </LinearLayout>

            </androidx.cardview.widget.CardView>

            <androidx.appcompat.widget.AppCompatButton
                android:id="@+id/checkout_paypal_btn"
                android:layout_width="match_parent"
                android:layout_height="40dp"
                android:text="PayPal"
                android:textSize="18sp"
                android:textColor="@color/white"
                android:textStyle="italic|bold"
                android:layout_gravity="center_horizontal"
                android:layout_marginTop="15dp"
                android:layout_marginBottom="10dp"
                android:background="@drawable/rounded_corners_button_blue" />


        </LinearLayout>

    </androidx.core.widget.NestedScrollView>


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

        <Button
            android:id="@+id/checkout_cancel_btn"
            android:layout_weight=".5"
            android:layout_width="0dip"
            android:layout_height="match_parent"
            android:text="@string/cancel"
            android:textAllCaps="true"
            android:textSize="16sp"
            android:textColor="@color/white"
            android:textAlignment="center"
            android:background="@color/colorAccent" />

        <Button
            android:id="@+id/checkout_order_btn"
            android:layout_weight=".5"
            android:layout_width="0dip"
            android:layout_height="match_parent"
            android:text="@string/btn_continue"
            android:textAllCaps="true"
            android:textSize="16sp"
            android:textColor="@color/white"
            android:textAlignment="center"
            android:background="@color/colorAccentGrey" />

    </LinearLayout>

</LinearLayout>
```

Ordertracking.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:tools="http://schemas.android.com/tools"
    tools:context="com.amanakash.ecommerce.activities.Order_Tracking_Activity">

    <fragment
        android:id="@+id/order_tracking_map"
        android:name="com.google.android.gms.maps.SupportMapFragment"
        android:layout_width="match_parent"
        android:layout_height="match_parent"/>

</RelativeLayout>
```

About.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.core.widget.NestedScrollView
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="10dp"
    android:background="@color/windowBackground" >

    <LinearLayout
        xmlns:android="http://schemas.android.com/apk/res/android"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical"
        android:layout_margin="10dp"
        android:gravity="center_horizontal">

        <ImageView
            android:layout_width="80dp"
            android:layout_height="80dp"
            android:layout_marginTop="5dp"
            android:src="@drawable/ic_logo" />

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="E-COMMERCE"
            android:textSize="15sp"
            android:layout_marginTop="10dp"
            android:textColor="@color/textColorSecondaryDark" />

        <WebView
            android:id="@+id/about_us_webView"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:padding="5dp"
            android:layout_marginTop="15dp" />
```

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:elevation="1dp"
    android:orientation="vertical"
    android:layout_marginTop="20dp"
    android:layout_marginBottom="15dp"
    android:gravity="center_horizontal"
    android:background="@color/dividerLight" >

    <TextView
        android:id="@+id/official_web"
        android:layout_width="match_parent"
        android:layout_height="45dp"
        android:text="@string/official_web"
        android:textSize="16sp"
        android:textColor="@color/textColorSecondary"
        android:clickable="true"
        android:paddingEnd="5dp"
        android:paddingStart="10dp"
        android:gravity="center_vertical"
        android:background="@color/white"
        android:drawableEnd="@drawable/ic_chevron_down" />
    <TextView
        android:id="@+id/official_whatsapp"
        android:layout_width="match_parent"
        android:layout_height="45dp"
        android:text="@string/official_whatsapp"
        android:textSize="16sp"
        android:textColor="@color/textColorSecondary"
        android:clickable="true"
        android:paddingEnd="5dp"
        android:paddingStart="10dp"
        android:layout_marginTop="1dp"
        android:gravity="center_vertical"
        android:background="@color/white"
        android:drawableEnd="@drawable/ic_chevron_down" />
    <TextView
        android:id="@+id/privacy_policy"
        android:layout_width="match_parent"
        android:layout_height="45dp"
        android:text="@string/privacy_policy"
        android:textSize="16sp"
        android:textColor="@color/textColorSecondary"
        android:clickable="true"
        android:paddingEnd="5dp"
        android:paddingStart="10dp"
        android:layout_marginTop="1dp"
        android:gravity="center_vertical"
        android:background="@color/white"
        android:drawableEnd="@drawable/ic_chevron_down" />
    <TextView
        android:id="@+id/refund_policy"
        android:layout_width="match_parent"
        android:layout_height="45dp"
        android:text="@string/refund_policy"
        android:textSize="16sp"
```

```
                android:textColor="@color/textColorSecondary"
                android:clickable="true"
                android:paddingEnd="5dp"
                android:paddingStart="10dp"
                android:layout_marginTop="1dp"
                android:gravity="center_vertical"
                android:background="@color/white"
                android:drawableEnd="@drawable/ic_chevron_down" />
            <TextView
                android:id="@+id/service_terms"
                android:layout_width="match_parent"
                android:layout_height="45dp"
                android:text="@string/service_terms"
                android:textSize="16sp"
                android:textColor="@color/textColorSecondary"
                android:clickable="true"
                android:paddingEnd="5dp"
                android:paddingStart="10dp"
                android:layout_marginTop="1dp"
                android:gravity="center_vertical"
                android:background="@color/white"
                android:drawableEnd="@drawable/ic_chevron_down" />

            <TextView
                android:id="@+id/a_z_terms"
                android:layout_width="match_parent"
                android:layout_height="45dp"
                android:text="@string/a_z"
                android:textSize="16sp"
                android:textColor="@color/textColorSecondary"
                android:clickable="true"
                android:paddingEnd="5dp"
                android:paddingStart="10dp"
                android:layout_marginTop="1dp"
                android:gravity="center_vertical"
                android:background="@color/white"
                android:drawableEnd="@drawable/ic_chevron_down" />

        </LinearLayout>

    </LinearLayout>

</androidx.core.widget.NestedScrollView>
```

Itemcardstyle.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="4dp"
    android:background="@color/windowBackground"
    app:cardUseCompatPadding="true"
    app:elevation="0dp">

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="match_parent"
        android:layout_height="@dimen/lg_card_height"
        android:scaleType="fitCenter"
        android:src="@drawable/rcard1" />

    <ImageView
        android:id="@+id/checkMark"
        android:layout_width="match_parent"
        android:layout_height="@dimen/lg_card_height"
        android:alpha="0.5"
        android:background="@color/black"
        android:clickable="true"
        android:paddingBottom="40dp"
        android:scaleType="center"
        android:src="@drawable/checked"
        android:tint="@color/white" />

</RelativeLayout>
```

Order_details.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:card_view="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:clickable="true"
    android:focusable="true"
    android:background="@color/layoutBackgroundLight">


    <androidx.core.widget.NestedScrollView
        android:layout_width="match_parent"
        android:layout_height="0dip"
        android:layout_weight="1"
        android:padding="15dp">

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


            <androidx.cardview.widget.CardView
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:background="@color/white"
                app:cardElevation="0dp"
                app:cardCornerRadius="2dp"
                card_view:cardCornerRadius="2dp"
                card_view:cardUseCompatPadding="true"
                card_view:cardPreventCornerOverlap="false" >

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

```
android:orientation="vertical">

<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="@string/order_details"
    android:textSize="16sp"
    android:textColor="@color/textColorSecondaryDark"
    android:padding="5dp"
    android:background="@color/windowBackgroundLight" />

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:padding="5dp">

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center_vertical"
        android:orientation="horizontal">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:maxLines="1"
            android:text="@string/no_of_products"
            android:textColor="@color/textColorSecondary"
            android:textSize="14sp"/>

        <TextView
            android:id="@+id/order_products_count"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignParentEnd="true"
            android:gravity="end"
            android:maxLines="1"
            android:text="@string/no_of_products"
            android:textAlignment="textEnd"
            android:textColor="@color/textColorSecondary"
            android:textSize="14sp"/>

    </RelativeLayout>
    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:gravity="center_vertical"
        android:layout_marginTop="3dp" >

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:maxLines="1"
            android:text="@string/total_price"
            android:textSize="14sp"
            android:textColor="@color/textColorSecondary" />
        <TextView
            android:id="@+id/order_price"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:maxLines="1"
            android:text="$00.0"
            android:textSize="14sp"
            android:textColor="@color/textColorSecondary"
            android:textAlignment="textEnd"
            android:gravity="end"
            android:layout_alignParentEnd="true" />

    </RelativeLayout>
    <!--<RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center_vertical"
        android:orientation="horizontal"
        android:layout_marginTop="3dp">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:maxLines="1"
            android:text="@string/shipping_method"
            android:textColor="@color/textColorSecondary"
            android:textSize="14sp"/>

        <TextView
            android:id="@+id/shipping_method"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignParentEnd="true"
            android:gravity="end"
            android:maxLines="1"
            android:text="@string/shipping_method"
            android:textAlignment="textEnd"
            android:textColor="@color/textColorSecondary"
            android:textSize="14sp"/>

    </RelativeLayout>
    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="3dp"
        android:gravity="center_vertical"
        android:orientation="horizontal">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:maxLines="1"
            android:text="@string/payment_method"
            android:textColor="@color/textColorSecondary"
            android:textSize="14sp"/>

        <TextView
```

```
        android:id="@+id/payment_method"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:gravity="end"
        android:maxLines="1"
        android:text="@string/payment_method"
        android:textAlignment="textEnd"
        android:textColor="@color/textColorSecondary"
        android:textSize="14sp"/>

</RelativeLayout>-->


<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="3dp"
    android:gravity="center_vertical"
    android:orientation="horizontal">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:maxLines="1"
        android:text="@string/order_date"
        android:textColor="@color/textColorSecondary"
        android:textSize="14sp"/>

    <TextView
        android:id="@+id/order_date"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:gravity="end"
        android:maxLines="1"
        android:text="@string/order_date"
        android:textAlignment="textEnd"
        android:textColor="@color/textColorSecondary"
        android:textSize="14sp"/>

</RelativeLayout>

<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="3dp"
    android:gravity="center_vertical"
    android:orientation="horizontal">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:maxLines="1"
        android:id="@+id/text"
        android:text="@string/order_status"
        android:textColor="@color/textColorSecondary"
        android:textSize="14sp"/>
```

```
                            <TextView
                                android:id="@+id/order_status"
                                android:layout_width="wrap_content"
                                android:layout_height="wrap_content"
                                android:layout_alignParentEnd="true"
                                android:gravity="end"
                                android:maxLines="1"
                                android:textAlignment="textEnd"
                                android:textColor="@color/textColorSecondary"
                                android:textSize="14sp"
                                android:layout_alignParentRight="true" />

                            <com.kofigyan.stateprogressbar.StateProgressBar
                                android:id="@+id/stateProgress"
                                android:visibility="gone"
                                android:layout_width="wrap_content"
                                android:layout_height="wrap_content"
                                app:spb_animateToCurrentProgressState="true"
                                app:spb_checkStateCompleted="true"
                                app:spb_currentStateDescriptionColor="#808080"
                                app:spb_currentStateNumber="four"
                                app:spb_maxStateNumber="four"
                                app:spb_stateBackgroundColor="#BDBDBD"
                                app:spb_stateDescriptionColor="#808080"
                                app:spb_stateDescriptionSize="10sp"

app:spb_stateForegroundColor="@color/colorAccentGreen"
                                app:spb_stateNumberBackgroundColor="#808080"
                                app:spb_stateNumberForegroundColor="#eeeeee"
                                app:spb_stateSize="20dp"
                                android:layout_below="@+id/text"/>

                        </RelativeLayout>


                    </LinearLayout>

                </LinearLayout>

            </androidx.cardview.widget.CardView>


            <androidx.cardview.widget.CardView
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:background="@color/white"
                android:layout_marginTop="10dp"
                app:cardElevation="0dp"
                app:cardCornerRadius="2dp"
                card_view:cardCornerRadius="2dp"
                card_view:cardUseCompatPadding="true"
                card_view:cardPreventCornerOverlap="false" >

                <LinearLayout
                    android:layout_width="match_parent"
                    android:layout_height="match_parent"
                    android:orientation="vertical">
                    <TextView
                        android:layout_width="match_parent"
                        android:layout_height="wrap_content"
                        android:text="@string/billing_address"
                        android:textSize="16sp"
                        android:textColor="@color/textColorSecondaryDark"
                        android:padding="5dp"
                        android:background="@color/windowBackgroundLight" />

                    <LinearLayout
                        android:layout_width="match_parent"
                        android:layout_height="wrap_content"
                        android:orientation="vertical"
                        android:padding="5dp" >

                        <TextView
                            android:id="@+id/billing_name"
                            android:layout_width="match_parent"
                            android:layout_height="wrap_content"
                            android:maxLines="1"
                            android:text="@string/full_name"
                            android:textSize="14sp"
                            android:textAlignment="viewStart"
                            android:textColor="@color/textColorSecondary" />

                        <TextView
                            android:id="@+id/billing_address"
                            android:layout_width="match_parent"
                            android:layout_height="wrap_content"
                            android:maxLines="1"
                            android:text="@string/city"
                            android:textSize="14sp"
                            android:textAlignment="viewStart"
                            android:textColor="@color/textColorSecondary" />

                        <TextView
                            android:id="@+id/billing_street"
                            android:layout_width="match_parent"
                            android:layout_height="wrap_content"
                            android:maxLines="1"
                            android:text="@string/address"
                            android:textSize="14sp"
                            android:textAlignment="viewStart"
                            android:textColor="@color/textColorSecondary" />

                    </LinearLayout>

                </LinearLayout>

            </androidx.cardview.widget.CardView>


            <androidx.cardview.widget.CardView
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:background="@color/white"
                android:layout_marginTop="10dp"
```

```
app:cardElevation="0dp"
app:cardCornerRadius="2dp"
card_view:cardCornerRadius="2dp"
card_view:cardUseCompatPadding="true"
card_view:cardPreventCornerOverlap="false" >

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

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/shipping_address"
        android:textSize="16sp"
        android:textColor="@color/textColorSecondaryDark"
        android:padding="5dp"
        android:background="@color/windowBackgroundLight" />

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:padding="5dp" >

        <TextView
            android:id="@+id/shipping_name"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:maxLines="1"
            android:text="@string/full_name"
            android:textSize="14sp"
            android:textAlignment="viewStart"
            android:textColor="@color/textColorSecondary" />

        <TextView
            android:id="@+id/shipping_address"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:maxLines="1"
            android:text="@string/city"
            android:textSize="14sp"
            android:textAlignment="viewStart"
            android:textColor="@color/textColorSecondary" />

        <TextView
            android:id="@+id/shipping_street"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:maxLines="1"
            android:text="@string/address"
            android:textSize="14sp"
            android:textAlignment="viewStart"
            android:textColor="@color/textColorSecondary" />

    </LinearLayout>
    </LinearLayout>

</androidx.cardview.widget.CardView>


<androidx.cardview.widget.CardView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="@color/white"
    android:layout_marginTop="10dp"
    app:cardElevation="0dp"
    app:cardCornerRadius="2dp"
    card_view:cardCornerRadius="2dp"
    card_view:cardUseCompatPadding="true"
    card_view:cardPreventCornerOverlap="false" >

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

        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="@string/shipping_method"
            android:textSize="16sp"
            android:textColor="@color/textColorSecondaryDark"
            android:padding="5dp"
            android:background="@color/windowBackgroundLight" />

        <TextView
            android:id="@+id/shipping_method"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:maxLines="1"
            android:padding="5dp"
            android:text="@string/shipping_method"
            android:textSize="14sp"
            android:textAlignment="viewStart"
            android:textColor="@color/textColorSecondary" />


    </LinearLayout>

</androidx.cardview.widget.CardView>


<androidx.cardview.widget.CardView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="@color/white"
    android:layout_marginTop="10dp"
    app:cardElevation="0dp"
    app:cardCornerRadius="2dp"
    card_view:cardCornerRadius="2dp"
    card_view:cardUseCompatPadding="true"
    card_view:cardPreventCornerOverlap="false" >
```

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

                <TextView
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:text="@string/products"
                    android:textSize="16sp"
                    android:textColor="@color/textColorSecondaryDark"
                    android:padding="5dp"
                    android:background="@color/windowBackgroundLight" />

                <androidx.recyclerview.widget.RecyclerView
                    android:id="@+id/checkout_items_recycler"
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content" />

            </LinearLayout>

        </androidx.cardview.widget.CardView>


        <androidx.cardview.widget.CardView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:background="@color/white"
            android:layout_marginTop="10dp"
            app:cardElevation="0dp"
            app:cardCornerRadius="2dp"
            card_view:cardCornerRadius="2dp"
            card_view:cardUseCompatPadding="true"
            card_view:cardPreventCornerOverlap="false" >

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

                <TextView
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:text="@string/subtotal"
                    android:textSize="16sp"
                    android:textColor="@color/textColorSecondaryDark"
                    android:padding="5dp"
                    android:background="@color/windowBackgroundLight" />

                <LinearLayout
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:orientation="vertical"
                    android:padding="5dp" >

                    <RelativeLayout
                        android:layout_width="match_parent"
                        android:layout_height="wrap_content"
```

```
    android:orientation="horizontal"
    android:gravity="center_vertical" >

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:maxLines="1"
        android:text="@string/subtotal"
        android:textSize="14sp"
        android:textColor="@color/textColorSecondary" />

    <TextView
        android:id="@+id/checkout_subtotal"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:maxLines="1"
        android:text="$00.0"
        android:textSize="14sp"
        android:textColor="@color/textColorSecondary"
        android:textAlignment="textEnd"
        android:gravity="end"
        android:layout_alignParentEnd="true" />

</RelativeLayout>
<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:gravity="center_vertical"
    android:layout_marginTop="3dp" >

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:maxLines="1"
        android:text="@string/tax"
        android:textSize="14sp"
        android:textColor="@color/textColorSecondary" />

    <TextView
        android:id="@+id/checkout_tax"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:maxLines="1"
        android:text="$00.0"
        android:textSize="14sp"
        android:textColor="@color/textColorSecondary"
        android:textAlignment="textEnd"
        android:gravity="end"
        android:layout_alignParentEnd="true" />

</RelativeLayout>
<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:gravity="center_vertical"
    android:layout_marginTop="3dp" >
```

```
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:maxLines="1"
            android:text="@string/shipping"
            android:textSize="14sp"
            android:textColor="@color/textColorSecondary" />

        <TextView
            android:id="@+id/checkout_shipping"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:maxLines="1"
            android:text="$00.0"
            android:textSize="14sp"
            android:textColor="@color/textColorSecondary"
            android:textAlignment="textEnd"
            android:gravity="end"
            android:layout_alignParentEnd="true" />

    </RelativeLayout>
    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:gravity="center_vertical"
        android:layout_marginTop="3dp" >

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:maxLines="1"
            android:text="@string/discount"
            android:textSize="14sp"
            android:textColor="@color/textColorSecondary" />

        <TextView
            android:id="@+id/checkout_discount"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:maxLines="1"
            android:text="$00.0"
            android:textSize="14sp"
            android:textColor="@color/textColorSecondary"
            android:textAlignment="textEnd"
            android:gravity="end"
            android:layout_alignParentEnd="true" />

    </RelativeLayout>
    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:gravity="center_vertical"
        android:layout_marginTop="3dp" >

        <TextView
```

```
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:maxLines="1"
                    android:text="@string/total"
                    android:textSize="14sp"
                    android:textColor="@color/textColorSecondary" />

                <TextView
                    android:id="@+id/checkout_total"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:maxLines="1"
                    android:text="$000.0"
                    android:textSize="14sp"
                    android:textColor="@color/textColorSecondary"
                    android:textAlignment="textEnd"
                    android:gravity="end"
                    android:layout_alignParentEnd="true" />

            </RelativeLayout>

        </LinearLayout>

    </LinearLayout>

</androidx.cardview.widget.CardView>


<androidx.recyclerview.widget.RecyclerView
    android:id="@+id/checkout_coupons_recycler"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" />


<androidx.cardview.widget.CardView
    android:id="@+id/buyer_comments_card"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="@color/white"
    android:layout_marginTop="10dp"
    app:cardElevation="0dp"
    app:cardCornerRadius="2dp"
    card_view:cardCornerRadius="2dp"
    card_view:cardUseCompatPadding="true"
    card_view:cardPreventCornerOverlap="false" >

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

        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="@string/order_notes"
            android:textSize="16sp"
            android:textColor="@color/textColorSecondaryDark"
            android:padding="5dp"
```

```
                android:background="@color/windowBackgroundLight" />

            <TextView
                android:id="@+id/buyer_comments"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:minLines="1"
                android:padding="5dp"
                android:text="@string/order_notes"
                android:textSize="14sp"
                android:textAlignment="viewStart"
                android:textColor="@color/textColorSecondary" />


        </LinearLayout>

    </androidx.cardview.widget.CardView>


    <androidx.cardview.widget.CardView
        android:id="@+id/seller_comments_card"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="@color/white"
        android:layout_marginTop="10dp"
        app:cardElevation="0dp"
        app:cardCornerRadius="2dp"
        card_view:cardCornerRadius="2dp"
        card_view:cardUseCompatPadding="true"
        card_view:cardPreventCornerOverlap="false" >

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

            <TextView
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:text="@string/order_notes"
                android:textSize="16sp"
                android:textColor="@color/textColorSecondaryDark"
                android:padding="5dp"
                android:background="@color/windowBackgroundLight" />

            <TextView
                android:id="@+id/seller_comments"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:minLines="1"
                android:padding="5dp"
                android:text="@string/order_notes"
                android:textSize="14sp"
                android:textAlignment="viewStart"
                android:textColor="@color/textColorSecondary" />


        </LinearLayout>
        </androidx.cardview.widget.CardView>


        <androidx.cardview.widget.CardView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:background="@color/white"
            android:layout_marginTop="10dp"
            app:cardElevation="0dp"
            app:cardCornerRadius="2dp"
            card_view:cardCornerRadius="2dp"
            card_view:cardUseCompatPadding="true"
            card_view:cardPreventCornerOverlap="false" >

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

                <TextView
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:text="@string/payment_method"
                    android:textSize="16sp"
                    android:textColor="@color/textColorSecondaryDark"
                    android:padding="5dp"
                    android:background="@color/windowBackgroundLight" />

                <TextView
                    android:id="@+id/payment_method"
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:maxLines="1"
                    android:padding="5dp"
                    android:text="@string/payment_method"
                    android:textSize="14sp"
                    android:textAlignment="viewStart"
                    android:textColor="@color/textColorSecondary" />


            </LinearLayout>

        </androidx.cardview.widget.CardView>


    </LinearLayout>

</androidx.core.widget.NestedScrollView>

<Button
    android:id="@+id/track_order_btn"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="@string/track_order"
    android:textAllCaps="true"
    android:textSize="16sp"
    android:textColor="@color/white"
```

```
        android:background="@color/colorAccent" />

</LinearLayout>
```

ShippingMethod.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/layoutBackgroundLight"
    android:clickable="true"
    android:focusable="true"
    android:gravity="center_horizontal"
    android:orientation="vertical">


    <androidx.core.widget.NestedScrollView
        android:id="@+id/scroll_container"
        android:layout_width="match_parent"
        android:layout_height="0dip"
        android:layout_weight="1">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical"
            android:padding="15dp">


            <LinearLayout
                android:id="@+id/ups_shipping_layout"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_marginBottom="15dp"
                android:orientation="vertical">

                <TextView
                    android:id="@+id/ups_shipping_title"
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:background="@color/windowBackgroundLight"
                    android:padding="5dp"
                    android:text="@string/shipping_method"
                    android:textColor="@color/textColorSecondaryDark"
                    android:textSize="16sp" />

                <androidx.recyclerview.widget.RecyclerView
                    android:id="@+id/ups_shipping_services_list"
                    android:layout_width="match_parent"
                    android:layout_height="match_parent" />

            </LinearLayout>

            <LinearLayout
                android:id="@+id/free_shipping_layout"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:orientation="vertical">

                <TextView
                    android:id="@+id/free_shipping_title"
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:background="@color/windowBackgroundLight"
                    android:padding="5dp"
                    android:text="@string/shipping_method"
                    android:textColor="@color/textColorSecondaryDark"
                    android:textSize="16sp" />

                <androidx.recyclerview.widget.RecyclerView
                    android:id="@+id/free_shipping_services_list"
                    android:layout_width="match_parent"
                    android:layout_height="match_parent" />

            </LinearLayout>

            <LinearLayout
                android:id="@+id/local_pickup_layout"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_marginTop="15dp"
                android:orientation="vertical">

                <TextView
                    android:id="@+id/local_pickup_title"
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:background="@color/windowBackgroundLight"
                    android:padding="5dp"
                    android:text="@string/shipping_method"
                    android:textColor="@color/textColorSecondaryDark"
                    android:textSize="16sp" />

                <androidx.recyclerview.widget.RecyclerView
                    android:id="@+id/local_pickup_services_list"
                    android:layout_width="match_parent"
                    android:layout_height="match_parent" />

            </LinearLayout>

            <LinearLayout
                android:id="@+id/flat_rate_layout"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_marginTop="15dp"
                android:orientation="vertical">

                <TextView
                    android:id="@+id/flat_rate_title"
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:background="@color/windowBackgroundLight"
                    android:padding="5dp"
                    android:text="@string/shipping_method"
                    android:textColor="@color/textColorSecondaryDark"
```

```
                    android:textSize="16sp" />

                <androidx.recyclerview.widget.RecyclerView
                    android:id="@+id/flat_rate_services_list"
                    android:layout_width="match_parent"
                    android:layout_height="match_parent" />

            </LinearLayout>

            <LinearLayout
                android:id="@+id/shipping_price_layout"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_marginTop="15dp"
                android:orientation="vertical">

                <TextView
                    android:id="@+id/shipping_price_title"
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:background="@color/windowBackgroundLight"
                    android:padding="5dp"
                    android:text="@string/shipping_method"
                    android:textColor="@color/textColorSecondaryDark"
                    android:textSize="16sp" />

                <androidx.recyclerview.widget.RecyclerView
                    android:id="@+id/shipping_price_list"
                    android:layout_width="match_parent"
                    android:layout_height="match_parent" />

            </LinearLayout>


        </LinearLayout>

    </androidx.core.widget.NestedScrollView>


    <androidx.appcompat.widget.AppCompatButton
        android:id="@+id/save_shipping_method_btn"
        android:layout_width="match_parent"
        android:layout_height="@dimen/button_height"
        android:background="@color/colorAccent"
        android:text="@string/next"
        android:textColor="@color/white"
        android:textSize="@dimen/button_text_size"/>


</LinearLayout>
```

Splashscreen.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/mainView"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center"
    android:orientation="vertical">

    <ImageView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:alpha=".7"
        android:scaleType="centerCrop"
        android:src="@drawable/splashbg" />

    <ProgressBar
        android:id="@+id/progressBar"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_alignParentBottom="true"
        android:layout_marginBottom="160dp"
        android:indeterminateTint="@color/white" />

</RelativeLayout>
```

7.4. Conclusion

The project entitled E-Commerce shopping Application was completed successfully. The system has been developed with much care and free of errors and at the same time it is efficient and less time consuming. The purpose of this project was to develop a web application CRM and an android application for purchasing items from a shop. This project helped us in gaining valuable information and practical knowledge on several topics like designing web pages using html & CSS, usage of responsive templates, designing of android applications, and management of database using MySQL. The entire system is secured. Also, the project helped us understanding about the development phases of a project and software development life cycle. We learned how to test different features of a project.

7.5. References

For Android:

https://developer.android.com/docs

For CRM:

https://laravel.com/docs/7.x

https://laravel.com/docs/5.1/quickstart

https://www.w3schools.com/js/DEFAULT.asp

https://www.linkedin.com/learning/

https://www.w3schools.com/html/

https://www.w3schools.com/css/

https://www.w3schools.com/bootstrap/

https://www.codecademy.com/learn/learn-bootstrap

For MySQL:

https://www.mysql.com/

http://www.mysqltutorial.org

For WAMP:

https://www.wampserver.com/download-wampserver-64bits/