A PROJECT REPORT

ON

"PRODUCT RECOMMENDATION SYSTEM IN WAREHOUSE MANAGEMENT"

Submitted to

UNIVERSITY OF MUMBAI

In Partial Fulfilment of the Requirement for the Award of

BACHELOR'S DEGREE IN
COMPUTER ENGINEERING

BY

Memon Mohd Sahil Sajid Afroz	17CO35
Bebal Altamash Abdul Hamid Sadika	17CO15
Sayyed Alfina Mohhammed Aziz	17CO10
Kokate Pranali Prakash Jyoti	17CO04

UNDER THE GUIDANCE OF
Prof. Kalpana R. Bodke

DEPARTMENT OF COMPUTER ENGINEERING
Anjuman-I-Islam's Kalsekar Technical Campus
SCHOOL OF ENGINEERING & TECHNOLOGY
Plot No. 2 3, Sector - 16, Near Thana Naka,
Khandagaon, New Panvel - 410206

2020-2021

AFFILIATED TO

UNIVERSITY OF MUMBAI

A PROJECT II REPORT
ON

”PRODUCT RECOMMENDATION SYSTEM IN WAREHOUSE MANAGEMENT”

Submitted to
UNIVERSITY OF MUMBAI

In Partial Fulfilment of the Requirement for the Award of

BACHELOR’S DEGREE IN COMPUTER ENGINEERING

BY

Memon Mohd Sahil Sajid Afroz	17CO35
Bebal Altamash Abdul Hamid Sadika	17CO15
Sayyed Alfina Mohhammed Aziz	17CO10
Kokate Pranali Prakash Jyoti	17CO04

UNDER THE GUIDANCE OF
Prof. Kalpana R. Bodke

DEPARTMENT OF COMPUTER ENGINEERING
Anjuman-I-Islam’s Kalsekar Technical Campus
SCHOOL OF ENGINEERING & TECHNOLOGY
Plot No. 2 3, Sector - 16, Near Thana Naka,
Khandagaon, New Panvel - 410206

2020-2021
AFFILIATED TO

UNIVERSITY OF MUMBAI

Anjuman-i-Islam's Kalsekar Technical Campus

Department of Computer Engineering

SCHOOL OF ENGINEERING & TECHNOLOGY

Plot No. 2 3, Sector - 16, Near Thana Naka,

Khandagaon, New Panvel - 410206

CERTIFICATE

This is certify that the project entitled

"Product Recommendation System In Warehouse Management"

submitted by

Memon Mohd Sahil Sajid Afroz	**17CO35**
Bebal Altamash Abdul Hamid Sadika	**17CO15**
Sayyed Alfina Mohammed Aziz	**17CO10**
Kokate Pranali Prakash Jyoti	**17CO04**

is a record of bonafide work carried out by them, in the partial fulfilment of the requirement for the award of Degree of Bachelor of Engineering (Computer Engineering) at ***Anjuman-I-Islam's Kalsekar Technical Campus, Navi Mumbai*** under the University of MUMBAI. This work is done during year 2020-2021, under our guidance.

Date: / /

(Prof. Kalpana R. Bodke)
Project Supervisor

(Prof. Kalpana R. Bodke)
Project Coordinator

(Prof. Tabrez Khan)
HOD, Computer Department

DR. ABDUL RAZAK HONNUTAGI
Director

External Examiner

Acknowledgements

I would like to take the opportunity to express my sincere thanks to my guide **Prof. Kalpana R. Bodke**, Assistant Professor, Department of Computer Engineering, AIKTC, School of Engineering, Panvel for her invaluable support and guidance throughout my project research work. Without her kind guidance & support this was not possible.

I am grateful to her for her timely feedback which helped me track and schedule the process effectively. Her time, ideas and encouragement that she gave is help me to complete my project efficiently.

We would like to express deepest appreciation towards **DR. ABDUL RAZAK HONNUTAGI**, Director, AIKTC, Navi Mumbai, **Prof. Tabrez Khan**, Head of Department of Computer Engineering and **Prof. Kalpana R. Bodke**, Project Coordinator whose invaluable guidance supported us in completing this project.

At last we must express our sincere heartfelt gratitude to all the staff members of Computer Engineering Department who helped me directly or indirectly during this course of work.

Memon Mohd Sahil Sajid Afroz
Bebal Altamash Abdul Hamid Sadika
Sayyed Alfina Mohhammed Aziz
Kokate Pranali Prakash Jyoti

Project I Approval for Bachelor of Engineering

This project entitled *"Product Recommendation System In Warehouse Management"* by ***Memon Mohd Sahil Sajid Afroz, Bebal Altamash Abdul Hamid Sadika, Sayyed Alfina Mohhammed, Aziz, Kokate Pranali Prakash Jyoti*** is approved for the degree of ***Bachelor of Engineering in Department of Computer Engineering***.

Examiners

1. ....
2. ....

Supervisors

1. ....
2. ....

Chairman

....

Declaration

I declare that this written submission represents my ideas in my own words and where others ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Memon Mohd Sahil
17CO35

Bebal Altamash
17CO15

Sayyed Alfina
17CO10

Kokate Pranali
17CO04

ABSTRACT

In the current pandemic situation of COVID-19, people are afraid to leave their homes even to get the basic day-to-day life essentials like groceries,vegetables,fruits etc. Everyone wants to maintain social distance, but in some cases even if they want to they can't. People wants to make less and less interaction with each other for their safety purpose. But they have to leave their homes to get their daily supplements. We never know which seller or vendor is infected.

The online shopping sites like amazon, flipkart, bigbasket etc. are providing groceries at home but the disadvantage is that they have limited slots and most of the time they are not available. So if one person wants to cook for today but will get vegetables and groceries after 2 days if he/she shops from these online sites, which makes no sense.We want customers shopping experience for groceries and daily essentials to be hassle free and comfortable.

The solution to this problem is an android application and website through which user will be able to shop for daily essentials which will be available to user with minimum time. The main aim of the proposed system is to provide user an easy and safe shopping interface where he/she will shop without any worries of social distancing and other health risks and will get his/her purchase with least time delay. Also, the admin can easily manage the warehouse with help of the admin panel developed.

Keywords: Android, COVID-19, Social distance, Groceries, Recommendation,warehouse management, website, online shopping,grocery apps.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

When covid-19 first begun to spread rapidly in India, people got scared to go out of their homes to buy daily needs in the fear of infection. Since it spreads mainly through contact with an infected person or when a person touches a surface that has the virus on it, the best way to be safe against it is to stay at home and follow precautionary measures. In India majority of population is used to buying groceries, fruits , vegetables etc. from local markets or stores. But due to pandemic situation, it has been difficult to leave homes to go out and buy the necessary things like vegies,fruits etc. Which are day to day life essentials.This has increased online shopping usage globally. It has led to rise in the first time e-commerce users.

1.1 Purpose

As the pandemic situation was increasing, more and more people were attracting towards online shopping since it hold less human interaction and less risk of getting infection. BigBasket which is one of the key online grocery player in India had following message “We’ll be back soon! We are currently experiencing unprecedented demand. In light of this we are restricting access to our website to existing customers only. Please try again in a few hours” displayed on their website.

Also they kept limited time slots for users to shop. Grofers which is another online grocery shopping site had a similar kind of message which said “Due to sudden rush, we have stopped servicing many locations, but are working to increase capacity and will be resuming operations shortly”. Amazon which is one of the most leading ecommerce player has announced that “Customers are relying on them like never before in their social distancing and self-quarantine efforts”. Hence Amazon is temporarily going to stop taking orders for lower priority products. Though people were moving towards online grocery shopping, they were facing time delays, unavailability of products, limited time slots and order rejections and many more hassles.

So to overcome all above difficulties and to provide user an easy and safe shopping experience along with recommendation we have designed a system through which user gets a comfortable and easy shopping experience with safety. The purpose of the system is to analyze and understand the customers requirements , to provide customers requirements with less delay and great safety.

1.2 Project Scope

This project is proposed to help people in the pandemic situation for safe grocery shopping and recommending products to them to make their shopping experience more safe and easy. The proposed system will users to fulfil their daily requirements like groceries, fruits, vegetables etc. Without worrying about social distancing and human interaction. With the proposed system user can avoid more human interaction, can save time and at the same time be safe.

1.3 Project Goals and Objectives

1.3.1 Goals

a. To provide better user interface.

b. To provide flexible system to user.

c. To provide more functionalities to user.

d. Target Large Audience and provide Benefits to them.

e. User can use it from anywhere through mobile or computer.

1.3.2 Objectives

a. To make shopping easier and comfortable.

b. To make process reasonably logical .

c. To serve customers without wasting their precious time.

d. To make people's daily needs for eatables available with ease.

e. To reach products to the customer's address with great care.

f. To promote our own country apps.

g. To make whole process independent.

h. To provide recommendation to user.

i. To provide user an application which will be easy to use and will satisfy user's requirements.

j. To provide an interface for admin to manage the warehouse as well as the customers.

1.4 Organization of Report

The report is organized as follows : The introduction is given in Chapter 1.It describes the fundamental terms used in this project.It describes the Goal,Objectives and scope of this project. The Chapter 2 describes the review of the relevant various techniques in the literature systems. It describes the pros and cons of each technique with how to overcome those cons using new technology.

The project planning includes members and capabilities of this project ,roles and responsibilities of each member,Budget of Project and Project timeline is describe in Chapter 3. The Chapter 4 describes Functional and Nonfunctional Requirements of project.Along with this it also explain features of system and constraints of system.

The Chapter 5 includes Design Information with Class Diagram, Sequence Diagram , Component Diagram and System Architecture. Implementation of each module is explained in Chapter 6. Chapter 7 shows final Test Cases and Test Results. Chapter 8 includes Screenshot of outputs and Conclusion and Future Scope of Project is described in Chapter 9.

Chapter 2

Literature Survey

2.1 Grocery Shopping Preferences during the COVID-19 Pandemic

A dynamic relationship of COVID-19 pandemic to the behavior of grocery shopper is observed. Variability in the behavior of grocery shoppers under various scenarios of the COVID-19 pandemic is concluded. If we consider the temporary closure of many food-away-from-home establishments and sources, expenditure of consumers on groceries during the COVID-19 pandemic has increased in great level. Grocery shopping is an essential activity in day to day life cycle, but there is not much known about the dynamic relationship of COVID-19 pandemic to the behavior of grocery shoppers. A survey was conducted in preference with purchasing methods, time windows, minimum order requirements, and fees to find variability in the behavior of grocery shoppers under various scenarios of the COVID-19 pandemic. As the pandemic is spreading rapidly, in this situation people are not ready to buy from local grocery stores.

2.1.1 Advantages of Paper

a. Most of the people prefers online grocery shopping methods.

b. Consumers also have relatively strong preferences for the time slot attribute.

2.1.2 Disadvantages of Paper

a. Most of the online services provided time windows for buying the products and the access was given to only the existing users.

b. There were paid memberships.

c. minimum order requirement constraints.

2.1.3 How to overcome the problems mentioned in Paper

a. No time slot method. User can shop according to his/her available time.

b. Minimum order requirement is not a compulsion for users to buy since we are developing the system so that user can shop at any time for the needed requirements.

c. It does not have the constraints of time and minimum order

2.2 Consumer behavior-Online grocery shopping in India: An overview

The present study in this paper deals with behavior of the consumers with respect to the consumption pattern of grocery products purchased online in India. It gives an insight into the drifting online consumption trends and also analyses the online consumer behavior in India with the advent of online grocers enabling the consumers to lessen their consumption pattern from the traditional shopping mode to trendy and modern, click and mortar consumption mode. Various online digital platforms such as bigbasket, amazon groceries, reliance fresh, zopnow, bazaar cart etc. have emerged in the grocery sectors to change consumption pattern of households in the present digital scenario of digital world. This paper here gives us an overview of these digital portals. Overall, the study in this paper covers the different dimensions of online consumption pattern of consumers and their behavior with the increased access to internet and more systematised and sophisticated approaches.

2.2.1 Advantages of Paper

a. The emergence of first generation start-ups and already established traditional grocery chains expanding to the digital platform in digital e-commerce world.

b. Growth of e-commerce in the Indian market

2.2.2 Disadvantages of Paper

a. Security of data

b. cost to develop digital platforms for grocery shopping are disadvantages.

2.2.3 How to overcome the problems mentioned in Paper

a. For data storage we are using Firebase which provides a full set of tools for managing security of application.

b. We are developing the application using android in which development is not expensive and reasonable so cost can be resolved

2.3 An Exploratory Study on Consumer Attitude Towards Online Grocery shopping

The research in this paper is aimed to carry out a study to identify the factors that are influencing the consumers to go for online grocery shopping and their attitude towards it. Nowadays, buyer's market is rapidly changing and hence identifying the needs and wants of the customers and understanding their attitude is now become challenging task. Shopping for groceries online, is a way of buying food, vegies, fruits and other necessities using a web-based shopping service. A consumer can order for groceries from various online platforms. When it comes to online grocery shopping, due to consumers' busy work schedule, the innovative shoppers or early adopters are finding ways for the changing technology to help them in newer ways of shopping.

The emerging online grocery shopping is being increasingly adopted by many consumers' in urban areas. There are many different factors which encourages people to shop for groceries on online platforms, but it is not known what factors influence them to go for online buying of groceries. The paper concludes the factors which makes people turn towards online grocery shopping and what features they expect in those online platforms.

2.3.1 Advantages of Paper

a. It is easy to browse the information through online rather than the traditional retail shopping.

b. Browse or search an online catalogue can save time and patience

2.3.2 Disadvantages of Paper

a. The e-grocers must provide clear and very detailed information about products so as to facilitate consumers with the ease and feeling of getting the product that they are looking for.

b. Particular website/application does not function smoothly

c. Time delays

2.3.3 How to overcome the problems mentioned in Paper

a. We are providing details of the product in a very detailed manner like its availability, Market prices, discounts, benefits of the products etc.

b. Also we have categorized the products and can be easily browsed/searched.

c. Both android application and website are developed with secured libraries and security so that they work smoothly even with the traffic.

2.4 Technical Review

Android is one of the most used technology in the digital world. Everyone is having a mobile phone or cellphone which consist of loads of application based on android. Most of the phones are android based and thus android applications are the must. All day to day activities can be done using android application.One of the most widely used mobile OS these days is ANDROID. Android is a software bunch comprising not only operating system but also middle-ware and key applications. People can shop,order, exchange, track their orders ,products, essentials etc. By just making use of an android application. Thus to target a large number of users, android application is the best option for the project.

2.4.1 Advantages of Technology

a. As an open-source platform, Android's Software Development Kit (SDK) is readily available to developers. Further, Android app development is cost-effective

b. The entire development cycle is not much pricey

c. Easy Customization

d. The biggest strength of the Android platform is its global presence

2.4.2 Reasons to use this Technology

a. Scope for Innovation: In a way, Android offers a wide scope for innovation and opens the doors to new business opportunities.

b. Evolving Platform: They keep on bringing new features to stay firm amid growing competition, and the Android developers' community quickly gets adapted to them. When we opt for an Android app, you also get the advantage of this evolving platform.

c. Easy to Integrate: Android is the best mobile platform between the application and processes architecture. Most of the platforms allow background processes helping us to integrate the apps

Chapter 3

Project Planning

3.1 Members and Capabilities

Table 3.1: Table of Capabilities

SR. No	Name of Member	Capabilities
1	Mohd Sahil Memon	Frontend, UI
2	Altamash Bebal	Django, Backend
3	Alfina Sayyed	Backend
4	Pranali Kokate	Frontend

Work Breakdown Structure

a. All of the members are equally important in developing the project.

b. We work on a different part of the project based on one's capability.

c. Firstly we came up with documentation, And based on the documentation we set our goal and created a blueprint.

d. We then started going hands-on with the project to develop it according to the flow as decided earlier.

3.2 Roles and Responsibilities

Table 3.2: Table of Responsibilities

SR. No	Name of Member	Role	Responsibilities
1	Mohd Sahil Memon	Team Leader	Frontend , UI Design
2	Altamash Bebal	Backend	Django, Integrating System
3	Alfina Sayyed	Backend	Backend, Embedding System
4	Pranali Kokate	Frontend	Frontend,UI

3.3 Assumptions and Constraints

a. User of the app Should know how to use browser and Internet.

b. User of the app should know how to purchase products online.

3.4 Project Management Approach

a. Planning of project.

b. Defining the scope of the project.

c. Estimation of time and It's management.

d. Creating Gant Charts and properly assigning tasks to members.

e. Reporting the progress of project with the guide.

3.5 Ground Rules for the Project

a. Properly planning and gathering relevant information is very important.

b. Developing a Blueprint of the project and work accordingly.

c. All the members should report to the guide whenever required

d. Setting up small goals every week.

e. Achieving the small goal within that span of time.

f. Keeping tracks of the progress towards project.

g. Participate in meeting.

h. Inform the leader about unavailability.

3.6 Project Budget

a. It is a sponsored project. The budget is decided by the client.

b. Cost of the project is minimal and efficient.

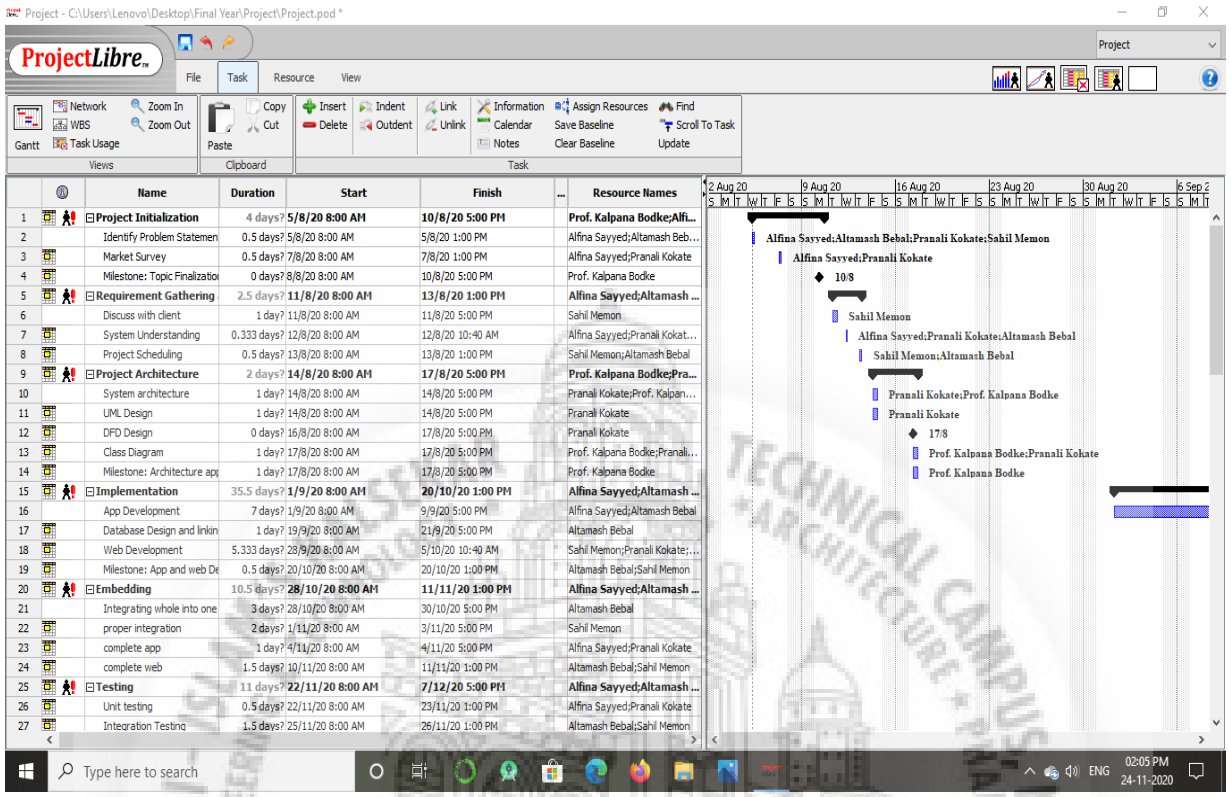
c. The cost of Cloud Would add in future scope of project.

d. Python Programming Language(Open Source)

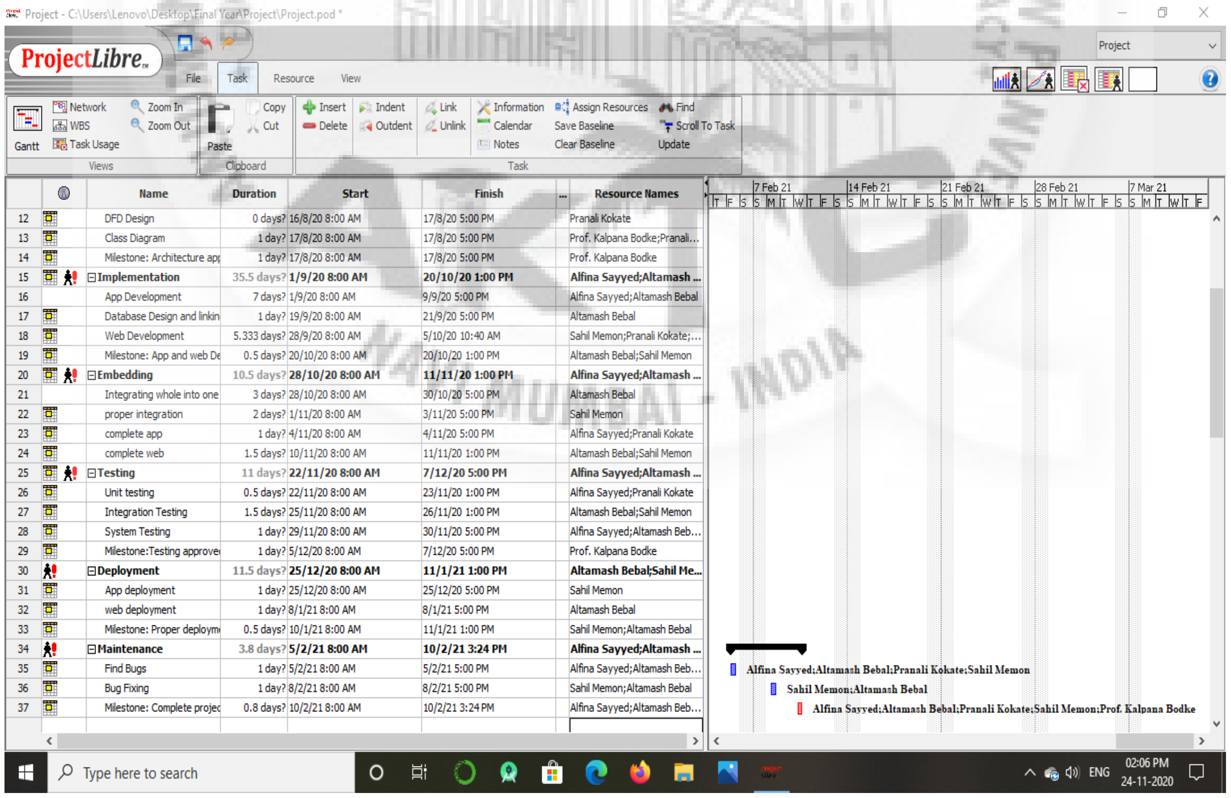
e. Firebase : Open Source

f. Frame Work : Django(Open Source)

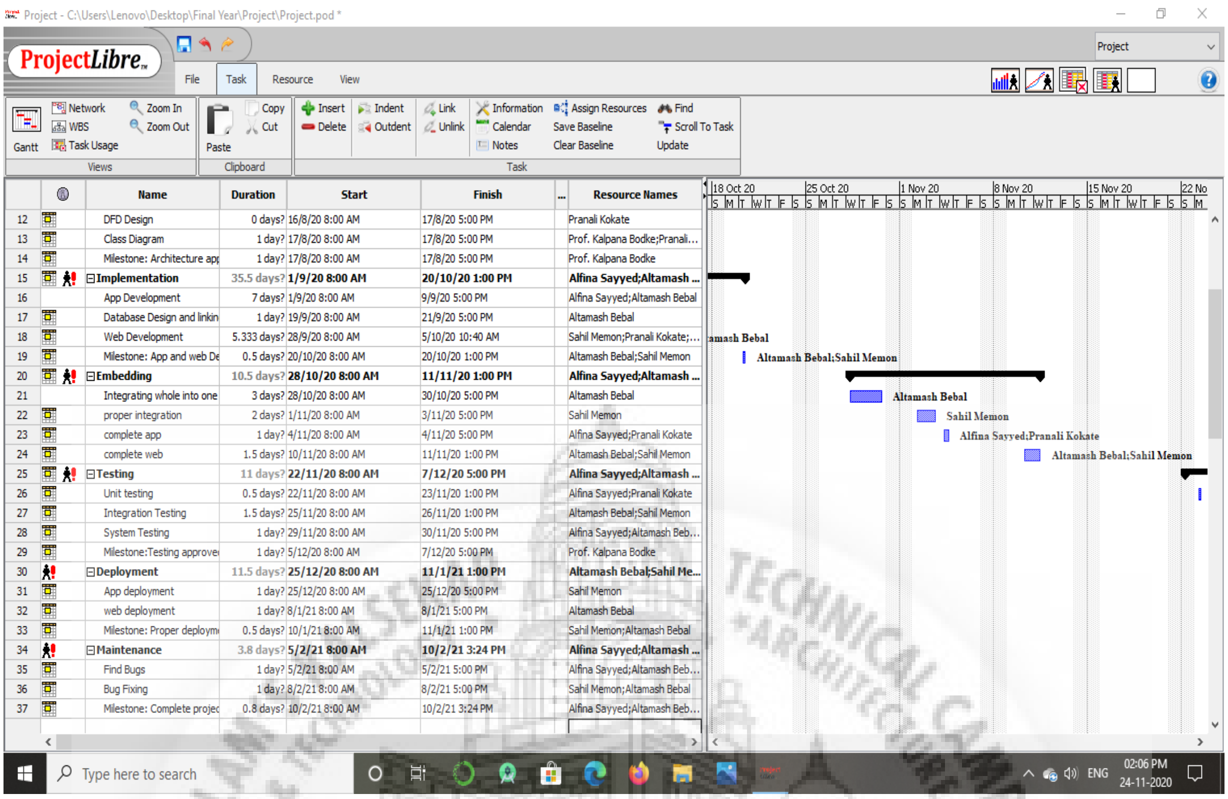
3.7 Project Timeline



	Name	Duration	Start	Finish	...	Resource Names
1	⊟**Project Initialization**	4 days?	**5/8/20 8:00 AM**	**10/8/20 5:00 PM**		**Prof. Kalpana Bodke;Alfi...**
2	Identify Problem Statemen	0.5 days?	5/8/20 8:00 AM	5/8/20 1:00 PM		Alfina Sayyed;Altamash Beb...
3	Market Survey	0.5 days?	7/8/20 8:00 AM	7/8/20 1:00 PM		Alfina Sayyed;Pranali Kokate
4	Milestone: Topic Finalizatio	0 days?	8/8/20 8:00 AM	10/8/20 5:00 PM		Prof. Kalpana Bodke
5	⊟**Requirement Gathering**	2.5 days?	**11/8/20 8:00 AM**	**13/8/20 1:00 PM**		**Alfina Sayyed;Altamash ...**
6	Discuss with client	1 day?	11/8/20 8:00 AM	11/8/20 5:00 PM		Sahil Memon
7	System Understanding	0.333 days?	12/8/20 8:00 AM	12/8/20 10:40 AM		Alfina Sayyed;Pranali Kokat...
8	Project Scheduling	0.5 days?	13/8/20 8:00 AM	13/8/20 1:00 PM		Sahil Memon;Altamash Bebal
9	⊟**Project Architecture**	2 days?	**14/8/20 8:00 AM**	**17/8/20 5:00 PM**		**Prof. Kalpana Bodke;Pra...**
10	System architecture	1 day?	14/8/20 8:00 AM	14/8/20 5:00 PM		Pranali Kokate;Prof. Kalpan...
11	UML Design	1 day?	14/8/20 8:00 AM	14/8/20 5:00 PM		Pranali Kokate
12	DFD Design	0 days?	16/8/20 8:00 AM	17/8/20 5:00 PM		Pranali Kokate
13	Class Diagram	1 day?	17/8/20 8:00 AM	17/8/20 5:00 PM		Prof. Kalpana Bodke;Pranali...
14	Milestone: Architecture app	1 day?	17/8/20 8:00 AM	17/8/20 5:00 PM		Prof. Kalpana Bodke
15	⊟**Implementation**	35.5 days?	**1/9/20 8:00 AM**	**20/10/20 1:00 PM**		**Alfina Sayyed;Altamash ...**
16	App Development	7 days?	1/9/20 8:00 AM	9/9/20 5:00 PM		Alfina Sayyed;Altamash Bebal
17	Database Design and linkin	1 day?	19/9/20 8:00 AM	21/9/20 5:00 PM		Altamash Bebal
18	Web Development	5.333 days?	28/9/20 8:00 AM	5/10/20 10:40 AM		Sahil Memon;Pranali Kokate;...
19	Milestone: App and web De	0.5 days?	20/10/20 8:00 AM	20/10/20 1:00 PM		Altamash Bebal;Sahil Memon
20	⊟**Embedding**	10.5 days?	**28/10/20 8:00 AM**	**11/11/20 1:00 PM**		**Alfina Sayyed;Altamash ...**
21	Integrating whole into one	3 days?	28/10/20 8:00 AM	30/10/20 5:00 PM		Altamash Bebal
22	proper integration	2 days?	1/11/20 8:00 AM	3/11/20 5:00 PM		Sahil Memon
23	complete app	1 day?	4/11/20 8:00 AM	4/11/20 5:00 PM		Alfina Sayyed;Pranali Kokate
24	complete web	1.5 days?	10/11/20 8:00 AM	11/11/20 1:00 PM		Altamash Bebal;Sahil Memon
25	⊟**Testing**	11 days?	**22/11/20 8:00 AM**	**7/12/20 5:00 PM**		**Alfina Sayyed;Altamash ...**
26	Unit testing	0.5 days?	22/11/20 8:00 AM	23/11/20 1:00 PM		Alfina Sayyed;Pranali Kokate
27	Integration Testing	1.5 days?	25/11/20 8:00 AM	26/11/20 1:00 PM		Altamash Bebal;Sahil Memon



	Name	Duration	Start	Finish	...	Resource Names
12	DFD Design	0 days?	16/8/20 8:00 AM	17/8/20 5:00 PM		Pranali Kokate
13	Class Diagram	1 day?	17/8/20 8:00 AM	17/8/20 5:00 PM		Prof. Kalpana Bodke;Pranali...
14	Milestone: Architecture app	1 day?	17/8/20 8:00 AM	17/8/20 5:00 PM		Prof. Kalpana Bodke
15	⊟**Implementation**	35.5 days?	**1/9/20 8:00 AM**	**20/10/20 1:00 PM**		**Alfina Sayyed;Altamash ...**
16	App Development	7 days?	1/9/20 8:00 AM	9/9/20 5:00 PM		Alfina Sayyed;Altamash Bebal
17	Database Design and linkin	1 day?	19/9/20 8:00 AM	21/9/20 5:00 PM		Altamash Bebal
18	Web Development	5.333 days?	28/9/20 8:00 AM	5/10/20 10:40 AM		Sahil Memon;Pranali Kokate;...
19	Milestone: App and web De	0.5 days?	20/10/20 8:00 AM	20/10/20 1:00 PM		Altamash Bebal;Sahil Memon
20	⊟**Embedding**	10.5 days?	**28/10/20 8:00 AM**	**11/11/20 1:00 PM**		**Alfina Sayyed;Altamash ...**
21	Integrating whole into one	3 days?	28/10/20 8:00 AM	30/10/20 5:00 PM		Altamash Bebal
22	proper integration	2 days?	1/11/20 8:00 AM	3/11/20 5:00 PM		Sahil Memon
23	complete app	1 day?	4/11/20 8:00 AM	4/11/20 5:00 PM		Alfina Sayyed;Pranali Kokate
24	complete web	1.5 days?	10/11/20 8:00 AM	11/11/20 1:00 PM		Altamash Bebal;Sahil Memon
25	⊟**Testing**	11 days?	**22/11/20 8:00 AM**	**7/12/20 5:00 PM**		**Alfina Sayyed;Altamash ...**
26	Unit testing	0.5 days?	22/11/20 8:00 AM	23/11/20 1:00 PM		Alfina Sayyed;Pranali Kokate
27	Integration Testing	1.5 days?	25/11/20 8:00 AM	26/11/20 1:00 PM		Altamash Bebal;Sahil Memon
28	System Testing	1 day?	29/11/20 8:00 AM	30/11/20 5:00 PM		Alfina Sayyed;Altamash Beb...
29	Milestone:Testing approve	1 day?	5/12/20 8:00 AM	7/12/20 5:00 PM		Prof. Kalpana Bodke
30	⊟**Deployment**	11.5 days?	**25/12/20 8:00 AM**	**11/1/21 1:00 PM**		**Altamash Bebal;Sahil Me...**
31	App deployment	1 day?	25/12/20 8:00 AM	25/12/20 5:00 PM		Sahil Memon
32	web deployment	1 day?	8/1/21 8:00 AM	8/1/21 5:00 PM		Altamash Bebal
33	Milestone: Proper deploym	0.5 days?	10/1/21 8:00 AM	11/1/21 1:00 PM		Sahil Memon;Altamash Bebal
34	⊟**Maintenance**	3.8 days?	**5/2/21 8:00 AM**	**10/2/21 3:24 PM**		**Alfina Sayyed;Altamash ...**
35	Find Bugs	1 day?	5/2/21 8:00 AM	5/2/21 5:00 PM		Alfina Sayyed;Altamash Beb...
36	Bug Fixing	1 day?	8/2/21 8:00 AM	8/2/21 5:00 PM		Sahil Memon;Altamash Bebal
37	Milestone: Complete projec	0.8 days?	10/2/21 8:00 AM	10/2/21 3:24 PM		Alfina Sayyed;Altamash Beb...



	Name	Duration	Start	Finish	Resource Names
12	DFD Design	0 days?	16/8/20 8:00 AM	17/8/20 5:00 PM	Pranali Kokate
13	Class Diagram	1 day?	17/8/20 8:00 AM	17/8/20 5:00 PM	Prof. Kalpana Bodke;Pranali...
14	Milestone: Architecture app	1 day?	17/8/20 8:00 AM	17/8/20 5:00 PM	Prof. Kalpana Bodke
15	**⊟Implementation**	35.5 days?	**1/9/20 8:00 AM**	**20/10/20 1:00 PM**	**Alfina Sayyed;Altamash ...**
16	App Development	7 days?	1/9/20 8:00 AM	9/9/20 5:00 PM	Alfina Sayyed;Altamash Bebal
17	Database Design and linkin	1 day?	19/9/20 8:00 AM	21/9/20 5:00 PM	Altamash Bebal
18	Web Development	5.333 days?	28/9/20 8:00 AM	5/10/20 10:40 AM	Sahil Memon;Pranali Kokate;...
19	Milestone: App and web De	0.5 days?	20/10/20 8:00 AM	20/10/20 1:00 PM	Altamash Bebal;Sahil Memon
20	**⊟Embedding**	10.5 days?	**28/10/20 8:00 AM**	**11/11/20 1:00 PM**	**Alfina Sayyed;Altamash ...**
21	Integrating whole into one	3 days?	28/10/20 8:00 AM	30/10/20 5:00 PM	Altamash Bebal
22	proper integration	2 days?	1/11/20 8:00 AM	3/11/20 5:00 PM	Sahil Memon
23	complete app	1 day?	4/11/20 8:00 AM	4/11/20 5:00 PM	Alfina Sayyed;Pranali Kokate
24	complete web	1.5 days?	10/11/20 8:00 AM	11/11/20 1:00 PM	Altamash Bebal;Sahil Memon
25	**⊟Testing**	11 days?	**22/11/20 8:00 AM**	**7/12/20 5:00 PM**	**Alfina Sayyed;Altamash ...**
26	Unit testing	0.5 days?	22/11/20 8:00 AM	23/11/20 1:00 PM	Alfina Sayyed;Pranali Kokate
27	Integration Testing	1.5 days?	25/11/20 8:00 AM	26/11/20 1:00 PM	Altamash Bebal;Sahil Memon
28	System Testing	1 day?	29/11/20 8:00 AM	30/11/20 5:00 PM	Alfina Sayyed;Altamash Beb...
29	Milestone:Testing approve	1 day?	5/12/20 8:00 AM	7/12/20 5:00 PM	Prof. Kalpana Bodke
30	**⊟Deployment**	11.5 days?	**25/12/20 8:00 AM**	**11/1/21 1:00 PM**	**Altamash Bebal;Sahil Me...**
31	App deployment	1 day?	25/12/20 8:00 AM	25/12/20 5:00 PM	Sahil Memon
32	web deployment	1 day?	8/1/21 8:00 AM	8/1/21 5:00 PM	Altamash Bebal
33	Milestone: Proper deploym	0.5 days?	10/1/21 8:00 AM	11/1/21 1:00 PM	Sahil Memon;Altamash Bebal
34	**⊟Maintenance**	3.8 days?	**5/2/21 8:00 AM**	**10/2/21 3:24 PM**	**Alfina Sayyed;Altamash ...**
35	Find Bugs	1 day?	5/2/21 8:00 AM	5/2/21 5:00 PM	Alfina Sayyed;Altamash Beb...
36	Bug Fixing	1 day?	8/2/21 8:00 AM	8/2/21 5:00 PM	Sahil Memon;Altamash Bebal
37	Milestone: Complete projec	0.8 days?	10/2/21 8:00 AM	10/2/21 3:24 PM	Alfina Sayyed;Altamash Beb...

Chapter 4

Software Requirements Specification

4.1 Overall Description

4.1.1 Product Perspective

The product is an open source. It is a web-app based system implementing client server model. This system is an independent from any other third party system. The user can use both website as well as android application. The main outcome of the system is to provide user a safe and comfortable shopping experience in the pandemic situation. The user will get to buy products without leaving their homes and need not worry about social distancing or getting infected. The system will provide recommendation of products using ML model in future scope.

4.1.2 Product Features

There are three major features in the system. Android application which is developed has very user friendly interface. It is very easy to use and anyone can use it efficiently. It is developed in such way that normal housewives can easily use it without waiting for hours to get deliveries of products. The website is developed in such way that the admin can easily maintain the overall system efficiently. Features for the admin are provided in such way that overall operations can be easily performed and maintained efficiently and easily. The features provided for user in the website are also efficient. User can purchase product through website with recommendations provided by recommendation model which is a future scope for the project. The refer and earn module is an attractive feature through which customer base increases as well as users get the advantages of earned coupons and discounts.

4.1.3 User Classes and Characteristics

The project is a sponsored project. The users of the system are mostly the people who does not want to go outside to buy daily essentials like groceries, fruits, vegetables etc. in the pandemic situation in the fear of getting infected. The user will be able to buy groceries and other essentials in the vicinity of his/her home being safe and also getting the desired products. The project is proposed to provide a safe environment for users in covid-19 to buy the daily essentials and satisfy their requirements.

4.1.4 Operating Environment

Software Requirements:

- OPERATING SYSTEM:Windows,Linux.
- python3,Django,html5,css3,JavaScript
- Android Studio
- Visual studio,sublime editor.
- Databases : Firebase
- Browser : Mozilla,Chrome etc.

Hardware Requirements:
This system can operate on any environment.

- Processor pentium4 or above.
- Ram:2GB or Above
- Hard-disk:40GB or Above
- Android version above 4

4.1.5 Design and Implementation Constraints

The Application is pure android application. GUI is simple and easy which make user to access the application easily and efficiently. The website is also user friendly and can be accessed efficiently. This system focuses one of the features at time.

4.2 System Features

The android application and website developed is very user friendly. The products are differentiated in the forms of categories which makes user to buy them easily. The system also provides an interface through which user can purchase required product, search for the products, place order or cancel order as per requirement. The user gets recommendations from ml model implemented based on its past searches, purchases and frequently visited products. The system also provides refer and earn option which is very trendy nowadays. The system is proposed in such way that the housewives and stay at home ladies can use it with ease and buy the day to day life products which they cannot purchase by going out in the pandemic situation. System provides a safe environment which in turn helps people to satisfy their needs. For admin we have designed a web interface through which admin will manage products, their availability, orders, customers etc.

4.2.1 System Feature

Android application developed in the project has user friendly GUI. The application provides various features which are categorized products, refer and earn, order history, coupons, search products, buy product etc.

Description and Priority

Features/functions provided by android application is one of the main feature of the project. As the user logs in to the system he gets to explore various products, check their availability, add them in cart, buy according to the quantity required, place the order, cancel order and refer and earn feature.

Stimulus/Response Sequences

Stimulus:User Logs in to the system
Response:Authorize the user and allow to log in
Stimulus:User clicks on Products
Response:Display product details (Image,Name,Price,Availability)
Stimulus:User clicks on check availability and enters pin code of his/her area.
Response:If delivery is available in area it displays available else unavailable.
Stimulus:User clicks on add to cart.
Response:Product gets added in to cart and user gets to choose quantity to order.
Stimulus:User clicks on continue.
Response:Alert box appears asking user to confirm.
Stimulus:User clicks on order history.
Response:User gets purchase details and can cancel order if wish.

Functional Requirements

REQ-1: Access to internet and application
REQ-2: Profile created in system
REQ-3: Area pin code to check availability

4.2.2 System Feature

Refer and Earn Module

Description and Priority

The android application developed in the project provides refer and earn feature for user. When user will invite his/her friends to use the application, user gets a coupon as a reward for it. The received coupon can be used by user on any successful order placed before its expiry date. Whenever any user signs up for first time, user gets a referral code and this code is used while inviting others.

Stimulus/Response Sequences

Stimulus:User clicks on refer and earn
Response:A window appears with 6 alphanumeric code and link to invite others.
Stimulus:User clicks on invite friends.
Response:User gets various option to share invite link (via whatsapp, sms, mail etc.)
Stimulus:User shares the link via any medium available to it.
Response:Person whom user has shared link receives the invite.
Stimulus:Person Downloads application from playstore and while signing up enters code shared by the invitee.
Response: new user gets the coupon user who invited him/her also gets a coupon.

Functional Requirements

REQ-1: Access to internet.
REQ-2: Medium to share invite.

4.2.3 System Feature

Product management for admin

Description and Priority

The product management in the warehouse system for admin is made in such a way that admin will be able to do all the task related to product management like adding products, availability of products, orders, customers etc.

Stimulus/Response Sequences

Stimulus:admin login
Response:login successful
Stimulus: add product
Response:adding required details and product added successfully
Stimulus:View orders and set their status
Response: orders description is displayed and status can be set as confirmed,dispatched, placed or delivered

Stimulus:View order summary and income summary
Response: order and income summary displayed successfully

Functional Requirements

REQ-1: Access to internet and browser
REQ-2: Access to database

4.3 External Interface Requirements

4.3.1 User Interfaces

The application developed is very user friendly to use. It is a very light weight app. The application has a nice GUI which user can easily operate. The application is developed such that user can easily view all products and buy the required ones. When application opens, user get to choose to register, login or skin and explore. Once the users registers and logs in, they get to explore products category wise and can add them to their cart or buy. Users can easily see their order history as well as can cancel their order if want. The app also has refer and earn section. The website developed has all the features which are in application. The admin side of the website provides features like add products, edit and view products, view orders and edit orders, income and order summary etc. Which makes admin to handle all operations easily and efficiently..

4.3.2 Hardware Interfaces

The application requires to give permission to access storage. It also needs to give permission to send and receive messages since the app needs to verify the contact number of user.

4.3.3 Software Interfaces

The software uses various libraries. The android application developed uses various libraries.The django python framework is used to developed the website for user as well as for admin. The database that is used is Firebase. It is a real time database with stronger security. OS support also needed.

4.3.4 Communications Interfaces

The Product is a light web-app, there is no such large communication in the system. Only Databases access, that also done in real time. The order communication related with the order confirmation is done via direct contact with user. Also https standard is used in-order to gain the access to the browser.

4.4 Nonfunctional Requirements

4.4.1 Performance Requirements

Performance of overall system is very efficient and well optimize. The product purchasing using either application or website is done within seconds. It does not require much time. The images and content takes few seconds to appear on screen since the internet connection is required to connect to database.

4.4.2 Safety Requirements

This system does not contain any critical data. Still it provide. The databases that are accessed are secure. In case of any updates in libraries used can lead to the failure in systems.

4.4.3 Security Requirements

All the Libraries used are certified and standard as well as all the framework provide basic security to the system. There is no Critical data in System. Although data stored in Databases with encryption. The access to the database is only given to the admin. The data stored is not given access to any outside third party.

Chapter 5

System Design

5.1 System Requirements Definition

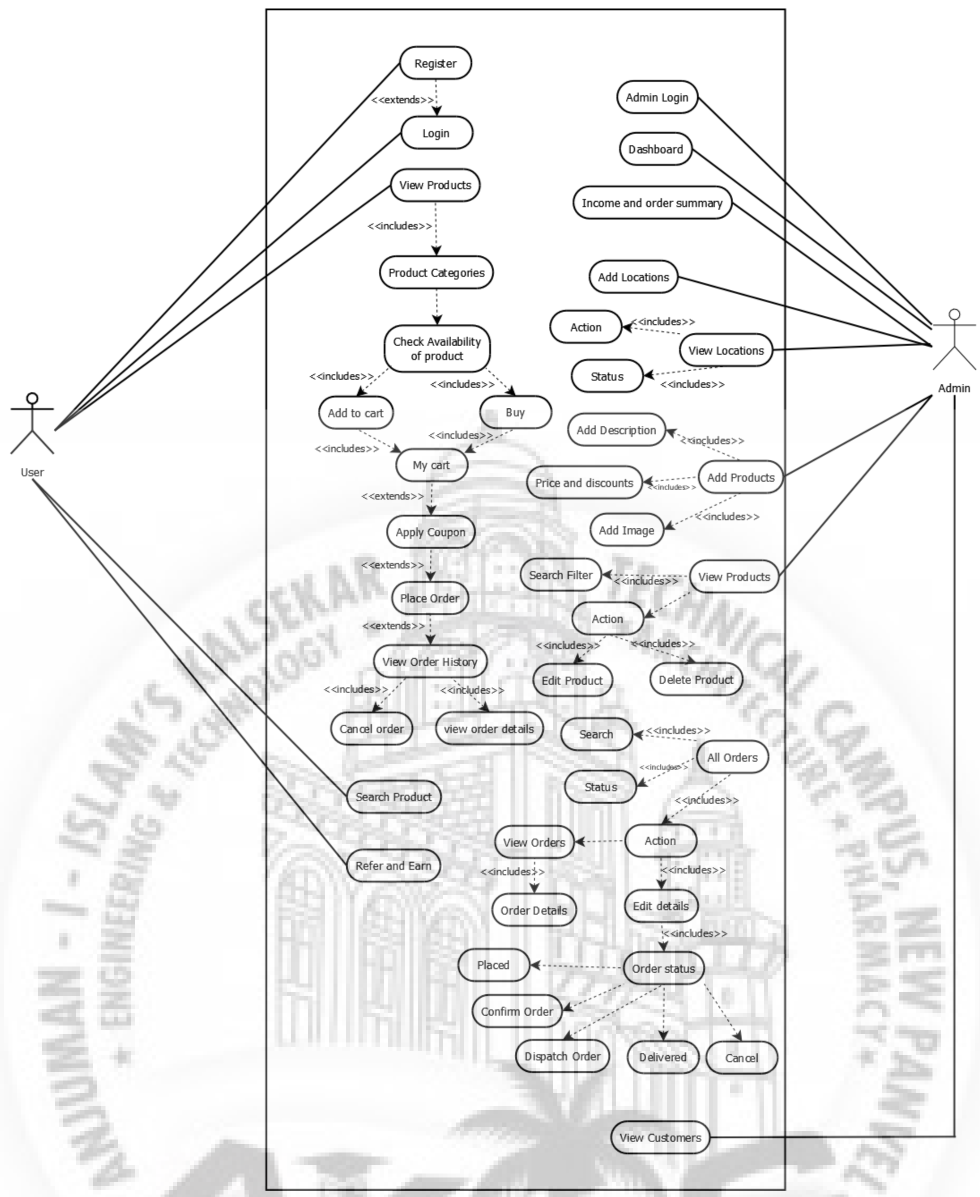
System requirement definitions specify what the system should do, its functionality and its essential and desirable system properties. The techniques applied to elicit and collect information in order to create system specifications and requirement definitions involve consultations, interviews, requirements workshop with customers and end users. The objective of the requirements definition phase is to derive the two types of requirement:

5.1.1 Functional requirements

They define the basic functions that the system must provide and focus on the needs and goals of the end users.

Use-case Diagram

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved.



Data-flow Diagram

A data-flow diagram is a way of representing a flow of a data of a process or a system. The DFD also provides information about the outputs and inputs of each entity and the process itself.

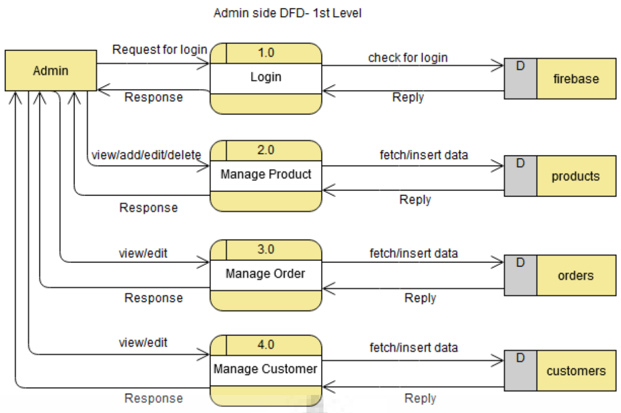


Figure 5.1: DFD Level 1 Admin

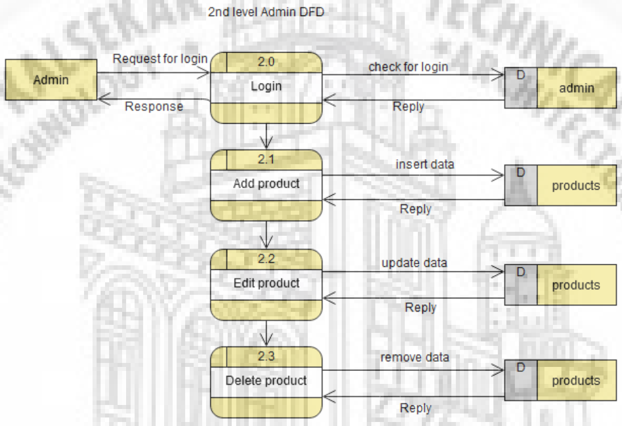


Figure 5.2: DFD Level 2 Admin

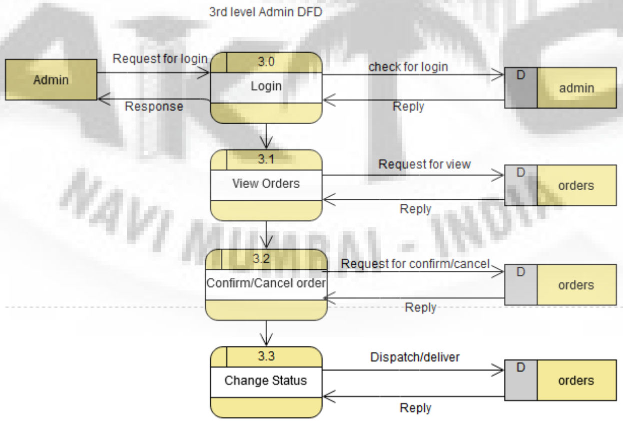


Figure 5.3: DFD Level 3 Admin

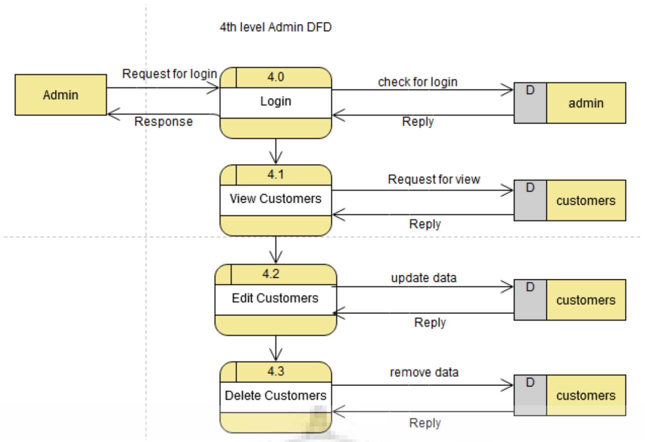


Figure 5.4: DFD Level 4 Admin

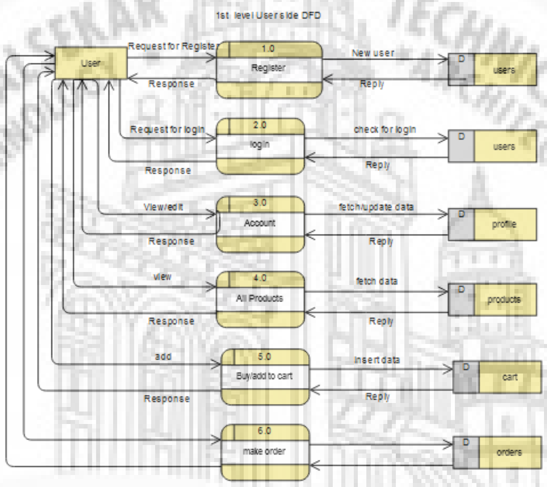


Figure 5.5: DFD Level 1 User

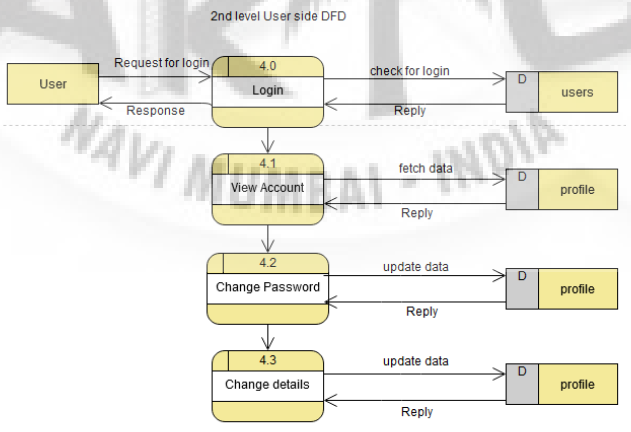


Figure 5.6: DFD Level 2 User

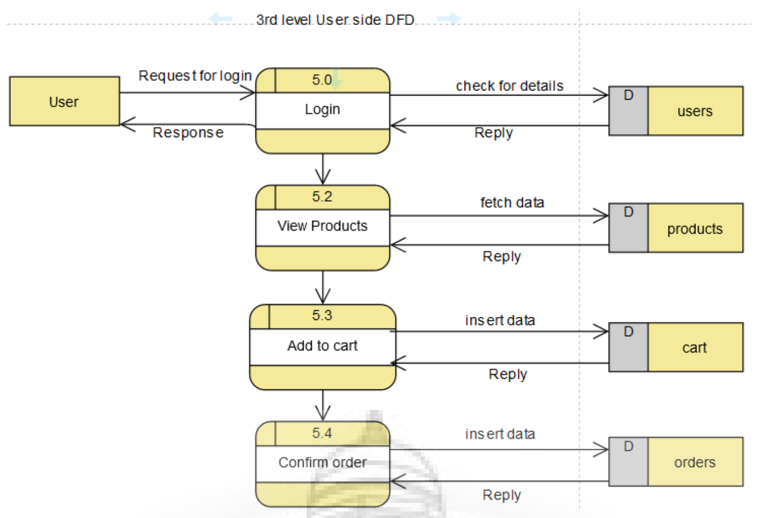


Figure 5.7: DFD Level 3 User

5.1.2 System requirements (non-functional requirements)

These are non-functional system properties such as availability, performance and safety etc. They define functions of a system, services and operational constraints in detail.

a. Usability - Application implementation is feasible using technologies that are accessible to the end-users.

b. The interfaces are compatible with Web View and Mobile view.

c. Performance Efficiency -Application is able to perform well in a proper time constraint.

d. Multi User System -Application is able to consider the presence of more than one user in the same environment. All the features of the system operates properly for all users and provides proper transparency.

e. Time Efficiency - Time taken for the executing of system is less.

5.2 System Architecture Design

A system architecture is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system.

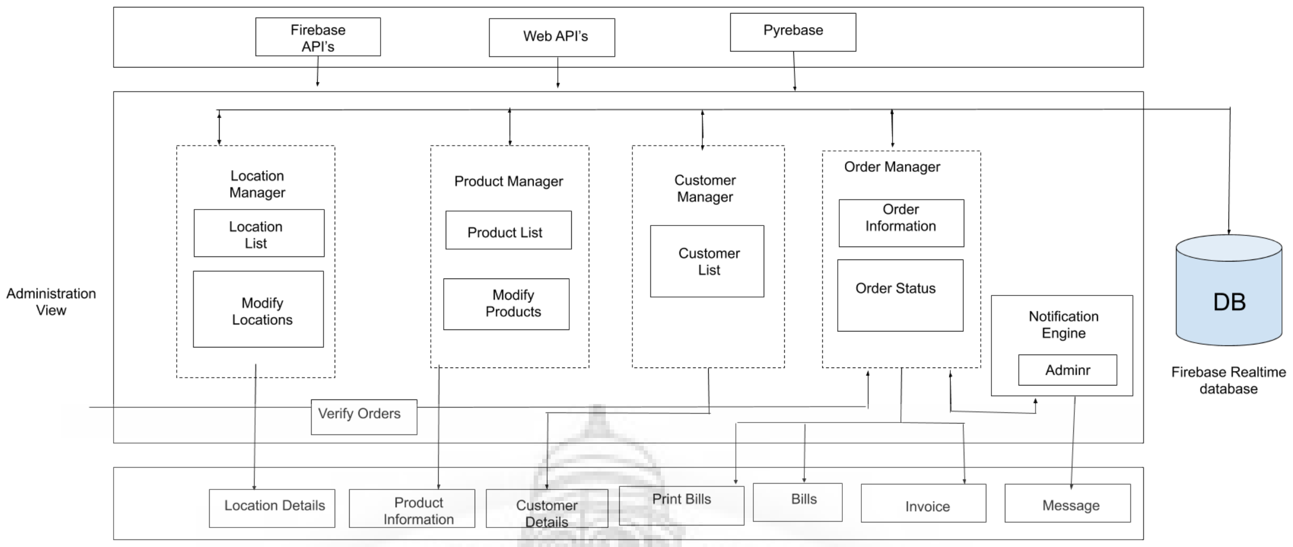


Figure 5.8: System Architecture Admin side

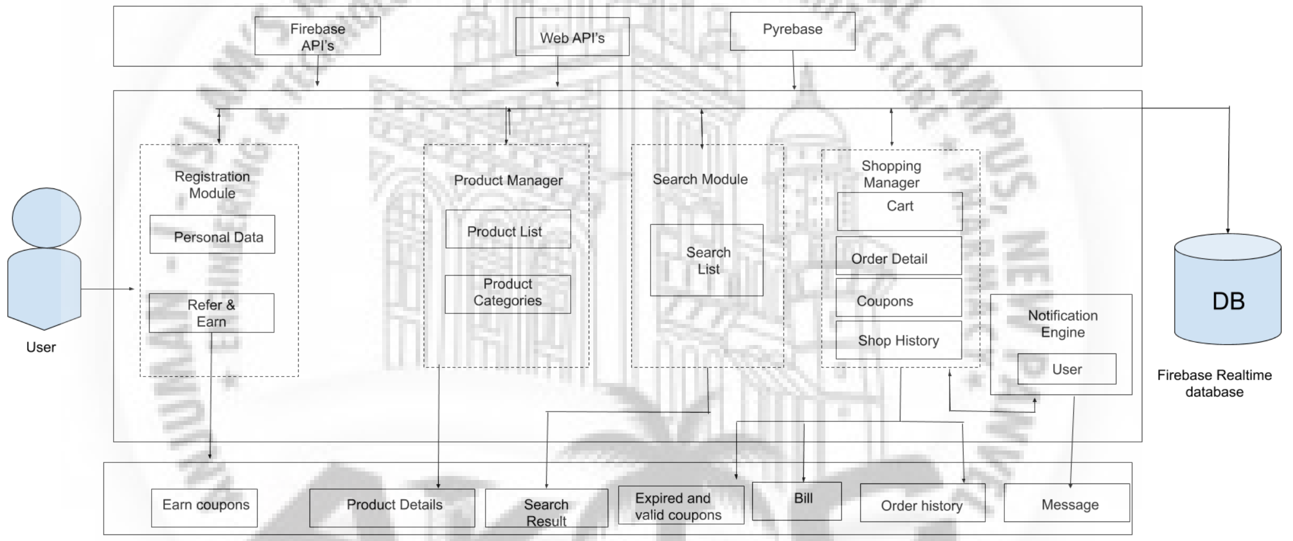


Figure 5.9: System Architecture User side

5.3 Sub-system Development

The system has two modules. First one is android application module and second is web module for user as well as for admin. The inputs for the android application are details from database like all product details, images etc. Also users details like its login info and profile details which are also fetched from database.The output provided by the module will be users purchase , order history, earned coupons. The input for web module for user are login details and output will be users purchase, order history, purchase details, order details etc. For admin, input will be its login credentials, in case of product addition an image, for order confirmation order details etc. As a output admin can view,edit or delete products,view orders, change status of orders, view customers, view income and order summaries etc.

5.3.1 Android Module

Android application module enables user to view all products available and can order according to requirements. It is an important module. This is the module through which user interacts with the system and satisfies its requirement of essential needs.

Registration

This module deals with registration of user in the system.The user has to register himself/herself in the system to use the services of the application. It takes details like Name,Address,Phone Number,Pincode etc. Once the user gets registered, he/she can log in to the app/website and start using the services.

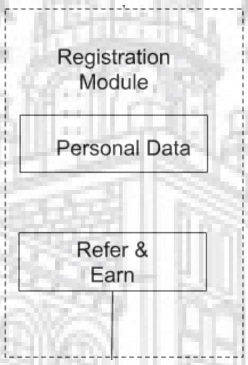


Figure 5.10: Registration

Products

This module deals with products that are available in the warehouse for user to buy. The products are divided into various categories. User can explore them by categories.

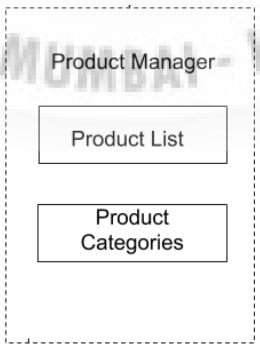


Figure 5.11: Products

Search

This module deals with the search of the products that are available for user. User can enter the product name that he/she wants to buy and the system will fetch the search result and the result will be displayed to user .

Search Module
Search List

Figure 5.12: Search

Shopping Manager

This module deals with the orders made by the user. User can add the products in the cart and can see them in the cart section. User can view the order history of the products made in the past as well as status of the current order. The user can apply the coupons if applicable while placing the order.

Figure 5.13: Shopping Manager

5.3.2 Web Module

Through web module user can explore all features provided by application. For admin, web module is important, through this web module admin is able to perform all product management and order management functions.It has following sub modules.

Shopping Manager

This module deals with cart, order details and coupons. When user clicks on the cart he gets to see the products added in the cart where he gets the options to increase/decrease the quantity of added product and remove the product from cart. User can see the total amount payable along with total savings. If user has the coupons he gets to apply them on the order. The user is then asked for confirmation to place the order and gets notification for the same. User can see order history, Bills and earned/expired coupon details.

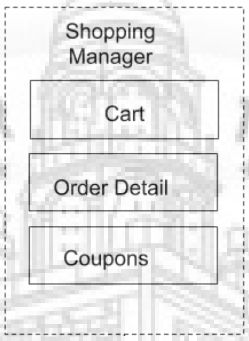


Figure 5.14: Shop Manager

Location Manager

This module deals with location management. This module is for admin. Admin can manages the locations where he wants to provide the services. When user checks if the delivery is available in his/her area, the pin code entered by the user is matched with the locations added by the admin. The admin can view all locations added and can modify locations i.e. can add new locations or delete existing locations.

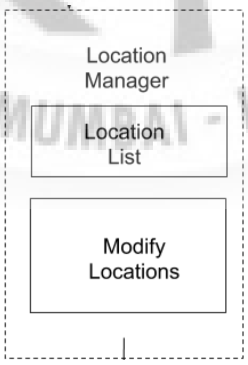


Figure 5.15: Location Manager

Product Manager

This module deals with product management. Admin can manage all the product related activities. Admin can see the products which are currently available for the customers to buy using the application. All product details are fetched from the database and displayed. Admin can add the new products by uploading the image and inserting the product details like its market price, discount, description etc. Admin can modify the product details anytime and the changes reflects in the database. Admin can delete the product if wants to and same changes reflects in the database.

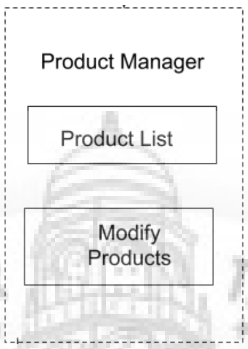


Figure 5.16: Product manager

Order Manager

This module deals with order management. Admin can perform all the functions related to order management like viewing orders, verifying orders, order confirmation, Status of order, bills, invoice etc. Once the order is placed by the user admin gets the notification of it by notification engine. Details of order are fetched from the database and admin can confirm order, print the bill and set the status of order as placed/delivered/dispatched etc.

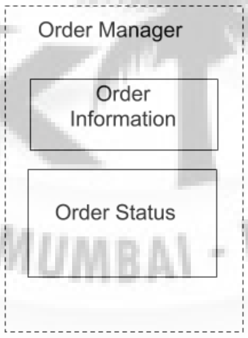


Figure 5.17: Order manager

Customer Manager

This module deals with customer management. Admin can manage the customers. The registered user's data is fetched from the database and admin can view the details. If the admin wants he can remove the unwanted customers whose id's are fake or details are incorrect.

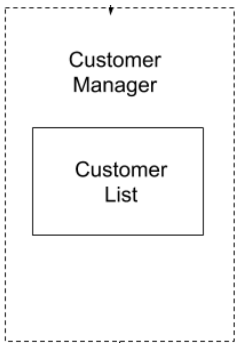


Figure 5.18: customer manager Manager

5.3.3 Django Module

Django is a high-level Python web framework that enables rapid development of secure and maintainable websites. Built by experienced developers, Django takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel. It is free and open source, has a thriving and active community, great documentation, and many options for free and paid-for support. We have used django framework for website development. We have used it for secure connection to firebase.

5.4 Systems Integration

System integration (SI) is an engineering process or phase concerned with joining different subsystems or components as one large system. It ensures that each integrated subsystem functions as required. Different Sub-Modules Integrated in one full System. SI is also used to add value to a system through new functionality provided by connecting functions of different systems.

5.4.1 Class Diagram

In software engineering, a class diagram in the Unified Modeling Language is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations, and the relationships among objects.

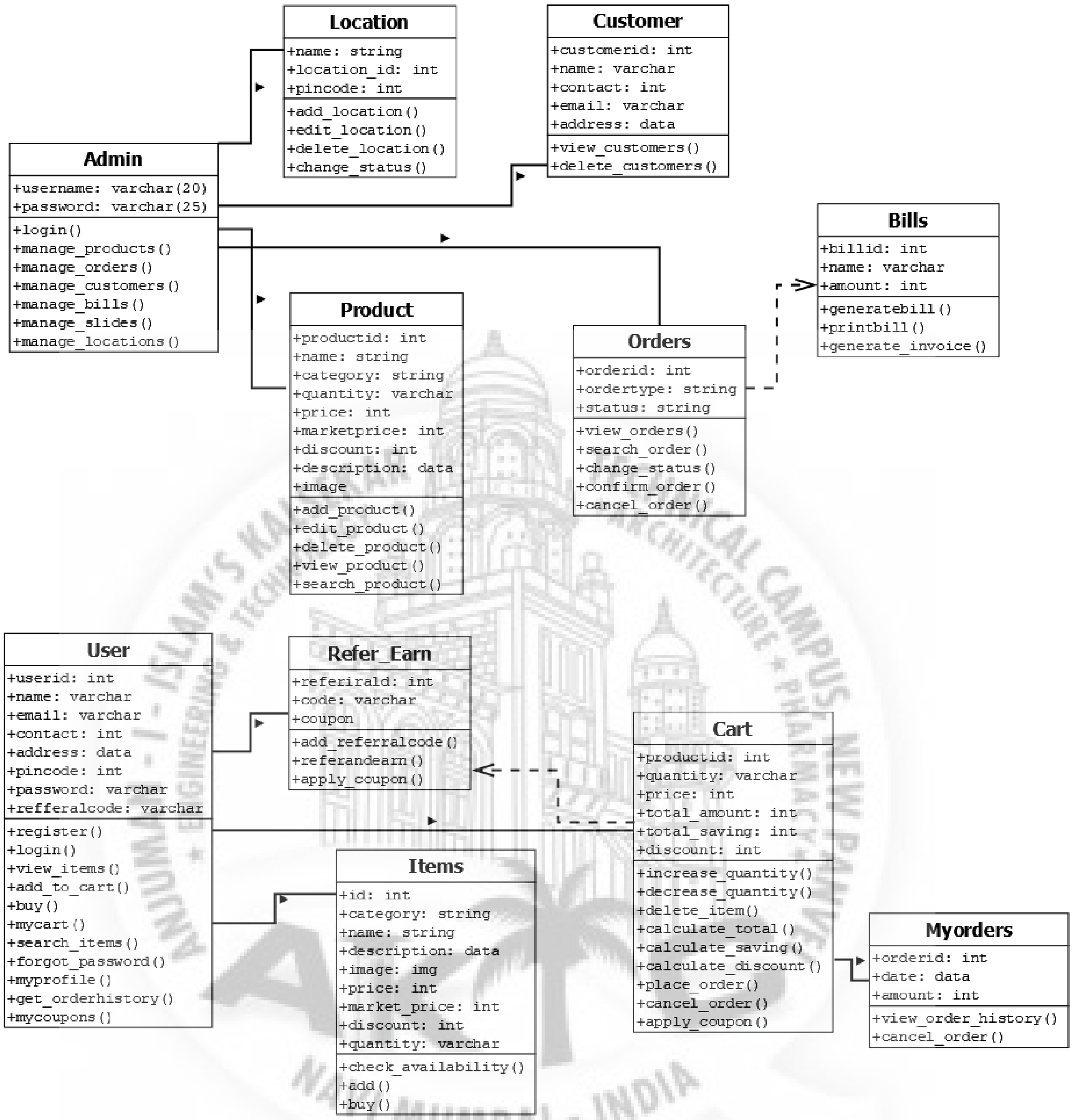


Figure 5.19: Class Diagram

5.4.2 Sequence Diagram

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.

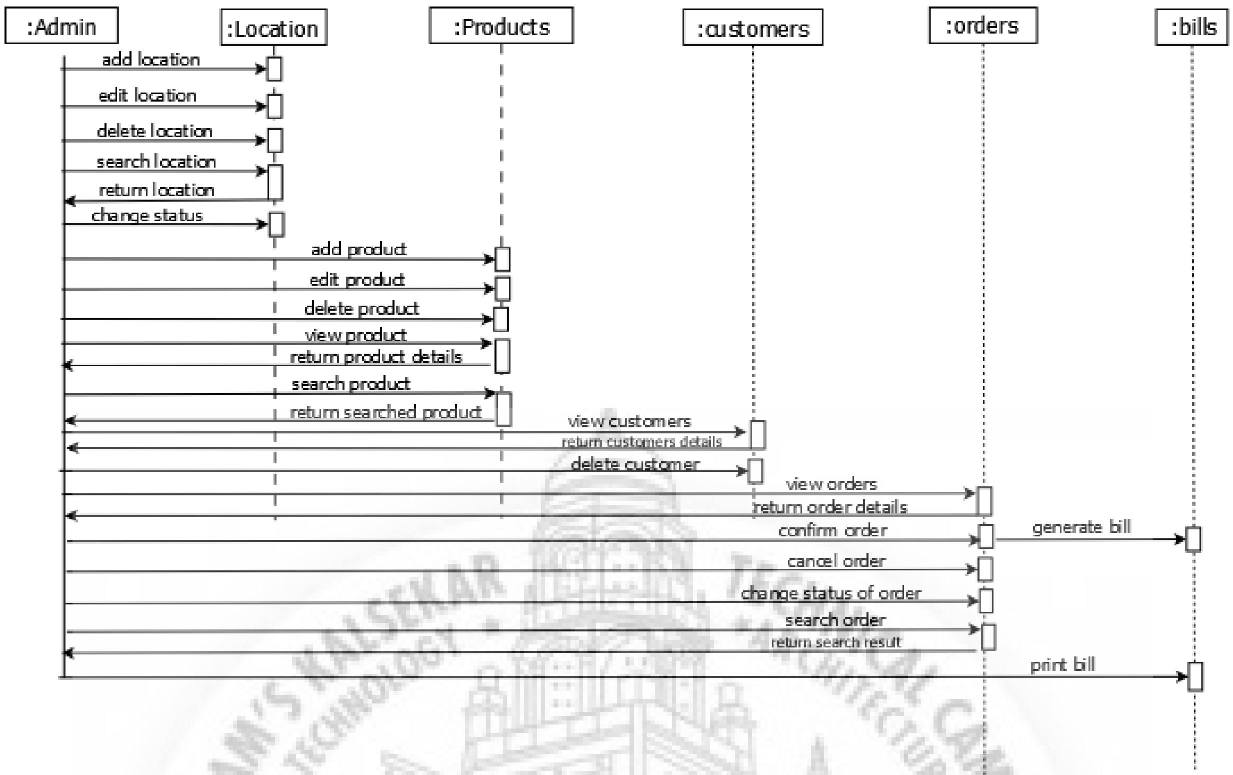


Figure 5.20: Sequence Diagram Admin side

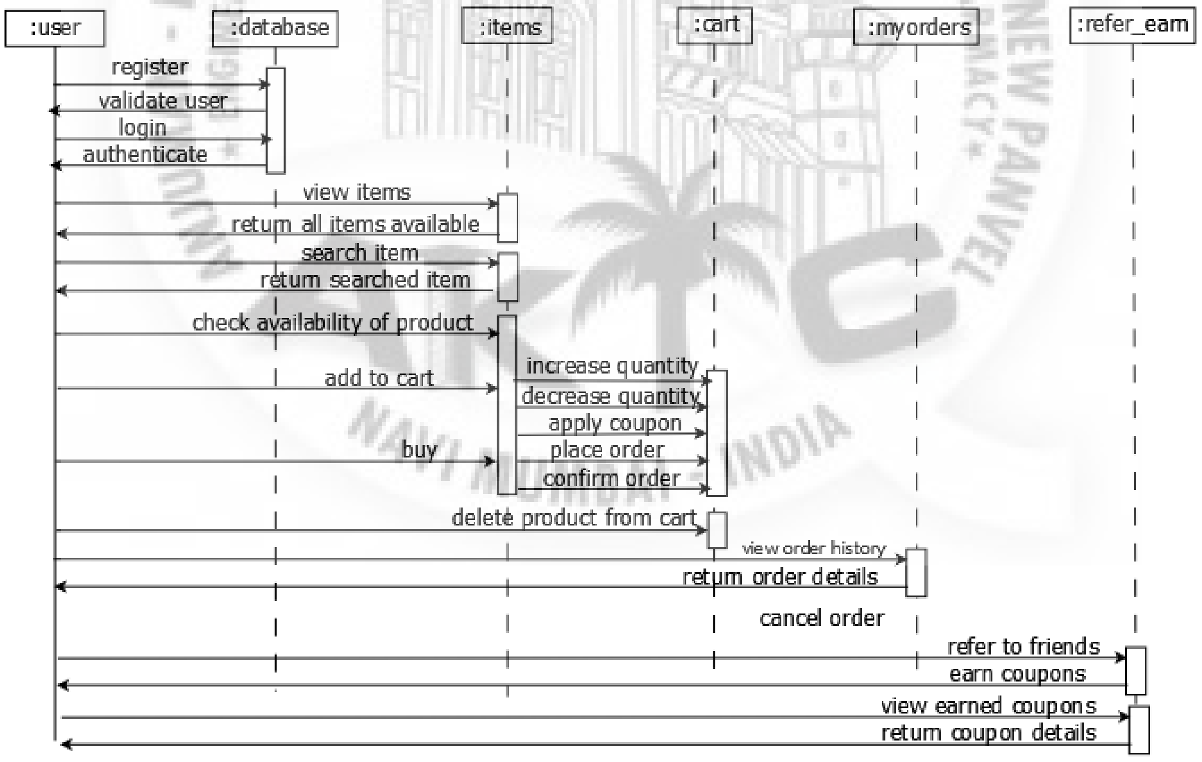


Figure 5.21: Sequence Diagram User side

Chapter 6

Implementation

6.1 Android Application Module

This module is developed using Android studio, java, xml. This module takes login credentials from user and allows user to explore all features provided via app.

build.gradle

```
apply plugin: 'com.android.application'
apply plugin: 'com.google.gms.google-services'
android {
    compileSdkVersion 29
    buildToolsVersion "29.0.3"
    defaultConfig {
        applicationId "com.AtgMart.anytimegroceries"
        minSdkVersion 22
        targetSdkVersion 29
        versionCode 1
        versionName "1.0"

        testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
    }

    buildTypes {
        release {
            debuggable false
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'
                ), 'proguard-rules.pro'
        }
    }
}

dependencies {
    implementation fileTree(dir: "libs", include: ["*.jar"])
    implementation 'de.hdodenhof:circleimageview:3.1.0'
    implementation 'com.airbnb.android:lottie:2.5.6 '
    implementation "com.android.support:support-v4:+"
//     implementation 'com.squareup.picasso:picasso:2.71828'
    implementation 'com.nineoldandroids:library:2.4.0'
    implementation 'com.daimajia.slider:library:1.1.5@aar'
    implementation 'androidx.appcompat:appcompat:1.1.0'
    implementation 'androidx.constraintlayout:constraintlayout:1.1.3'
    implementation 'androidx.recyclerview:recyclerview:1.1.0'
    implementation 'com.google.android.material:material:1.1.0'
    implementation 'androidx.legacy:legacy-support-v4:1.0.0'
//    implementation 'com.google.firebase:firebase-auth:19.2.0'
//    implementation 'com.google.firebase:firebase-database:19.2.0'
    implementation 'androidx.navigation:navigation-runtime:2.1.0'
    implementation 'com.airbnb.android:lottie:2.7.0'
    implementation 'com.google.firebase:firebase-database:19.3.1'
    implementation 'com.google.firebase:firebase-auth:19.3.2'
    testImplementation 'junit:junit:4.12'
    androidTestImplementation 'androidx.test.ext:junit:1.1.1'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.2.0'
    implementation 'com.google.firebase:firebase-storage:19.1.1'
    implementation 'com.google.android.gms:play-services-location:17.0.0'
    implementation 'com.google.firebase:firebase-core:17.2.2'
    implementation 'com.google.firebase:firebase-firestore:21.3.1'
    implementation 'com.firebaseui:firebase-ui-storage:6.2.0'
    implementation 'com.firebaseui:firebase-ui-database:6.2.0'
    implementation 'androidx.lifecycle:lifecycle-extensions:2.0.0'
    implementation 'com.firebaseui:firebase-ui:0.4.3'
    implementation "androidx.navigation:navigation-fragment:2.3.0"
    implementation "androidx.navigation:navigation-ui:2.3.0"

//    implementation fileTree(dir: "libs", include: ["*.jar"])
//    implementation 'de.hdodenhof:circleimageview:3.1.0'
//    implementation 'com.airbnb.android:lottie:2.7.0'
    implementation "com.android.support:support-v4:+"
    implementation 'com.squareup.picasso:picasso:2.5.2'
    //Never update this version because its supports Slideshow
    implementation 'com.nineoldandroids:library:2.4.0'
    implementation 'com.daimajia.slider:library:1.1.5@aar'
    implementation 'androidx.constraintlayout:constraintlayout:1.1.3'
    implementation 'com.google.android.material:material:1.1.0'
    implementation 'androidx.cardview:cardview:1.0.0'
    androidTestImplementation 'androidx.test.ext:junit:1.1.1'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.2.0'

    implementation("com.mikepenz:materialdrawer:6.0.8@aar") {
        transitive = true
    }
    implementation('com.mikepenz:aboutlibraries:6.0.6@aar') {
        transitive = true
    }
    implementation "com.mikepenz:itemanimators:1.0.1@aar"

}
```

AndroidManifest.xml

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    package="com.AtgMart.anytimegroceries">
    <!-- if you want to load images from the internet -->
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.INTERNET" /> <!-- if you
        want to load images from a file OR from the internet -->
    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />

    <application
        android:name="com.AtgMart.anytimegroceries.MyApp"
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/MyTheme">
        <activity  android:name="com.AtgMart.anytimegroceries.SummaryActivity"
            android:screenOrientation="portrait"/>
        <activity  android:name="com.AtgMart.anytimegroceries.CouponActivity"
            android:screenOrientation="portrait"/>
        <activity  android:name="com.AtgMart.anytimegroceries.ReferActivity"
            android:screenOrientation="portrait"/>
        <activity
            android:name="com.AtgMart.anytimegroceries.CancelOrder"
            android:screenOrientation="portrait" />
        <activity
            android:name="com.AtgMart.anytimegroceries.PlacedOrderActivity"
            android:screenOrientation="portrait" />
        <activity
            android:name="com.AtgMart.anytimegroceries.Splash2"
            android:screenOrientation="portrait" />
        <activity
            android:name="com.AtgMart.anytimegroceries.EmptyCart"
            android:screenOrientation="portrait" />
        <activity
            android:name="com.AtgMart.anytimegroceries.ValidateOTP"
            android:screenOrientation="portrait" />
        <activity
            android:name="com.AtgMart.anytimegroceries.StartActivity"
            android:screenOrientation="portrait" />
        <activity
            android:name="com.AtgMart.anytimegroceries.Splash"
            android:screenOrientation="portrait">
            <intent-filter>
                <action  android:name="android.intent.action.MAIN" />

                <category  android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity
            android:name="com.AtgMart.anytimegroceries.SearchResultsActivity"
            android:screenOrientation="portrait" />
        <activity
            android:name="com.AtgMart.anytimegroceries.RegisterActivity"
            android:screenOrientation="portrait" />
        <activity
            android:name="com.AtgMart.anytimegroceries.Recent_Products_Adapter"
            android:screenOrientation="portrait" />
        <activity
            android:name="com.AtgMart.anytimegroceries.ProfileActivity"
            android:screenOrientation="portrait" />
        <activity
            android:name="com.AtgMart.anytimegroceries.ProductDetailActivity"
            android:screenOrientation="portrait" />
        <activity
            android:name="com.AtgMart.anytimegroceries.Product"
            android:screenOrientation="portrait" />
        <activity
            android:name="com.AtgMart.anytimegroceries.OtpForgetActivity"
            android:screenOrientation="portrait" />
        <activity
```

```
                android:name="com.AtgMart.anytimegroceries.OrderDetailsActivity"
                android:screenOrientation="portrait" />
        <activity
                android:name="com.AtgMart.anytimegroceries.OrderActivity"
                android:screenOrientation="portrait" />
        <activity
                android:name="com.AtgMart.anytimegroceries.MyCart"
                android:screenOrientation="portrait" />
        <activity
                android:name="com.AtgMart.anytimegroceries.LoginActivity"
                android:screenOrientation="portrait" />
        <activity
                android:name="com.AtgMart.anytimegroceries.Item_Order_Detail"
                android:screenOrientation="portrait" />
        <activity
                android:name="com.AtgMart.anytimegroceries.HomeActivity"
                android:screenOrientation="portrait" />
        <activity
                android:name="com.AtgMart.anytimegroceries.GridItem"
                android:screenOrientation="portrait" />
        <activity
                android:name="com.AtgMart.anytimegroceries.ForgetActivity"
                android:screenOrientation="portrait" />
        <activity
                android:name="com.AtgMart.anytimegroceries.Customer_Order_Adapter"
                android:screenOrientation="portrait" />
        <activity android:name="com.AtgMart.anytimegroceries.Converter" />
        <activity android:name="com.AtgMart.anytimegroceries.
            ConnectivityReceiver" />
        <activity
                android:name="com.AtgMart.anytimegroceries.ChangePassword"
                android:screenOrientation="portrait" />
        <activity
                android:name="com.AtgMart.anytimegroceries.ChangeClientDetail"
                android:screenOrientation="portrait" />
        <activity
                android:name="com.AtgMart.anytimegroceries.Category_wise_products"
                android:screenOrientation="portrait" />
        <activity
                android:name="com.AtgMart.anytimegroceries.Cart_Item_Adapter"
                android:screenOrientation="portrait" />
        <activity
                android:name="com.AtgMart.anytimegroceries.Bsp_Grid"
                android:screenOrientation="portrait" />
        <activity
                android:name="com.AtgMart.anytimegroceries.AddToCart"
                android:screenOrientation="portrait" />
        <activity
                android:name="com.AtgMart.anytimegroceries.AddorRemoveCallbacks"
                android:screenOrientation="portrait" />
        <activity
                android:name="com.AtgMart.anytimegroceries.NoOrders"
                android:screenOrientation="portrait" />
        <activity
                android:name="com.AtgMart.anytimegroceries.VerifyPhone"
                android:screenOrientation="portrait" />

        <meta-data
                android:name="preloaded_fonts"
                android:resource="@array/preloaded_fonts" />
    </application>
```

```

</manifest>
```

6.2 Web and Django Module

This module is implemented using Python programming language and Django framework for web Development. The website created for user allows user to perform all the functionalities same as that of android application. The user can shop using application as well as through website. The web module created for admin, allows admin to perform product management and order management efficiently with ease. It helps admin to manage the warehouse efficiently.

Figure 6.1: Django Web Module

Views.py : User

```
 from django.shortcuts import render, redirect
import pyrebase
from django.contrib import auth as dauth

firebaseConfig = {
    "apiKey": "AIzaSyC6F7FprRkBSoziZjlPnJec6dQGzpjZmQo",
    "authDomain": "newatg-11a13.firebaseapp.com",
    "databaseURL": "https://newatg-11a13.firebaseio.com",
    "projectId": "newatg-11a13",
    "storageBucket": "newatg-11a13.appspot.com",
    "messagingSenderId": "45402315445",
    "appId": "1:45402315445:web:7bd5a80c1b31cfb1f4fcfc",
    "measurementId": "G-9NZL1LL0YB"
  }
firebase = pyrebase.initialize_app(firebaseConfig)
database = firebase.database()
db = firebase.database()


def home(request):
    slideshow=db.child("SlideShow").get()
    slideshow2={}
    for order in slideshow.each():
        slideshow2[order.key()]=order.val()
    allProducts=db.child("Warehouse").child("AllProducts").get()
    allProducts2={}
    allVegetables=db.child("Warehouse").child("Vegetables").get()
    allVegetables2={}
    allFruits=db.child("Warehouse").child("Fruits").get()
    allFruits2={}
    allCombo=db.child("Warehouse").child("Combo").get()
    allCombo2={}
    allHerbs=db.child("Warehouse").child("Herbs").get()
    allHerbs2={}
    for product in allProducts.each():
        allProducts2[product.key()]=product.val()

    for product in allVegetables.each():
        allVegetables2[product.key()]=product.val()

    for product in allFruits.each():
        allFruits2[product.key()]=product.val()

    for product in allCombo.each():
        allCombo2[product.key()]=product.val()

    for product in allHerbs.each():
        allHerbs2[product.key()]=product.val()
    return render(request,"Frontend/index.html",{"All":allProducts2,"
        allVegetables":allVegetables2,
    "allFruits":allFruits2,"allCombo":allCombo2,"allHerbs":allHerbs2,"SlideShow"
        :slideshow2})

    # return render(request,"Frontend/shop_grid.html",{"All":allProducts2})

def placeOrder(request):
    return render(request, "Frontend/checkout.html")

def updateProfile(request,uid):
    customer=db.child("Customers").child(uid).get()
    return render(request, "Frontend/update_profile.html",{"Cust":customer.val()
        })


def postsignup(request):
    print("In SignUp")
    auth=firebase.auth()
    fullname = request.POST.get('fullname')
    emailaddress = request.POST.get('emailaddress')
    phone = request.POST.get('phone')
    flat = request.POST.get('flat')
    street = request.POST.get('street')
    pincode = request.POST.get('pincode')
    locality = request.POST.get('locality')
    password = request.POST.get('password1')
    address = flat +", "+ street +", "+ locality
    print(emailaddress)
    try:
        user = auth.create_user_with_email_and_password(emailaddress, password)
        user = auth.refresh(user['refreshToken'])
        # print(user['userId'])

        # print(user)
        uid = user['userId']
        data = {"Address":address, "Email":emailaddress, "Mobile":phone, "Name":
            fullname, "Pincode":pincode, "Referral Code":'', "Uid":uid}

        # print(uid)
        # results = db.child("users").push(data, user['idToken'])

        database.child("Customers").child(uid).set(data)
        message = "Signup Successful Please Login"
        # redirect('signin')
        return render(request,"Frontend/sign_in.html")

    except Exception as e:
        message = "Unable to create account try again"
        print(e)
        return render(request,"Frontend/sign_up.html",{"messg":message})
    return render(request,"Frontend/shop_grid.html")

def postsignin(request):
    # auth=firebase.auth()
    # emailaddress = request.POST.get("emailaddress")
    # password = request.POST.get("password1")
    # db = firebase.database()
    # allProducts=db.child("Warehouse").child("AllProducts").get()
    # allProducts2={}
    # for product in allProducts.each():
    #     # print(product.key())
    #     allProducts2[product.key()]=product.val()
    # try:
    #     user = auth.sign_in_with_email_and_password(emailaddress, password)
    #     return render(request, "Frontend/shop_grid.html",{"User":dauth.
        get_user_model,"All":allProducts2})

    # except:
    #     message="Invalid Emailid or Password"
    return render(request,"Frontend/sign_in.html")
    # session_id=user['idToken']
    # request.session['uid']=str(session_id)

def logout(request):
    dauth.logout(request)
    return render(request,'Frontend/shop_grid.html')

def signin(request):
   return render(request, "Frontend/sign_in.html")

def signup(request):
   return render(request, "Frontend/sign_up.html")

def recoverpassword(request):
    auth=firebase.auth()
    email = request.POST.get("emailaddress")
    try:
        auth.send_password_reset_email(email)
        message="Password recovery link has been sent on your E-mail id"
    except Exception:
        message="Unable to send password recovery link"
        return render(request, "Frontend/forgot_password.html", {"messg":message
            })
    return render(request, "Frontend/forgot_password.html", {"messg":message})
```

```
def forgotpassword(request):
    return render(request, "Frontend/forgot_password.html")


def aboutus(request):
    return render(request, "Frontend/about_us.html")

def bill(request):
    return render(request, "Frontend/bill.html")

def blogdetailview(request):
    return render(request, "Frontend/blog_detail_view.html")

def blogleftsidebar(request):
    return render(request, "Frontend/blog_left_sidebar.html")

def blogleftsidebarsingleview(request):
    return render(request, "Frontend/blog_left_sidebar_single_view.html")

def blognosidebar(request):
    return render(request, "Frontend/blog_no_sidebar.html")

def blogrightsidebar(request):
    return render(request, "Frontend/blog_right_sidebar.html")

def career(request):
    return render(request, "Frontend/career.html")

def checkout(request):
    return render(request, "Frontend/checkout.html")

def contactus(request):
    return render(request, "Frontend/contact_us.html")

def dashboardmyaddresses(request):
    return render(request, "Frontend/dashboard_my_addresses.html")

def dashboardmyorders(request):
    return render(request, "Frontend/dashboard_my_orders.html")

def dashboardmyrewards(request):
    return render(request, "Frontend/dashboard_my_rewards.html")


def dashboardmywallet(request):
    return render(request, "Frontend/dashboard_my_wallet.html")


def dashboardmywishlist(request):
    return render(request, "Frontend/dashboard_my_wishlist.html")

def dashboardoverview(request):
    return render(request, "Frontend/dashboard_overview.html")

def faq(request):
    return render(request, "Frontend/faq.html")

def jobdetailview(request):
    return render(request, "Frontend/job_detail_view.html")

def offers(request):
    return render(request, "Frontend/offers.html")

def orderplaced(request):
    return render(request, "Frontend/order_placed.html")

def ourblog(request):
    return render(request, "Frontend/our_blog.html")

def press(request):
    return render(request, "Frontend/press.html")

def privacypolicy(request):
    return render(request, "Frontend/privacy_policy.html")

def refundreturnpolicy(request):
    return render(request, "Frontend/refund_and_return_policy.html")

def requestproduct(request):
    return render(request, "Frontend/request_product.html")

def shopgrid(request):
    return render(request, "Frontend/shop_grid.html")

def signout(request):
    return render(request, "Frontend/signout.html")

def singleproductview(request,Name,Cat):
    # print(Cat)
    db=firebase.database()
    allProducts=db.child("Warehouse").child("AllProducts").child(Name).get()
    catWise=db.child("Warehouse").child(Cat).get()
    catWise2={}
    counter=0
    for product in catWise.each():
        if counter==3:
            break
        if product.key()!=Name:
            catWise2[product.key()]=product.val()
        counter+=1
    return render(request, "Frontend/single_product_view.html",{"Product":
        allProducts.val(),"Cat":catWise2})

def termandconditions(request):
    return render(request, "Frontend/term_and_conditions.html")


def grid(request,cat):
    return render(request, "Frontend/shop_grid.html",{"Cat":cat})


def CustomerUpdate(request ,uid):
    # print("In SignUp")
    # auth=firebase.auth()
    # fullname = request.POST.get('fullname')
    # emailaddress = request.POST.get('emailaddress')
    # phone = request.POST.get('phone')
    # # flat = request.POST.get('flat')
    # add=request.POST.get('add')
    # # uid=request.POST.get('uid')
    # print(uid)
    # updateData={
```

```
    # 'Address':add,
    # 'Email':emailaddress,
    # 'Name':fullname,
    # 'Mobile':phone,
    # }
    # db = firebase.database()
    # print(updateData)
    print(uid)
    return redirect('INDEX')
    # db.child("Customers").child(uid).update(updateData)
    # return render(request, "Frontend/dashboard_overview.html")


def ordernow(request, uid, ordid, pro_avail):
    print(uid, ordid, pro_avail)
    pro_avail = pro_avail.split(",")
    ts = timestamp()
    for i in pro_avail:
        ordering(uid, i, ordid, ts)
    # return render(request,"Frontend/index.html")
    return render(request, "Frontend/order_placed.html")

def ordering(uid, name, orderid, ts):
    print(uid, name, orderid)
    db=firebase.database()
    allProducts = db.child("Customers").child(uid).child("Cart").child(name).get
        ()
    print(allProducts.val())
    allProduct = dict(allProducts.val())
    prqty = allProduct["ProductQty"]
    rp = allProduct["RetailPrice"]
    mp = allProduct["MarketPrice"]
    cat = allProduct["Catogory"]

    ast = db.child("Warehouse").child("AllProducts").child(name).child("Stock").
        get().val()
    cst = db.child("Warehouse").child(cat).child(name).child("Stock").get().val
        ()

    x = str(int(ast) - int(prqty))
    y = str(int(cst) - int(prqty))

    db.child("Warehouse").child("AllProducts").child(name).update({"Stock": x})
    db.child("Warehouse").child(cat).child(name).update({"Stock": y})

    print("\n\n",name, prqty, rp, mp)

    disc = db.child("Customers").child(uid).child("OrderDiscount").get().val()
    pri = db.child("Customers").child(uid).child("OrderPrice").get().val()

    disc = str(int(disc) + int(prqty) * (int(mp) - int(rp)))
    pri = str(int(pri) + int(prqty) * int(rp))
    print("disc-",disc, "pri-", pri)

    db.child("Customers").child(uid).update({
        "OrderDiscount": disc,
        "OrderPrice": pri
    })

    cart_disc = db.child("Customers").child(uid).child("OverallDiscount").get().
        val()
    cart_pri = db.child("Customers").child(uid).child("OverallProductPrice").get
        ().val()

    cart_disc = str(int(cart_disc) - (int(prqty) * (int(mp) - int(rp))))
    cart_pri = str(int(cart_pri) - (int(prqty) * int(rp)))

    print("cart_disc -",disc, "cart_pri -", pri, "\n\n")

    db.child("Customers").child(uid).update({
        "OverallDiscount": cart_disc,
        "OverallProductPrice": cart_pri
    })

    db.child("Customers").child(uid).child("Orders").child(orderid).child("
        Products").child(name).update(allProduct)
    data = {
        "DateTime": ts,
        "OrderId": orderid,
        "Status": "Placed",
        "UID": uid,
        "OverallDiscount": str(disc),
        "OverallProductPrice": str(pri)
    }
    db.child("Customers").child(uid).child("Orders").child(orderid).update(data)
    db.child("Customers").child(uid).child("Cart").child(name).remove()

    db.child("AllOrders").child(orderid).child("Products").child(name).update(
        allProduct)
    db.child("AllOrders").child(orderid).update(data)

    return True

def timestamp():
    from datetime import datetime
    dt = datetime.now()
    # ts = dt.strftime(str(dt.day))+'/'+dt.strftime(str(dt.month))+'/'+dt.
        strftime(str(dt.year))+' '+dt.strftime(str(dt.hour))+':'+dt.strftime(str
        (dt.minute))+':'+dt.strftime(str(dt.second))
    ts = "{:02d}/{:02d}/{:4d} {:02d}:{:02d}:{:02d}".format(dt.day,dt.month,dt.
        year,dt.hour,dt.minute,dt.second)
    return ts



def summary(request,uid):
    total=0
    db=firebase.database()
    allProducts = db.child("Customers").child(uid).child("Cart").get()
    finalProducts={}
    for pro in allProducts.each():
        prod=pro.val()
        # print(prod['ProductQty'])
        qty=int(prod['ProductQty'])
        stock = db.child("Warehouse").child("AllProducts").child(prod['
            ProductName']).child("Stock").get()
        stock=stock.val()
        stock=int(stock)
        st=stock-qty
        if(st>=0):
            finalProducts[pro.key()]=pro.val
            rp = db.child("Warehouse").child("AllProducts").child(prod['
                ProductName']).child("RetailPrice").get()
                rp=rp.val()
                total+=qty*int(rp)
            else:
                print("Out Of Stock")


    return render(request,"Frontend/checkout.html",{"Final":finalProducts,"Total
        ":total})

def orderplaced2(request,uid,orderid):
    disc=0
    print(orderid)
    # orderid="1888ZZZZZ"
    counter=0
    total=0
    db=firebase.database()
    allProducts = db.child("Customers").child(uid).child("Cart").get()
    finalProducts={}
    for pro in allProducts.each():
        counter+=1
        prod=pro.val()
        # print(prod['ProductQty'])
        qty=int(prod['ProductQty'])
        stock = db.child("Warehouse").child("AllProducts").child(prod['
            ProductName']).child("Stock").get()
        stock=stock.val()
        stock=int(stock)
        st=stock-qty
        if(st>=0):
            rp = db.child("Warehouse").child("AllProducts").child(prod['
                ProductName']).child("RetailPrice").get()
            rp=rp.val()
            mp = db.child("Warehouse").child("AllProducts").child(prod['
                ProductName']).child("MarketPrice").get()
            mp=mp.val()
            mp=int(mp)
            rp=int(rp)
            disc+=qty*mp-qty*rp
            newStock = db.child("Warehouse").child("AllProducts").child(prod['
                ProductName']).child("Stock").get()
            newStock=newStock.val()
            newStock=int(newStock)-qty
            db.child("Warehouse").child("AllProducts").child(prod['ProductName'
                ]).update({"Stock":str(newStock)})
            db.child("Warehouse").child(prod['Catogory']).child(prod['
                ProductName']).update({"Stock":str(newStock)})
            total+=qty*rp
            # db.child("Customers").child(uid).child("Cart").child(prod['
                ProductName']).remove()
            finalProducts[pro.key()]=pro.val()
            data={
                "DateTime":str(timestamp()),
                "OrderId":orderid,
                "OverallDiscount":str(disc),
                "OverallProductPrice":str(total),
                "Products":finalProducts,
                "Status":"Placed",
                "UID":uid
            }
            if counter==1:
                # db.child("Customers").child(uid).child("Orders").child(orderid).update(data)
                db.child("AllOrders").child(str(orderid)).set(data)
                db.child("Customers").child(str(uid)).child("Orders").child(str(orderid)).set(data)
                db.child("Customers").child(uid).child("Cart").child(prod['ProductName']).remove()
            else:
                db.child("AllOrders").child(str(orderid) ).set(data)
                db.child("Customers").child(str(uid)).child("Orders").child(str(orderid)).update(data)
                db.child("Customers").child(uid).child("Cart").child(prod['ProductName']).remove()
        else:
            print("Out Of Stock")
    return redirect('dashboardmyorders')



def proceed(request,uid):
    avl=[]
    navl=[]
    allProducts = db.child("Customers").child(uid).child("Cart").get()
    for pro in allProducts.each():
        prod=pro.val()
        # print(prod['ProductQty'])
        qty=int(prod['ProductQty'])
        stock = db.child("Warehouse").child("AllProducts").child(prod['ProductName']).child("Stock").get()
        stock=stock.val()
        stock=int(stock)
        st=stock-qty
        if(st>=0):
            avl.append(prod["ProductName"])

            # finalProducts[pro.key()]=pro.val
            # rp = db.child("Warehouse").child("AllProducts").child(prod['ProductName']).child("RetailPrice").get()
            # rp=rp.val()
            # total+=qty*int(rp)
        else:
            navl.append(prod["ProductName"])


    if len(navl)!=0 and len(avl)!=0:
        notAv = ' ,'.join([str(item) for item in navl ])
        avv = ' ,'.join([str(item) for item in avl ])
        message="{} currently out of stock would you like to place order for {}".format(notAv,avv)
        return render(request,"Frontend/index.html",{"msg":message})
    elif len(avl)==0:
        notAv = ' ,'.join([str(item) for item in navl ])

        message="{} currently out of stock.".format(notAv)
        return render(request,"Frontend/index.html",{"msg2":message})
    elif len(navl)==0:
        return redirect('SUMMARY',uid=uid)
```

Webpages/index.html : User

```
{% extends 'Frontend/base.html' %}
{% load static %}
{% block title %} AtgMart {% endblock title %}

{% block body %}

{% if msg %}
<script>
                if (confirm("{{msg}}") == true){
                  orderPlaced();
                         }
                    else {
                      window.location.replace("{% url 'INDEX' %}")

                  }
              function orderPlaced(){
                  firebase.auth().onAuthStateChanged(function(user) {
                                 if (user) {
                                       var uid = user.uid;
                  url="https://www.atgmart.com/summary/"+uid+"/";
                  window.location.replace(url)
                                 }
                           });

                      }

                  function orderId() {
                      var random4 = Math.floor(1000 + Math.random() * 9000);
                      var result = random4.toString() + "-";
                      var characters = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789';
                      var charactersLength = characters.length;
                      for ( var i = 0; i < 5; i++ ) {
                          result += characters.charAt(Math.floor(Math.random() *
                              charactersLength));
                      }
                      return result;
                  }

</script>
{% endif %}

{% if msg2 %}
<script>
alert("{{msg2}}");
window.location.replace("{% url 'INDEX' %}")
</script>
{% endif %}
<div class="wrapper">

    <div class="main-banner-slider">
        <div class="container">
            <div class="row">
                <div class="col-md-12">
                    <div class="owl-carousel offers-banner owl-theme">
                        {% for p,v in SlideShow.items %}
                        <div class="item">
                            <div class="offer-item">
                                <div class="offer-item-img">
                                    <div class="gambo-overlay"></div>
                                    <img src="{{v.ImageUrl}}" alt="{{forloop.
```

```
                                        counter}}">
                                    </div>
                                    <div class="offer-text-dt">
                                        <!-- <div class="offer-top-text-banner">
                                            <p>63% Off</p>
                                            <div class="top-text-1">Buy More & Save
                                                More</div>
                                            <span>Fresh Vegetables</span>
                                        </div> -->
                                        <!-- <a href="#" class="Offer-shop-btn hover
                                            -btn">Shop Now</a> -->
                                    </div>
                                </div>
                            </div>
                            {% endfor %}

                        </div>
                    </div>
                </div>
            </div>
        </div>


        <div class="section145">
            <div class="container">
                <div class="row">
                    <div class="col-md-12">
                        <div class="main-title-tt">
                            <div class="main-title-left">
                                <span>Shop By</span>
                                <h2>Categories</h2>
                            </div>
                        </div>
                    </div>
                    <div class="col-md-12">
                        <div class="owl-carousel cate-slider owl-theme">
                            <div class="item">
                                <a href="{% url 'All' 'Vegetables' %}" class="
                                    category-item">
                                    <div class="cate-img">
                                        <img src="{% static 'Frontend/images/
                                            category/icon-1.svg' %}" alt="fr">
                                    </div>
                                    <h4>Vegetables </h4>
                                </a>
                            </div>
                            <div class="item">
                                <a href="{% url 'All' 'Fruits' %}" class="category-
                                    item">
                                    <div class="cate-img">
                                        <img src="{% static '/Frontend/images/
                                            category/icon-2.svg' %}" alt="">
                                    </div>
                                    <h4>Fruits</h4>
                                </a>
                            </div>
                            <div class="item">
                                <a href="{% url 'All' 'Herbs' %}" class="category-
                                    item">
                                    <div class="cate-img">
                                        <img src="{% static '/Frontend/images/
                                            category/icon-3.svg' %}" alt="">
                                    </div>
                                    <h4>Herbs</h4>
                                </a>
                            </div>
                            <div class="item">
                                <a href="{% url 'All' 'Combo' %}" class="category-
                                    item">
                                    <div class="cate-img">
                                        <img src="{% static '/Frontend/images/
                                            category/icon-4.svg' %}" alt="">
                                    </div>
                                    <h4>Combos</h4>
                                </a>
                            </div>
                    </div>
                </div>
            </div>
        </div>
        <!-- Categories End -->
        <!-- Featured Products Start -->
        <div class="section145">
            <div class="container">
                <div class="row">
                    <div class="col-md-12">
                        <div class="main-title-tt">
                            <div class="main-title-left">
                                <span>For You</span>
                                <h2>Combo Deals</h2>
                            </div>
                            <a href="all/Combo/" class="see-more-btn">See All</a>
                        </div>
                    </div>
                    <div class="col-md-12">
                        <div class="owl-carousel featured-slider owl-theme" id="
                            allForYou">
                            {% for product,value in allCombo.items %}

                            <div class="item">
                                <div class="product-item">
                                    <a href="{% url 'SingleProduct' value.
                                            ProductName value.Catogory   %}" class="
                                            product-img">

                                    <!-- <a href="/single_product_view/{{value.
                                            ProductName}}/" class="product-img"> -->
                                        <img src="{{value.ImageUrl}}" alt=""style="
                                            width: 156;height: 156;">
                                        <div class="product-absolute-options">
                                            <span class="offer-badge-1">{{value.
                                                Discount}} off</span>
                                        </div>
                                    </a>
                                    <div class="product-text-dt">
                                        <p>Available<span>(Per {{ value.StockType
                                            }})</span></p>
                                        <h4>{{ value.ProductName }}</h4>
                                        <div class="product-price">  {{value.
                                            RetailPrice}} <span>  {{value.
                                            MarketPrice}}</span></div>
                                        <div class="qty-cart">
                                            <span class="cart-icon" onclick="checker
                                                ('{{value.ProductName}}')"><i class=
                                                "uil uil-shopping-cart-alt"></i></
                                                span>
                                        </div>
                                    </div>
                                </div>
                            </div>
                            {% endfor %}

                        </div>
                    </div>
                </div>
            </div>
        </div>

        <div class="section145">
            <div class="container">
                <div class="row">
                    <div class="col-md-12">
                        <div class="main-title-tt">
                            <div class="main-title-left">
                                <span>For You</span>
                                <h2>Fresh Vegetables</h2>
                            </div>
                            <a href="all/Vegetables/" class="see-more-btn">See All</
                                a>
                        </div>
                    </div>
                    <div class="col-md-12">
                        <div class="owl-carousel featured-slider owl-theme">
                            {% for product, value in allVegetables.items %}

                            <div class="item">
                                <div class="product-item">
                                    <a href="{% url 'SingleProduct' value.
                                            ProductName value.Catogory    %}" class="
                                            product-img">
                                        <img src="{{value.ImageUrl}}" alt="" style="
                                            width: 156;height: 156;">
                                        <div class="product-absolute-options">
                                            <span class="offer-badge-1">{{value.
                                                Discount}} off</span>
                                        </div>
                                    </a>
                                    <div class="product-text-dt">
                                        <p>Available<span>(Per {{ value.StockType
                                            }})</span></p>
                                        <h4>{{ value.ProductName }}</h4>
                                        <div class="product-price">  {{value.
                                            RetailPrice}} <span>  {{value.
                                            MarketPrice}}</span></div>
                                        <div class="qty-cart">
                                            <span class="cart-icon" onclick="checker
                                                ('{{value.ProductName}}')"><i class=
                                                "uil uil-shopping-cart-alt"></i></
                                                span>
                                        </div>
                                    </div>
                                </div>
                            </div>
                            {% endfor %}

                        </div>
                    </div>
                </div>
            </div>
        </div>
        <div class="section145">
            <div class="container">
                <div class="row">
                    <div class="col-md-12">
                        <div class="main-title-tt">
                            <div class="main-title-left">
                                <span>For You</span>
                                <h2>Fresh Fruits </h2>
                            </div>
                            <a href="all/Fruits/" class="see-more-btn">See All</a>
                        </div>
                    </div>
                    <div class="col-md-12">
                        <div class="owl-carousel featured-slider owl-theme">
                            {% for product, value in allFruits.items %}

                            <div class="item">
                                <div class="product-item">
                                    <a href="{% url 'SingleProduct' value.
                                            ProductName value.Catogory  %}" class="
                                            product-img">
                                        <img src="{{value.ImageUrl}}" alt=""style="
                                            width: 156;height: 156;">
                                        <div class="product-absolute-options">
                                            <span class="offer-badge-1">{{value.
                                                Discount}} off</span>
                                        </div>
                                    </a>
                                    <div class="product-text-dt">
                                        <p>Available<span>(Per {{ value.StockType
                                            }})</span></p>
                                        <h4>{{ value.ProductName }}</h4>
                                        <div class="product-price">  {{value.
                                            RetailPrice}} <span>  {{value.
                                            MarketPrice}}</span></div>
                                        <div class="qty-cart">
                                            <span class="cart-icon" onclick="checker
                                                ('{{value.ProductName}}')"><i class=
                                                "uil uil-shopping-cart-alt"></i></
                                                span>
                                        </div>
                                    </div>
                                </div>
                            </div>
                            {% endfor %}


                        </div>
                    </div>
                </div>
            </div>
        </div>

        <div class="section145">
```

```
        <div class="container">
            <div class="row">
                <div class="col-md-12">
                    <div class="main-title-tt">
                        <div class="main-title-left">
                            <span>For You</span>
                            <h2>Fresh Herbs</h2>
                        </div>
                        <a href="all/Herbs/" class="see-more-btn">See All</a>
                    </div>
                </div>
                <div class="col-md-12">
                    <div class="owl-carousel featured-slider owl-theme">
                        {% for product,value in allHerbs.items %}

                        <div class="item">
                            <div class="product-item">
                                <a href="{% url 'SingleProduct' value.
                                    ProductName value.Catogory    %}" class="
                                    product-img">
                                    <img src="{{value.ImageUrl}}" alt=""style="
                                        width: 156;height: 156;">
                                    <div class="product-absolute-options">
                                        <span class="offer-badge-1">{{value.
                                            Discount}} off</span>
                                    </div>
                                </a>
                                <div class="product-text-dt">
                                    <p>Available<span>(Per {{ value.StockType
                                        }})</span></p>
                                    <h4>{{ value.ProductName }}</h4>
                                    <div class="product-price">  {{value.
                                        RetailPrice}} <span>  {{value.
                                        MarketPrice}}</span></div>
                                    <div class="qty-cart">
                                        <span class="cart-icon" onclick="checker
                                            ('{{value.ProductName}}')"><i class=
                                            "uil uil-shopping-cart-alt"></i></
                                            span>
                                    </div>
                                </div>
                            </div>
                        </div>
                        {% endfor %}


                    </div>
                </div>
            </div>
        </div>
    </div>
</div>
{% load static %}
<footer class="footer">
  <div class="footer-first-row">
    <div class="container">
      <div class="row">
        <div class="col-md-6 col-sm-6">
          <ul class="call-email-alt">
            <li><a href="#" class="callemail"><i class="uil uil-dialpad-alt"></i
                >+91-74001-4773</a></li>
```

```
              <li><a href="#" class="callemail"><i class="uil uil-envelope-alt"></
                  i>anytimegrocery0@gmail.com</a></li>
            </ul>
          </div>
          <div class="col-md-6 col-sm-6">
            <div class="social-links-footer">
              <ul>
                <li><a href="https://www.facebook.com/anytimegroceries/"><i class=
                    "fab fa-facebook-f"></i></a></li>
                                        <li><a href="https://www.instagram.com/atgmart/"><i
                                            class="fab fa-instagram"></i></a></li>
              </ul>
            </div>
          </div>
        </div>
      </div>
    </div>
    <div class="footer-second-row">
      <div class="container">
        <div class="row">
          <div class="col-lg-3 col-md-6 col-sm-6">
            <div class="second-row-item">
              <h4>Categories</h4>
              <ul>
                <li><a href="{% url 'All' 'Fruits' %}">Fruits</a></li>
                <li><a href="{% url 'All' 'Vegetables' %}">Vegetables</a></li>
                <li><a href="{% url 'All' 'Herbs' %}">Herbs</a></li>
                <li><a href="{% url 'All' 'Combo' %}">Combo's</a></li>
              </ul>
            </div>
          </div>
          <div class="col-lg-3 col-md-6 col-sm-6">
            <div class="second-row-item">
              <h4>Useful Links</h4>
              <ul>

                <li><a href="{% url 'about_us' %}">About US</a></li>
                <li><a href="{% url 'offers' %}">Offers</a></li>
                <li><a href="{% url 'contactus' %}">Contact Us</a></li>
              </ul>
            </div>
          </div>
          <div class="col-lg-3 col-md-6 col-sm-6">
            <div class="second-row-item">
              <h4>Delivering In</h4>
              <ul id="cities">
                                        <script>
                                        var query = firebase.database().ref().child("
                                            Pincodes");

                                        query.on("child_added",snap =>{
                                        var city = snap.val();
                                        console.log(city);
                                        var html = "<li><a href='#'>"+city+"</a></li>";
                                        console.log(html);
                                        $("#cities").append(html)

                                        });
                                        </script>
              </ul>
            </div>
          </div>
          <div class="col-lg-3 col-md-6 col-sm-6">
            <div class="second-row-item-app">
              <h4>Download App</h4>
              <ul>
                <li><a href="#"><img class="download-btn" src="{% static 'Frontend
                    /images/download-1.svg' %}" alt=""></a></li>
              </ul>
            </div>
            <div class="second-row-item-payment">
              <h4>Payment Method</h4>
              <div class="footer-payments">
                <ul id="paypal-gateway" class="financial-institutes">
                  <li class="financial-institutes__logo">
                  <img alt="Visa" title="Visa" src="{% static 'Frontend/images/
                      footer-icons/pyicon-6.svg' %}">
                  </li>
                </ul>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
  <div class="footer-last-row">
    <div class="container">
      <div class="row">
        <div class="col-md-12">
          <div class="footer-bottom-links">
            <ul>
              <li><a href="{% url 'about_us' %}">About</a></li>
              <li><a href="{% url 'contactus' %}">Contact</a></li>
              <li><a href="{% url 'privacypolicy' %}">Privacy Policy</a></li>
              <li><a href="{% url 'termandconditions' %}">Term & Conditions</a
                  ></li>
              <li><a href="{% url 'refundreturn' %}">Refund & Return Policy</a
                  ></li>
            </ul>
          </div>
          <div class="copyright-text">
            <i class="uil uil-copyright"></i>Copyright 2020 <b>Any Time
                Groceries</b> . All rights reserved
          </div>
        </div>
      </div>
    </div>
  </div>
</footer>
<!-- Footer End -->

<!-- Javascripts -->
<script src="{% static 'Frontend/js/jquery-3.3.1.min.js' %}"></script>
<link href = "https://code.jquery.com/ui/1.12.0/themes/ui-lightness/jquery-ui.
    css" rel = "stylesheet">
<script src = "https://code.jquery.com/jquery-1.12.1.js"></script>
<script src = "https://code.jquery.com/ui/1.12.1/jquery-ui.js"></script>
<script src="{% static 'Frontend/js/jquery.autocomplete.min.js' %}"></script>
<link href="{% static 'Frontend/css/styles.css' %}" rel="stylesheet">
<script src="{% static 'Frontend/js/jquery.countdown.min.js' %}"></script>
<script src="{% static 'Frontend/vendor/bootstrap/js/bootstrap.bundle.min.js'
```

```
    %}"></script>
<script src="{% static 'Frontend/vendor/OwlCarousel/owl.carousel.js' %}"></
    script>
<script src="{% static 'Frontend/vendor/semantic/semantic.min.js' %}"></script>
<script src="{% static 'Frontend/js/jquery.countdown.min.js' %}"></script>
<script src="{% static 'Frontend/js/custom.js' %}"></script>
<script src="{% static 'Frontend/js/offset_overlay.js' %}"></script>
<script src="{% static 'Frontend/js/night-mode.js' %}"></script>
</body>
</html>

{% endblock body %}
```

Views.py : Admin

```
from django.shortcuts import render,redirect
import pyrebase
import datetime
from django.utils.timezone import utc
from django.contrib import auth as dauth
from django.contrib.auth import authenticate,get_user,login,logout
from django.core.files.storage import FileSystemStorage
status=False
firebaseConfig = {
    "apiKey": "AIzaSyC6F7FprRkBSoziZjlPnJec6dQGzpjZmQo",
    "authDomain": "newatg-11a13.firebaseapp.com",
    "databaseURL": "https://newatg-11a13.firebaseio.com",
    "projectId": "newatg-11a13",
    "storageBucket": "newatg-11a13.appspot.com",
    "messagingSenderId": "45402315445",
    "appId": "1:45402315445:web:7bd5a80c1b31cfb1f4fcfc",
    "measurementId": "G-9NZL1LL0YB"
  }
firebase = pyrebase.initialize_app(firebaseConfig)



storage = firebase.storage()
db=firebase.database()


def viewProduct(request,Name):
    db = firebase.database()
    product=db.child("Warehouse").child("AllProducts").child(Name).get()

    pro={product.key():product.val()}
    # for order in product.each():
    #     pro[order.key()]=order.val()
    #     print(order.key())

    return render(request,"Admin/product_view.html",{"Product":pro})

def home(request):
    # if status==True:
    #     return render(request,"Admin/index.html",{"msg":"ok"})
    # else:
    return render(request,"Admin/index.html")


def products(request):
    # db = firebase.database()
    # # allProducts=db.child("Warehouse").child("AllProducts").child("Apple").
        get()
    # # print(allProducts.val())
    allProducts2={}
    # for product in allProducts.each():
    #     allProducts2[product.key()]=product.val()
    return render(request,"Admin/products.html",{"All":allProducts2})


def add(request):

    return render(request,"Admin/add_product.html")

def customers(request):
    # db = firebase.database()
    # allProducts=db.child("Customers").get()

    allCust={}
    # for cust in allProducts.each():
    #     allCust[cust.key()]=cust.val()


    return render(request,"Admin/customers.html",{"All":allCust})

def orders(request):
    global status

    allOrders={}



    return render(request,"Admin/orders.html",{"All":allOrders})

def categories(request):
    return render(request,"Admin/category.html")

def addcategories(request):


    return render(request,"Admin/add_category.html")

def posts(request):


    return render(request,"Admin/posts.html")

def addnewposts(request):


    return render(request,"Admin/add_post.html")

def postcategories(request):


    return render(request,"Admin/post_categories.html")

def post_tags(request):
    global status

    return render(request,"Admin/post_tags.html")
```

```


def addlocations(request):
    global status

    return render(request,"Admin/add_location.html")

def areas(request):


    return render(request,"Admin/areas.html")

def addareas(request):

    return render(request,"Admin/add_area.html")

def shops(request):


    return render(request,"Admin/shops.html",{"Status":status})

def addshops(request):


    return render(request,"Admin/add_shop.html",{"Status":status})

def offers(request):

    return render(request,"Admin/offers.html",{"Status":status})

def addoffers(request):


    return render(request,"Admin/add_offer.html",{"Status":status})

def pages(request):


    return render(request,"Admin/pages.html")

def menu(request):

    return render(request,"Admin/menu.html")

def addmenu(request):

    return render(request,"Admin/add_menu.html")

def updater(request):


    return render(request,"Admin/updater.html")

def generalsettings(request):


    return render(request,"Admin/general_setting.html")

def paymentsettings(request):


    return render(request,"Admin/payment_setting.html")

def emailsettings(request):


    return render(request,"Admin/email_setting.html")


def reports(request):


    return render(request,"Admin/reports.html",{"Status":status})

def login2(request):

    emailaddress = request.POST.get('email')
    password = request.POST.get('password1')
    # auth=firebase.auth()
    if emailaddress==None:
        return render(request,"Admin/sign_in.html")
    if password==None:
        return render(request,"Admin/sign_in.html")


    # try:
    #     # global status

    #     # user = auth.sign_in_with_email_and_password(emailaddress, password)
    #     # # print(user)
    #     # email=user['email']
    #     # print(email)
    if(emailaddress=="anytimegrocery0@gmail.com"):
        user = authenticate(request,username='Admin1', password=password)
        if user is not None:
            login(request,user)
            print(user)
            return redirect('HOME')

        else:
            message="Invalide Email OR Password"
            return render(request,"Admin/sign_in.html",{"msg":message})
    else:
        # global status
        message="Invalide Email OR Password"
        return render(request,"Admin/sign_in.html",{"msg":message})


def editprofile(request):

    return render(request,"Admin/edit_profile.html")

def changepassword(request):
    auth=firebase.auth()
    r=auth.send_password_reset_email("anytimegrocery0@gmail.com")
    print(r)
    msg="Password Reset Email Has Been Set"
    return render(request,"Admin/sign_in.html",{'msg':msg})
```

```
# db.child("users").child("Morty").update({"name": "Mortiest Morty"})

def updateProduct(request):
    Category = request.POST.get('category')
    name=request.POST.get('name')
    Discription=request.POST.get('disc')
    StockType=request.POST.get('type')
    Stock=request.POST.get('stock')
    MarketPrice=request.POST.get('mp')
    RetailPrice=request.POST.get('rp')
    Discount=request.POST.get('discount')
    # print("Discount::"+Discount)
    now=datetime.datetime.now()
    date_time = now.strftime("%d/%m/%Y, %H:%M:%S")

    updateData={
        'Catogory':Category,
        'Discount':Discount,
        'Discription':Discription,
        'MarketPrice':MarketPrice,
        'ProductName':name,
        'RetailPrice':RetailPrice,
        'Stock':Stock,
        'StockType':StockType,
        'DateTime':str(date_time)
    }
    print(updateData)
    db = firebase.database()
    db.child("Warehouse").child(Category).child(str(name)).update(updateData)
    db.child("Warehouse").child("AllProducts").child(str(name)).update(
        updateData)
    return redirect('Product')



def newLocation(request):

    name=request.POST.get('name')
    pincode=request.POST.get('pincode')
    db = firebase.database()
    db.child("Pincodes").update({pincode:name})
    return render(request,"Admin/add_location.html")







def Orderview(request,OrderId):

    db=firebase.database()
    Products=db.child("AllOrders").child(str(OrderId)).child("Products").get()
    total=db.child("AllOrders").child(str(OrderId)).child("OverallProductPrice")
        .get()
    Status=db.child("AllOrders").child(str(OrderId)).child("Status").get()
    date=db.child("AllOrders").child(str(OrderId)).child("DateTime").get()

    User=db.child("AllOrders").child(str(OrderId)).child("UID").get()
    Username=db.child("Customers").child(User.val()).child("Name").get()
    Add=db.child("Customers").child(User.val()).child("Address").get()


    allPro={}
    for Pro in Products.each():
        allPro[Pro.key()]=Pro.val()

    return render(request,"Admin/order_view.html",{"OrderId":OrderId,"Products":
        allPro,"Total":total.val(),"Status2":Status.val(),
    "Name":Username.val(),"Add":Add.val,"Date":date.val()})



def OrderEdit(request,OrderId):

    db=firebase.database()
    Products=db.child("AllOrders").child(str(OrderId)).child("Products").get()
    total=db.child("AllOrders").child(str(OrderId)).child("OverallProductPrice")
        .get()
    Status=db.child("AllOrders").child(str(OrderId)).child("Status").get()
    User=db.child("AllOrders").child(str(OrderId)).child("UID").get()
    Username=db.child("Customers").child(User.val()).child("Name").get()
    Add=db.child("Customers").child(User.val()).child("Address").get()
    call=db.child("Customers").child(User.val()).child("Mobile").get()

    date=db.child("AllOrders").child(str(OrderId)).child("DateTime").get()

    allPro={}
    for Pro in Products.each():
        allPro[Pro.key()]=Pro.val()

    return render(request,"Admin/order_edit.html",{"OrderId":OrderId,"Products":
        allPro,"Total":total.val(),
    "Status2":Status.val(),"Name":Username.val(),"Add":Add.val,"uid":User.val(),
        "Date":date.val(),"Call":call.val()})

def logout2(request):
    logout(request)
    return render(request,"Admin/sign_in.html")


def slides(request):

    return render(request,"Admin/allSlids.html")

def slideView(request,Name):

    db = firebase.database()
    product=db.child("SlideShow").child(Name).get()
    pro={product.key():product.val()}
    # for order in product.each():
    return render(request,"Admin/slideView.html",{"Slide":pro})

def EditLoc(request,pin,loc):
    loc=loc.replace('%',' ')
    db = firebase.database()
    contex={pin:loc}
    # except:
    #     print("No Location found")
    db.child("Pincodes").update({pin:loc})
    return render(request,"Admin/editLocation.html",{"loc":contex})


```

```
def addNewSlide(request):
    # if request.method=='POST':
    #     myfile=request.POST.get('myFile', False)
    #     print(request.POST['name'])
    #     print(myfile)
    return render(request,"Admin/addSlide.html")

def AddSlideFinal(request):
    # print("myfile")
    # if request.method=='POST':

    #     myfile=request.FILES['myFile']

    #     print(request.POST['name'])
    #     print(myfile)
    #     fs=FileSystemStorage()
    #     fileName=fs.save(myfile.name,myfile)
    #     url=fs.url(fileName)
    #     print(url)

    #     img = Image.open('/your image path/image.jpg') # image extension *.png
        ,*.jpg
    #     new_width  = 400
    #     new_height = 200
    #     img = img.resize((new_width, new_height), Image.ANTIALIAS)
        # img.save('.png')
    return redirect("Slide")


def locations(request):
    usernew=dauth.get_user_model
    print(usernew)
    if request.user.is_authenticated:
        print("Logged in")
    else:
        print("Not logged in")
    db = firebase.database()
    allLocations=db.child("Pincodes").get()
    locations={}

    for order in allLocations.each():
        # print(order.key+"Loc:"+order.val())
        locations[order.key()]=order.val()


    return render(request,"Admin/locations.html",{"Locations":locations})

def edit(request,Name,Cat):
    db = firebase.database()
    product=db.child("Warehouse").child("AllProducts").child(Name).get()
    pro={product.key():product.val()}

    return render(request,"Admin/editProduct.html",{"Product":pro,"Name":Name})

def DeleteLoc(request,pin,loc):
    db = firebase.database()
    db.child("Pincodes").child(pin).remove()
    return render(request,"Admin/locations.html")

def addProductNew(request):
    Category = request.POST.get('category')
    name=request.POST.get('name')
    Discription=request.POST.get('disc')
    StockType=request.POST.get('type')
    Stock=request.POST.get('stock')
    MarketPrice=request.POST.get('mp')
    RetailPrice=request.POST.get('rp')
    Discount=request.POST.get('discount')
    # print("Discount::"+Discount)
    now=datetime.datetime.now()
    date_time = now.strftime("%d/%m/%Y, %H:%M:%S")
    updateData={
        'Catogory':Category,
        'Discount':Discount,
        'Discription':Discription,
        'MarketPrice':MarketPrice,
        'ProductName':name,
        'RetailPrice':RetailPrice,
        'Stock':Stock,
        'StockType':StockType,
        'DateTime':str(date_time)
    }
    print(updateData)
    # if(db.child("Warehouse").child("AllProducts").child(str(name)).update(
        updateData))
    db = firebase.database()
    db.child("Warehouse").child(Category).child(str(name)).update(updateData)
    db.child("Warehouse").child("AllProducts").child(str(name)).update(
        updateData)
    return redirect("Product")
```

Webpages/index.html : Admin

```
{% extends 'Admin/base.html' %}

{% load static %}
{% block title %}Dashboard{% endblock title %}

{% block body %}
<div id="layoutSidenav_content">

  <main>
    <div class="container-fluid">
      <h2 class="mt-30 page-title">Dashboard</h2>
      <ol class="breadcrumb mb-30">
        <li class="breadcrumb-item active">Dashboard</li>
      </ol>
      <div class="row">
        <div class="col-xl-3 col-md-6">
          <div class="dashboard-report-card purple">
            <div class="card-content">
              <span class="card-title">Order Pending</span>
              <span id="Pending" class="card-count">0</span>
            </div>
            <div class="card-media">
              <i class="fab fa-rev"></i>
            </div>
          </div>
        </div>
        <div class="col-xl-3 col-md-6">
          <div class="dashboard-report-card red">
```

```
            <div class="card-content">
              <span class="card-title">Order Cancel</span>
              <span id="Cancel" class="card-count">0</span>
            </div>
            <div class="card-media">
              <i class="far fa-times-circle"></i>
            </div>
          </div>
        </div>
        <div class="col-xl-3 col-md-6">
          <div class="dashboard-report-card info">
            <div class="card-content">
              <span class="card-title">Order Process</span>
              <span id="Process" class="card-count">0</span>
            </div>
            <div class="card-media">
              <i class="fas fa-sync-alt rpt_icon"></i>
            </div>
          </div>
        </div>
        <div class="col-xl-3 col-md-6">
          <div class="dashboard-report-card success">
            <div class="card-content">
              <span class="card-title">Today Income</span>
              <span id="income" class="card-count"> 0 </span>
            </div>
            <div class="card-media">
              <i class="fas fa-money-bill rpt_icon"></i>
            </div>
          </div>
        </div>
        <div class="col-xl-12 col-md-12">
          <div class="card card-static-1 mb-30">
            <div class="card-body">
              <div id="earningGraph"></div>
            </div>
          </div>
        </div>
        <div class="col-xl-12 col-md-12">
          <div class="card card-static-2 mb-30">
            <div class="card-title-2">
              <h4>Recent Orders</h4>
              <a href="{% url 'Order' %}" class="view-btn hover-btn">View All</a
                >
            </div>
            <div class="card-body-table">
              <div class="table-responsive">
                <table id="table" class="table ucp-table table-hover">
                  <thead>
                    <tr>
                      <th style="width:130px">Number</th>
                      <th style="width:130px">Order ID</th>
                      <th style="width:200px">Date</th>
                      <th style="width:130px">Status</th>
                      <th style="width:130px">Total</th>
                      <th style="width:100px">Action</th>
                    </tr>
                  </thead>
                  <tbody id="tbody">

                  </tbody>
```

```
                        </table>
                    </div>
                </div>
            </div>
        </div>
    </div>
  </div>
 </main>
 <footer class="py-4 bg-footer mt-auto">
    <div class="container-fluid">
        <div class="d-flex align-items-center justify-content-between small">
            <div class="text-muted-1">  <b>Any Time Groceries</b>.</div>
            <div class="footer-links">
                <a href="http://gambolthemes.net/html-items/gambo_supermarket_demo/
                    privacy_policy.html">Privacy Policy</a>
                <a href="http://gambolthemes.net/html-items/gambo_supermarket_demo/
                    term_and_conditions.html">Terms &amp; Conditions</a>
            </div>
        </div>
    </div>
 </footer>
</div>
</div>

<script src="{% static 'Admin/vendor/bootstrap/js/bootstrap.bundle.min.js' %}"
    ></script>
<script src="{% static 'Admin/vendor/chart/highcharts.js' %}"></script>
<script src="{% static 'Admin/vendor/chart/exporting.js' %}"></script>
<script src="{% static 'Admin/vendor/chart/export-data.js' %}"></script>
<script src="{% static 'Admin/vendor/chart/accessibility.js' %}"></script>
<script src="{% static 'Admin/js/scripts.js' %}"></script>
<script src="{% static 'Admin/js/chart.js' %}" ></script>
</body>
</html>


{% endblock body %}
```

Chapter 7

System Testing

System testing is a level of software testing where a complete and integrated software is tested. The purpose of this test is to evaluate the system's compliance with the specified requirements.

7.1 Test Cases and Test Results

Test ID	**Test Case Title**	**Test Condition**	**System Behavior**	**Expected Result**
T01	Clicking and viewing product info	An android application	On clicking, product info is shown	Product info on clicking on any product
T02	Add product to cart	A registered email address or phone number	Product added into the cart and can view all details.	Product added into cart successfully
T03	Order products	A registered email address or phone number and selected produc	Confirm order and show order Id, status and payable amount	Order placed successfully and details of order displayed.
T04	Cancel order	Already placed order	Cancel the order and confirm it with user	Order cancelled successfully
T05	Refer and Earn	Referral code	Share the link and code with friend and earn a coupon	Earn a coupon

T06	Add, edit and delete product (for admin)	Admin login credentials	Add new products, edit details of previously added products and delete products	Products added successfully. Edited details of product successfully. Deleted products successfully
T07	View orders and set status	Admin login credentials	View all orders and set their status	Viewed all products successfully. Status set successfully

7.2 Sample of a Test Case

Title: Clicking and viewing product info
Description: A user should be able to successfully view product information such as price, market price, discount, availability of product and description.

Precondition: An android application.
Assumption: a supported android application is being used.

Test Steps:

1. Open the application
2. On home page most popular products are shown
3. Select the product shown on home page or
4. Select product from category
5. Click on the product whose info is needed to display

Expected Result: Product info is shown on clicking any product.
Actual Result: On clicking product info is shown successfully.

Title: Add product to cart
Description: User will be able to add the product in its cart and can buy it.

Precondition: A registered email address or phone number

Assumption: a supported android application is being used.

Test Steps:

1. Open the application
2. Login with valid credentials
3. Select the product which user wants to add
4. The page shows product details along with options to 'Add to Cart' and 'Buy Now'
5. Click on Add to cart or Buy Now
6. The product is added in the cart and message is displayed for the same

Expected Result: Product should get added into the cart and view all details.
Actual Result: Product is added into the cart and and see the details of product in cart.

Title: Order products
Description: The user should be able to place the order for the required products.
Precondition: A registered email address or phone number
*Assumption:*a supported android application is being used.

Test Steps:

1. Open the application
2. Add required product in the cart or click on Buy Now
3. Click on My Cart. (The page shows all products in the cart with options to increase/decrease quantity and delete option. Apply coupon is also provided if have any along with total savings and payable amount)

4. Click on continue
5. Confirm the order by clicking on yes
6. Order gets successfully placed

Expected Result: User should get notification for the order placed and can see the order details like order id, status and amount etc. In order history.
Actual Result: Oder is placed successfully and details are seen in the order history successfully

Title: Cancel order
Description: The user should be able to cancel the order which was placed by him/her for any reason it needs to get cancelled
*Precondition:*A registered email address or phone number.
*Assumption:*Already placed order by user.

Test Steps:

1. Open the application
2. Go to order history
3. Click on the order which is already placed and need to cancel
4. The order details are shown along with option to cancel it
5. Click on cancel
6. A dialogue box appears to confirm the cancellation
7. Click on yes option

Expected Result: User should get notification for the order placed and can see the order details like order id, status and amount etc. in order history.
Actual Result: Oder is placed successfully and details are seen in the order history successfully.

Title: Refer and Earn
Description: The user can refer his/her friends or family members to use the application for their shopping and as a reward he/she can get the reward in the form of coupons which can be applied on his/her order of above 50 Rs. Also the referred user on installing the application and using referral code also gets the coupon.

*Precondition:*The user should be logged in with proper credentials and has referral code in his/her Refer and Earn section.

*Assumption:*a supported android application is being used, social media accounts like whatsapp, instagram or gmail acoount etc.or messaging application.

Test Steps:

1. Open application
2. Go to Refer and Earn section
3. Click on 'Invite Friends' option
4. Available Options to share the code and link are displayed
5. Select the preferable one and send the invitation
6. After referred user installs app using link and uses referral code while registering, User will get one coupon and can see its details in 'My Coupons' section.

Expected Result: User should get the coupon and able to see its details in My Coupon section
Actual Result: User gets the coupon and is able to view its details in My Coupon section.

Title: Add, edit and delete products(admin)
Description: The admin should be able to add,edit and delete products which will reflect in both application and website.

Precondition: A registered email address of admin.

Test Steps:

1. Login in website via admin login
2. Go to products
3. Select add product option
4. Fill all details related to product ad upload image
5. Click on add and product is added successfully
6. Click on all products
7. Products details with option for edit and delete id displayed
8. Click on edit to edit product details
9. Click on delete to delete product

Expected Result: Product should get added, edited and deleted successfully
Actual Result: Product is added successfully. Products details are edited successfully and deleted product successfully.

Title: View all orders and set status
Description: The admin should be able to view all orders customer has placed through android application. Admin should be able to set status of order like confirmed, placed, dispatched, delivered etc.

*Precondition:*A registered email address or phone number.

Assumption: Admin has some orders already placed by customers

Test Steps:

1. Login in website with admin credentials
2. Go to orders
3. Click on all orders
4. Select order placed by customer
5. Go to status and change status according to availability

Expected Result: All orders should be viewed successfully and status of order should be set successfully .
Actual Result: All orders placed by customers are viewed successfully and status of order is set successfully.

7.2.1 Software Quality Attributes

Availability-1 : The system shall be available to users all the time.
Availability-2 : The system shall always have something to function and always pop up error messages in case of component failure.
Effificiency-1 : The system makes users shopping experience 75% better than other applications and sites.
Effificiency-2 : The system shall provide the right tools to support all its features.
caption subcaption subfigure babel,blindtext

Chapter 8

Screenshots of Project

8.1 Android Application: User

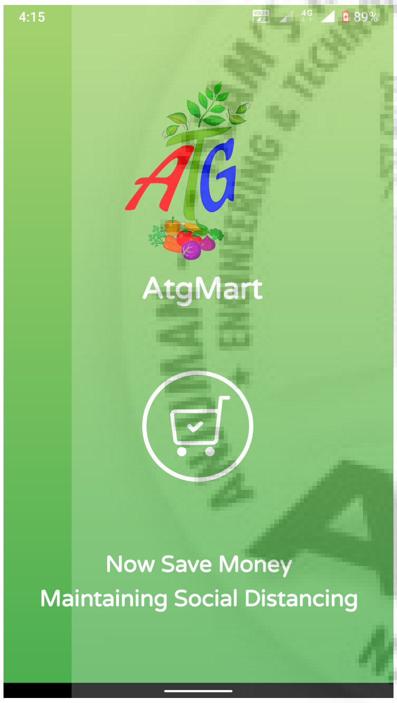


Figure 8.1: Landing Page

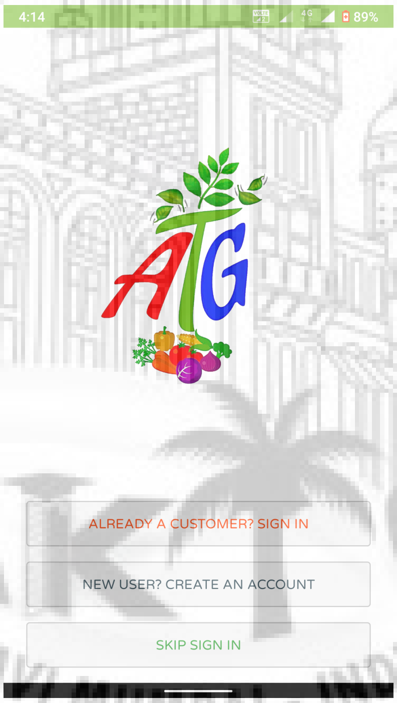
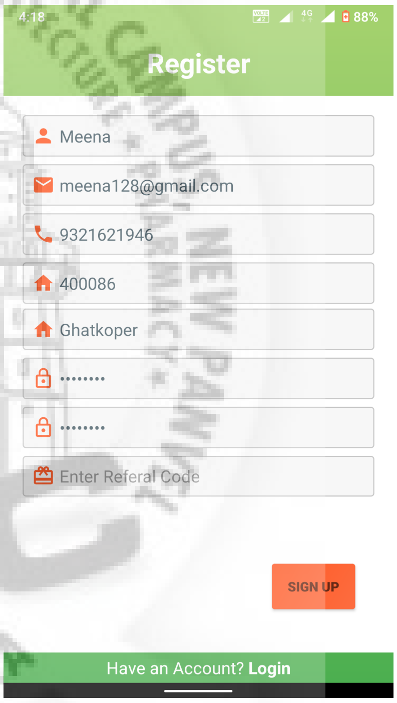


Figure 8.2: Sign In,Sign Up Skip



Meena
meena128@gmail.com
9321621946
400086
Ghatkoper
••••••••
••••••••
Enter Referal Code

Figure 8.3: Register/Sign Up

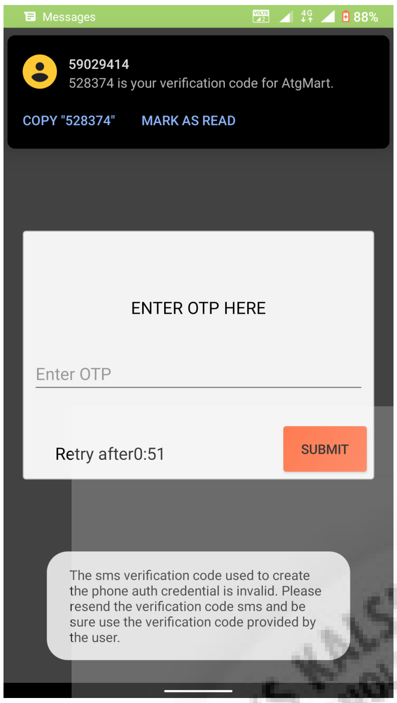

Messages
59029414
528374 is your verification code for AtgMart.
COPY "528374"
MARK AS READ
ENTER OTP HERE
Enter OTP
Retry after0:51
SUBMIT
The sms verification code used to create the phone auth credential is invalid. Please resend the verification code sms and be sure use the verification code provided by the user.

Figure 8.4: OTP Verification

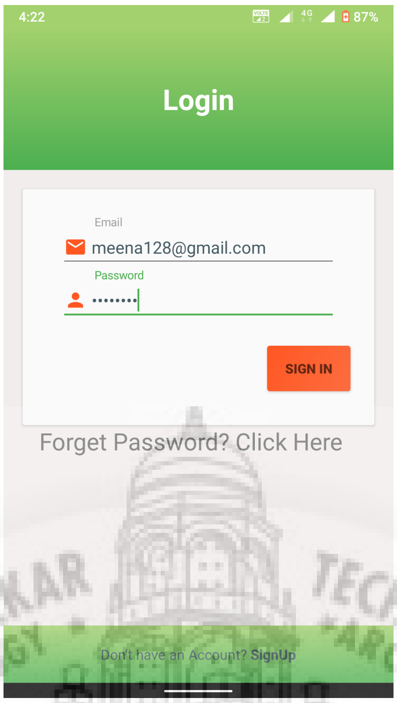


Figure 8.5: Login

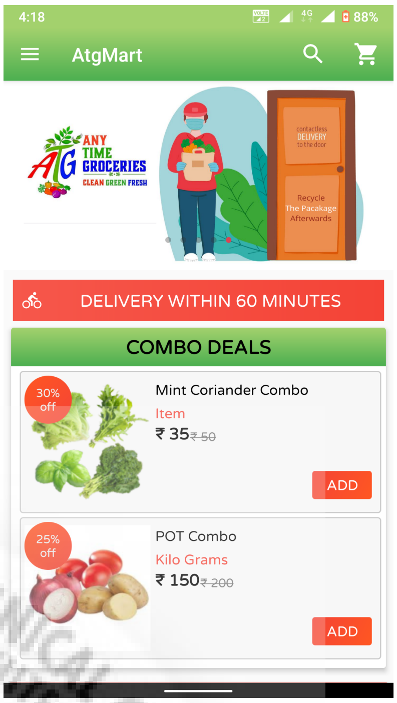


Figure 8.6: Main Page

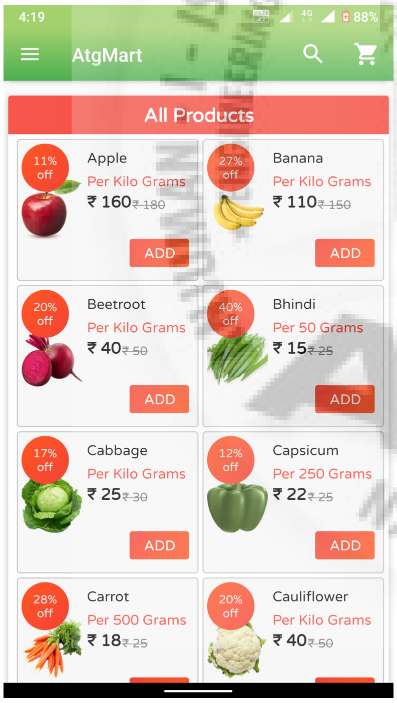

AtgMart
All Products
Apple Per Kilo Grams ₹ 160 ₹ 180 ADD
Banana Per Kilo Grams ₹ 110 ₹ 150 ADD
Beetroot Per Kilo Grams ₹ 40 ₹ 50 ADD
Bhindi Per 50 Grams ₹ 15 ₹ 25 ADD
Cabbage Per Kilo Grams ₹ 25 ₹ 30 ADD
Capsicum Per 250 Grams ₹ 22 ₹ 25 ADD
Carrot Per 500 Grams ₹ 18 ₹ 25
Cauliflower Per Kilo Grams ₹ 40 ₹ 50

Figure 8.7: All Products

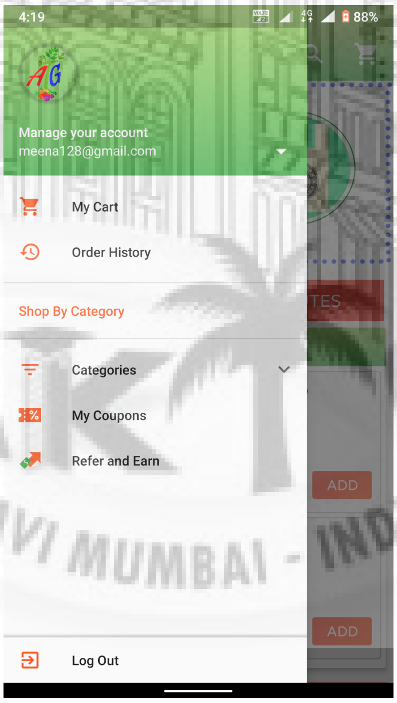


Figure 8.8: Sidebar

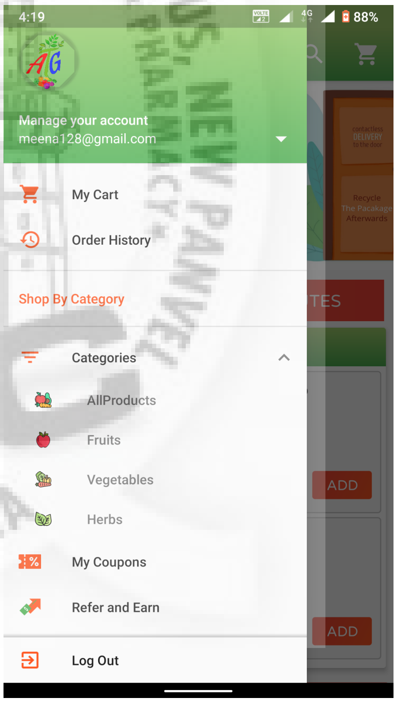
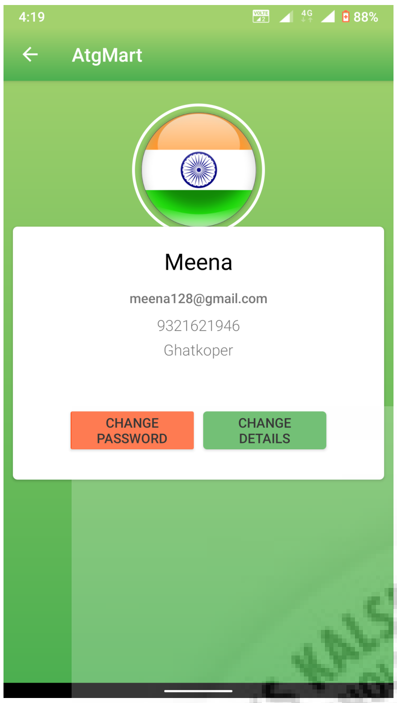


Figure 8.9: Categories


4:19 88%

AtgMart

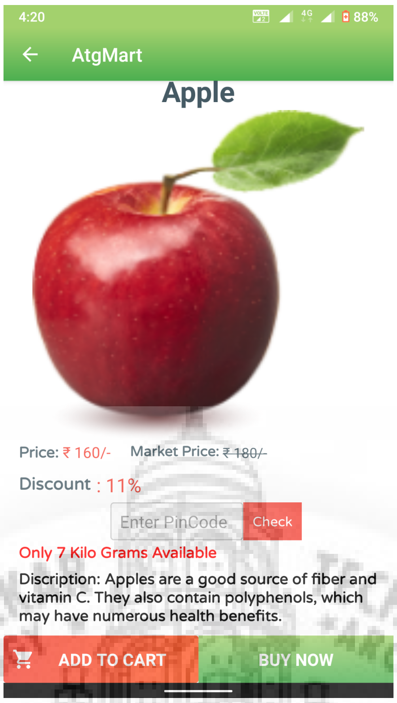
Meena

meena128@gmail.com

9321621946

Ghatkoper

Figure 8.10: My Profile


4:20 88%

AtgMart

Apple

Price: ₹ 160/- Market Price: ~~₹ 180/-~~

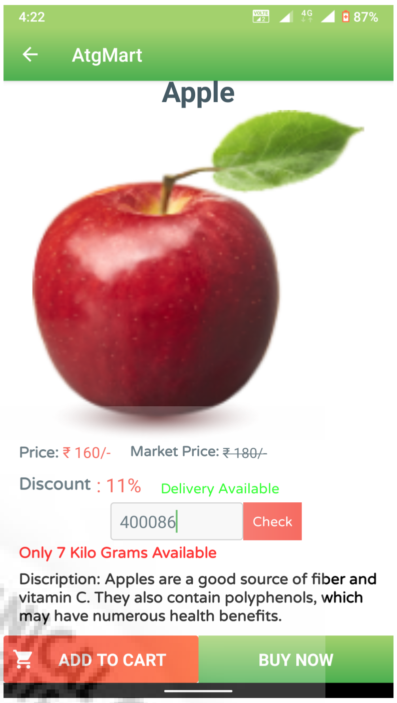
Discount : 11%

Enter PinCode Check

Only 7 Kilo Grams Available

Discription: Apples are a good source of fiber and vitamin C. They also contain polyphenols, which may have numerous health benefits.

Figure 8.11: Product Details


4:22 87%

AtgMart

Apple

Price: ₹ 160/- Market Price: ~~₹ 180/-~~

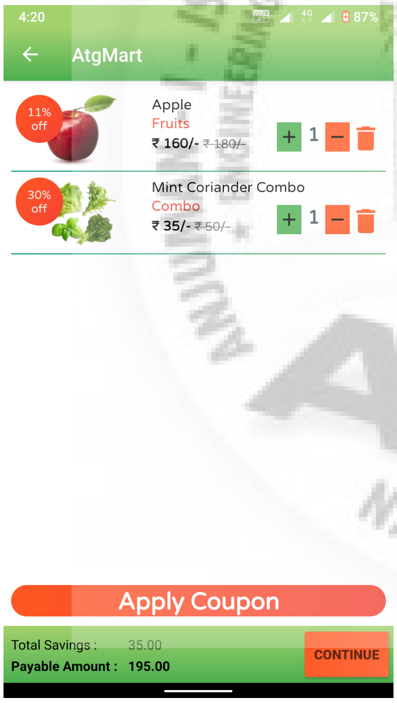
Discount : 11% Delivery Available

400086 Check

Only 7 Kilo Grams Available

Discription: Apples are a good source of fiber and vitamin C. They also contain polyphenols, which may have numerous health benefits.

Figure 8.12: Product Availability


4:20 87%

AtgMart

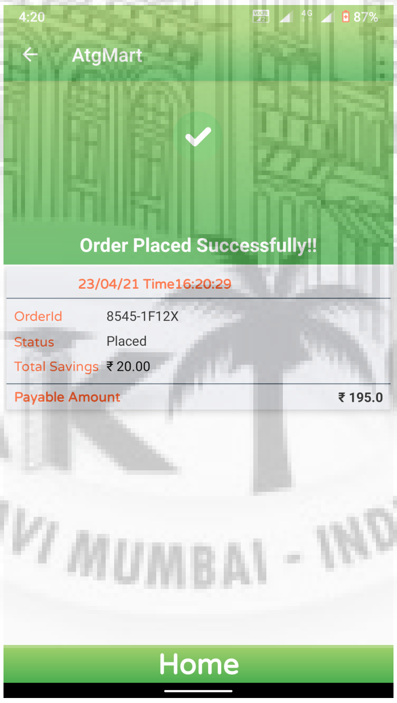
11% off Apple Fruits ₹ 160/- ~~₹ 180/-~~ 1

30% off Mint Coriander Combo Combo ₹ 35/- ~~₹ 50/-~~ 1

Total Savings : 35.00

Payable Amount : 195.00

Figure 8.13: My Cart


4:20 87%

AtgMart

Order Placed Successfully!!

23/04/21 Time16:20:29

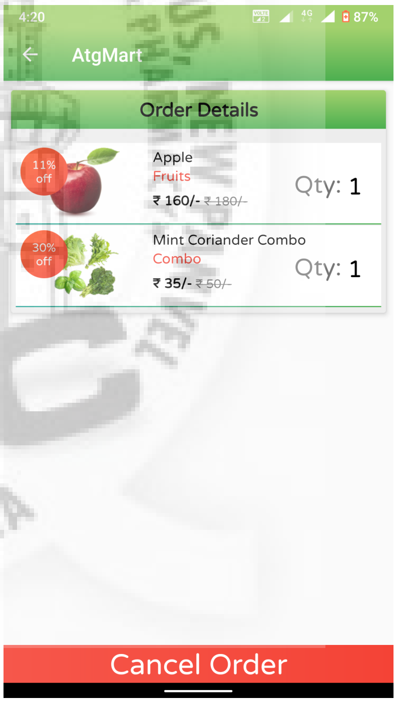
OrderId 8545-1F12X

Status Placed

Total Savings ₹ 20.00

Payable Amount ₹ 195.0

Figure 8.14: Order Placed


4:20 87%

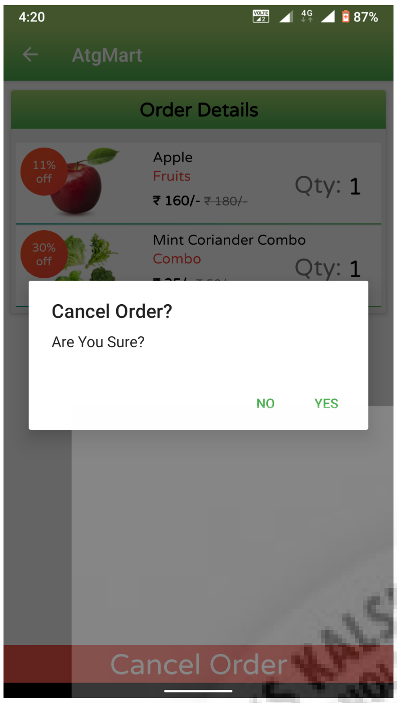
AtgMart

Order Details

11% off Apple Fruits ₹ 160/- ~~₹ 180/-~~ Qty: 1

30% off Mint Coriander Combo Combo ₹ 35/- ~~₹ 50/-~~ Qty: 1

Figure 8.15: Order Details

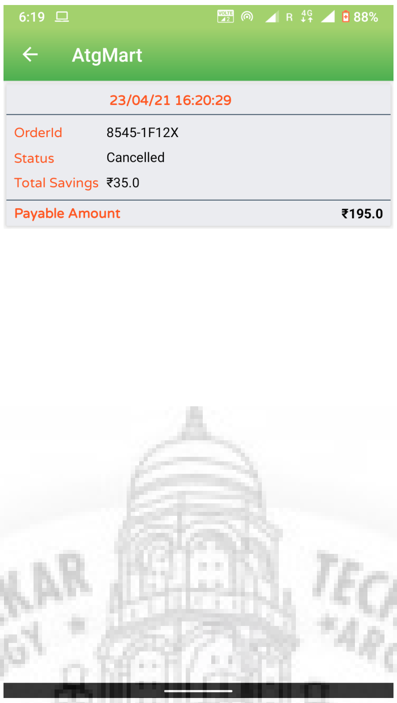


Cancel Order?

Are You Sure?

NO YES

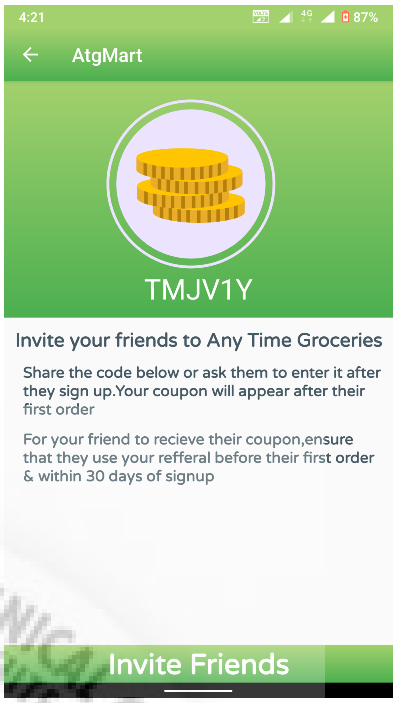
Figure 8.16: Cancel Order



23/04/21 16:20:29

OrderId	8545-1F12X
Status	Cancelled
Total Savings	₹35.0
Payable Amount	₹195.0

Figure 8.17: Order History



Invite your friends to Any Time Groceries

Share the code below or ask them to enter it after they sign up.Your coupon will appear after their first order

For your friend to recieve their coupon,ensure that they use your refferal before their first order & within 30 days of signup

Figure 8.18: Refer Earn

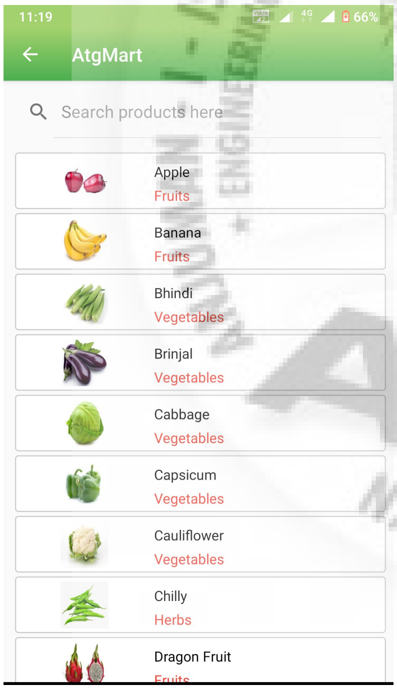
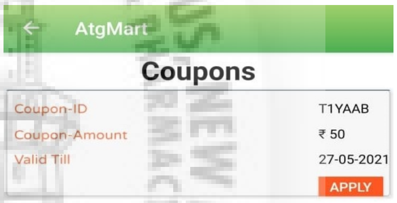


Figure 8.19: Search



Coupons

Coupon-ID	T1YAAB
Coupon-Amount	₹ 50
Valid Till	27-05-2021

APPLY

Figure 8.20: My Coupons

8.2 Admin Panel: Warehouse Management

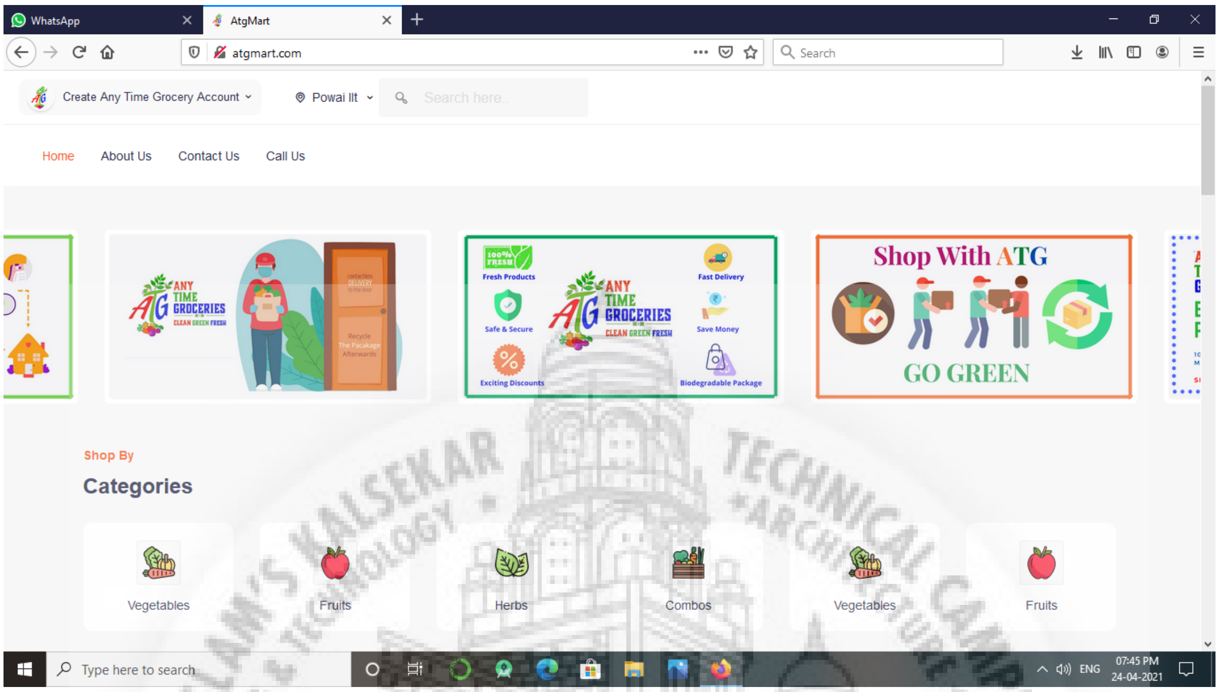


Figure 8.21: Landing Page

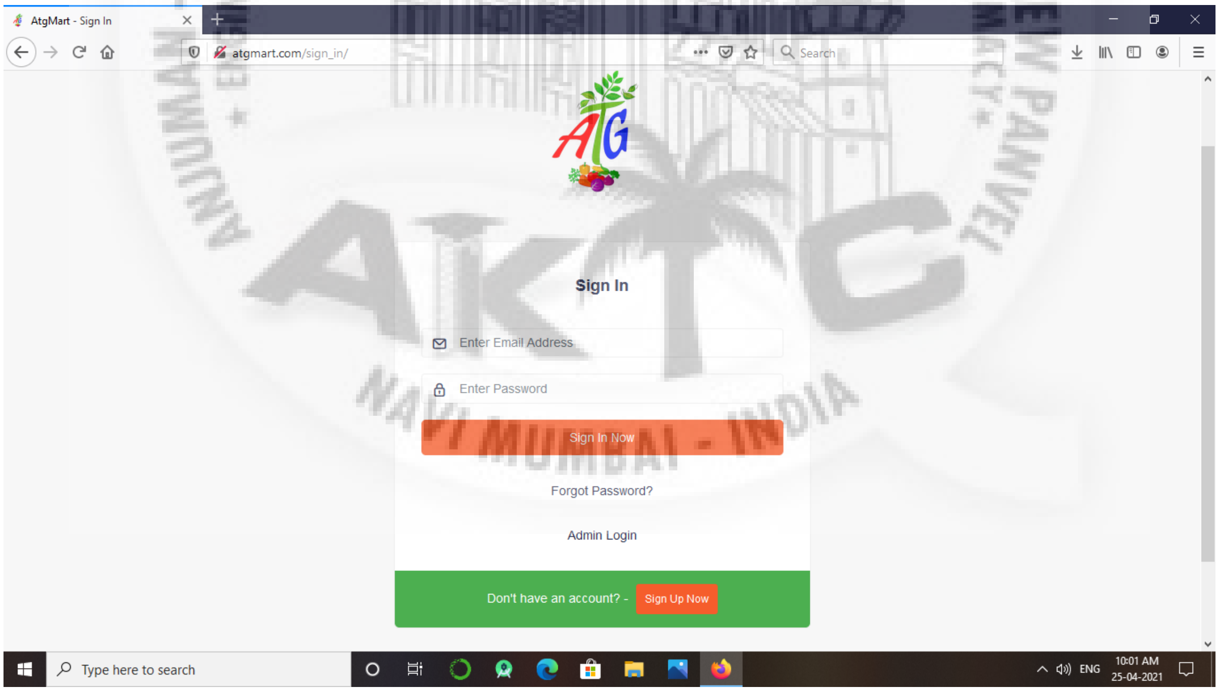


Figure 8.22: Login Page

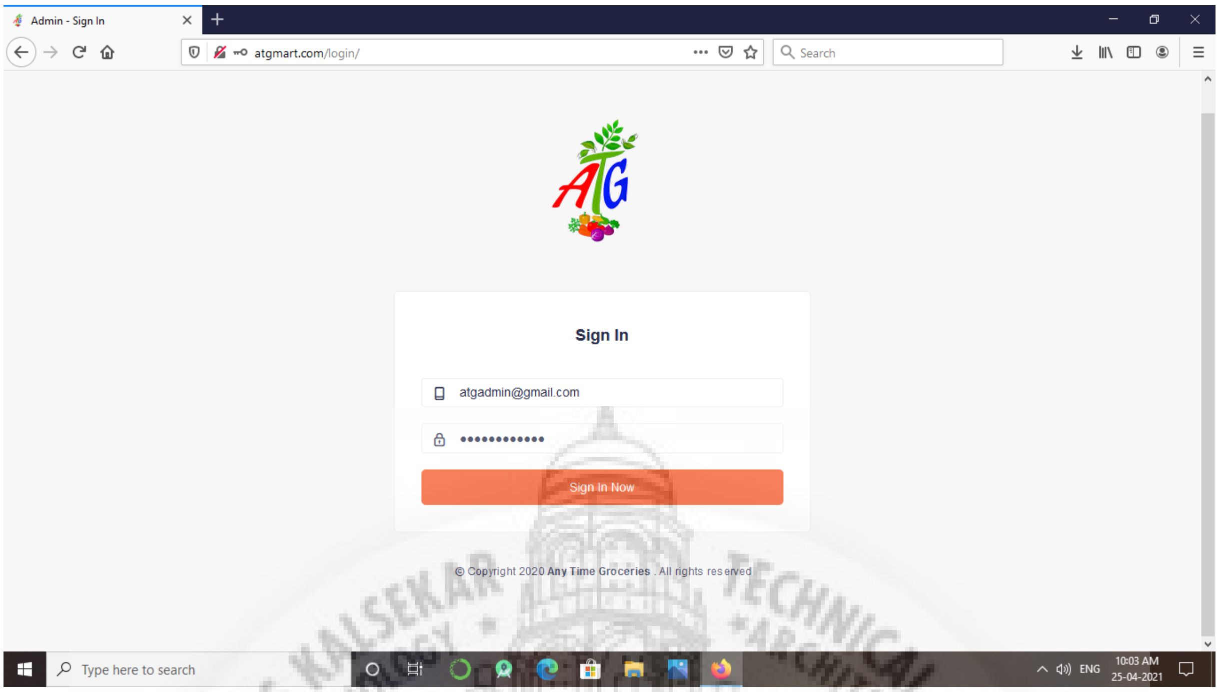


Figure 8.23: Admin Login

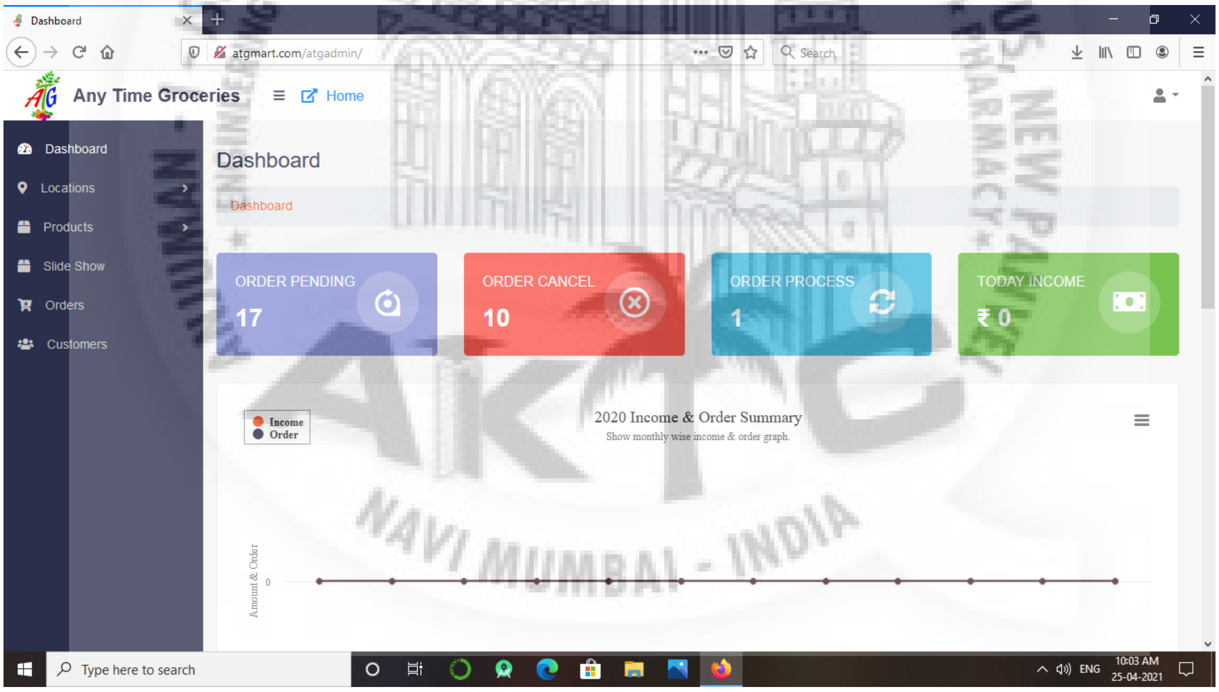
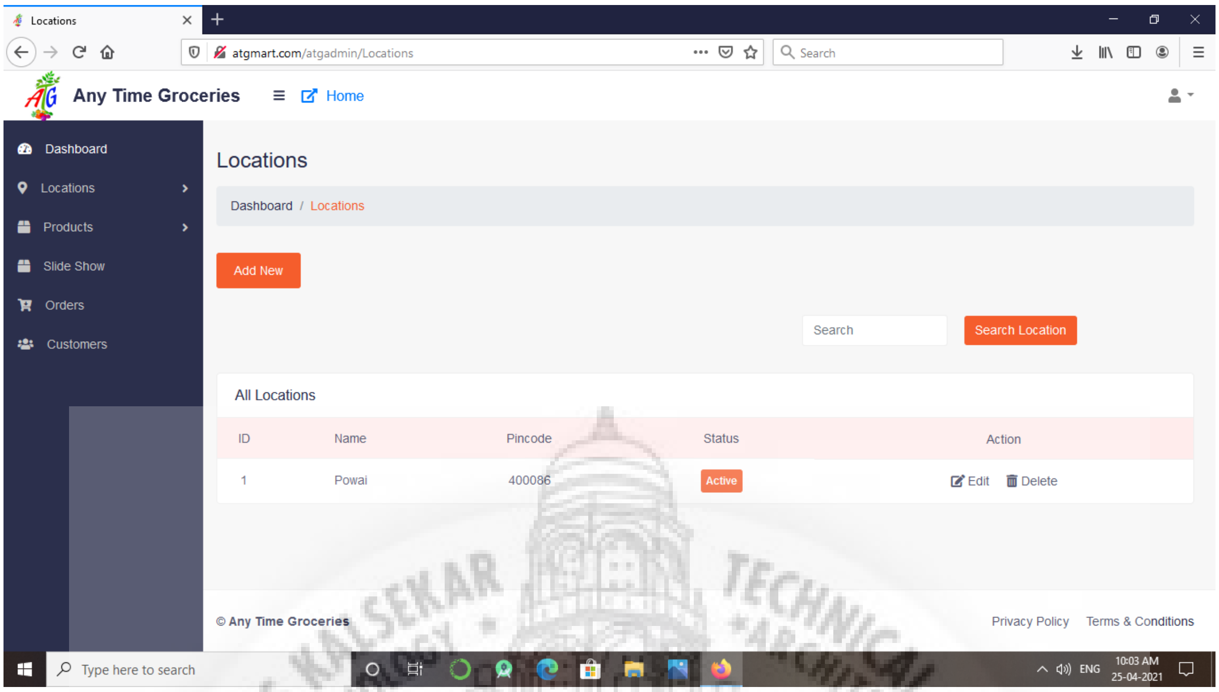


Figure 8.24: Admin Dashboard



Locations

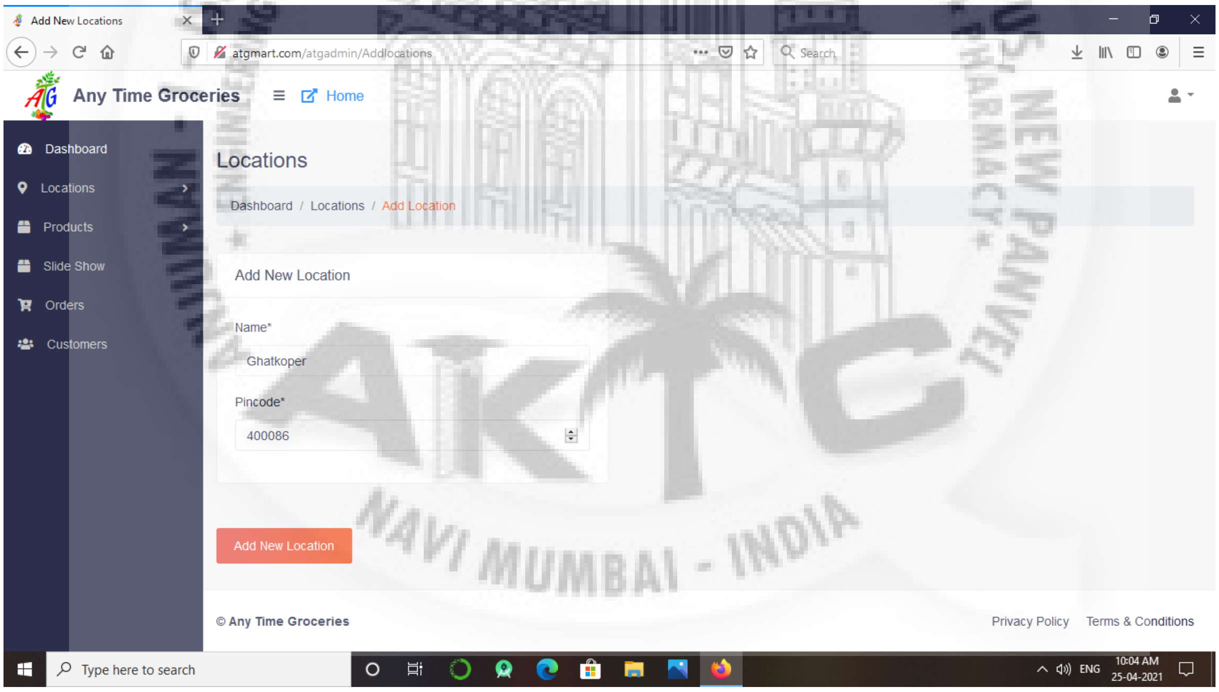
Dashboard / Locations

All Locations

ID	Name	Pincode	Status	Action
1	Powai	400086	Active	Edit Delete

© Any Time Groceries

Figure 8.25: All Locations



Locations

Dashboard / Locations / Add Location

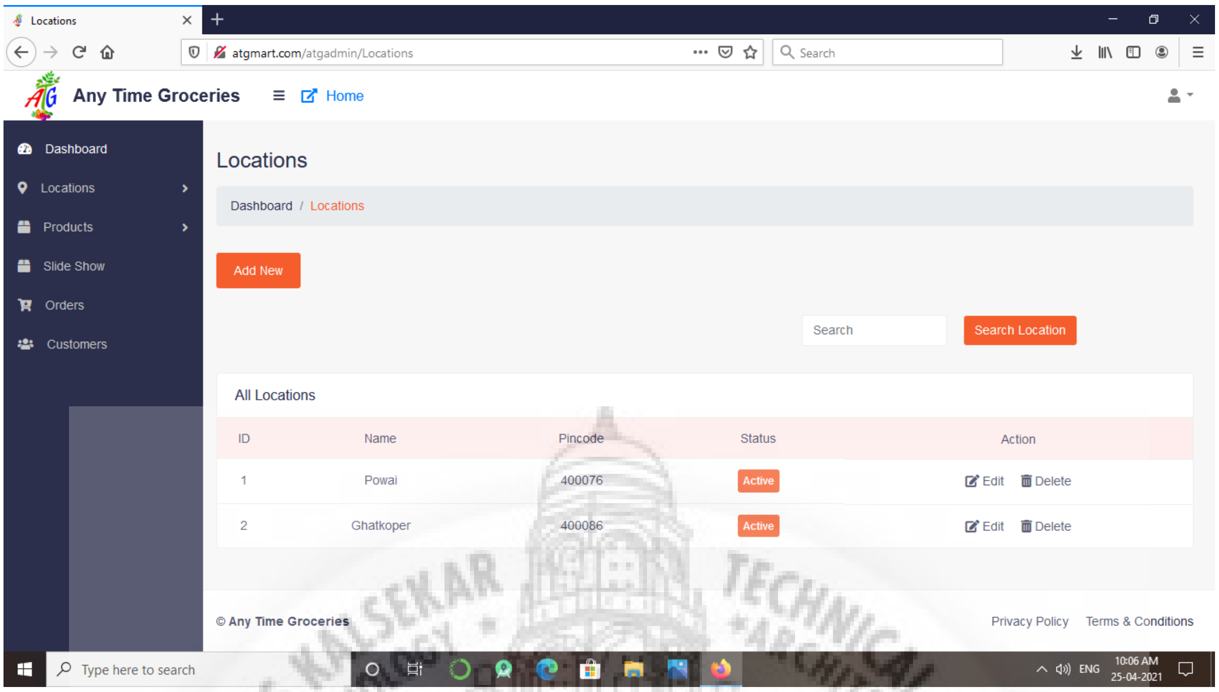
Add New Location

Name*: Ghatkoper

Pincode*: 400086

© Any Time Groceries

Figure 8.26: Add New Location



Locations

Dashboard / Locations

All Locations

ID	Name	Pincode	Status	Action
1	Powai	400076	Active	Edit Delete
2	Ghatkoper	400086	Active	Edit Delete

Figure 8.27: All locations with newly added location

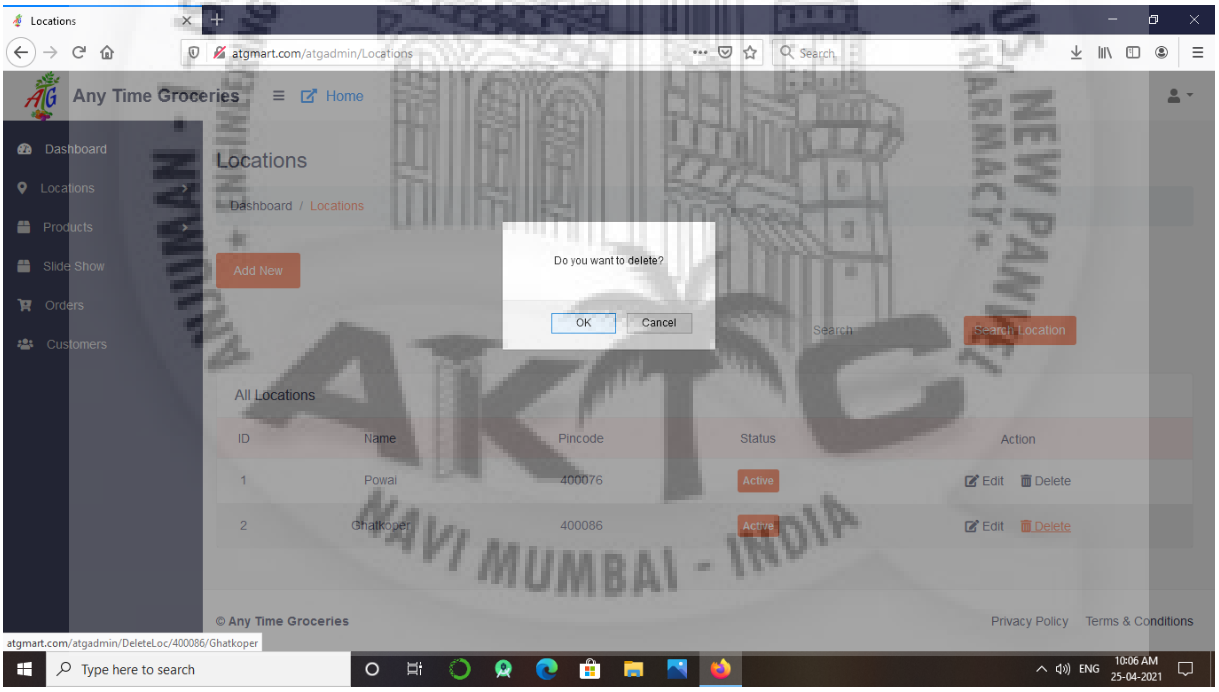
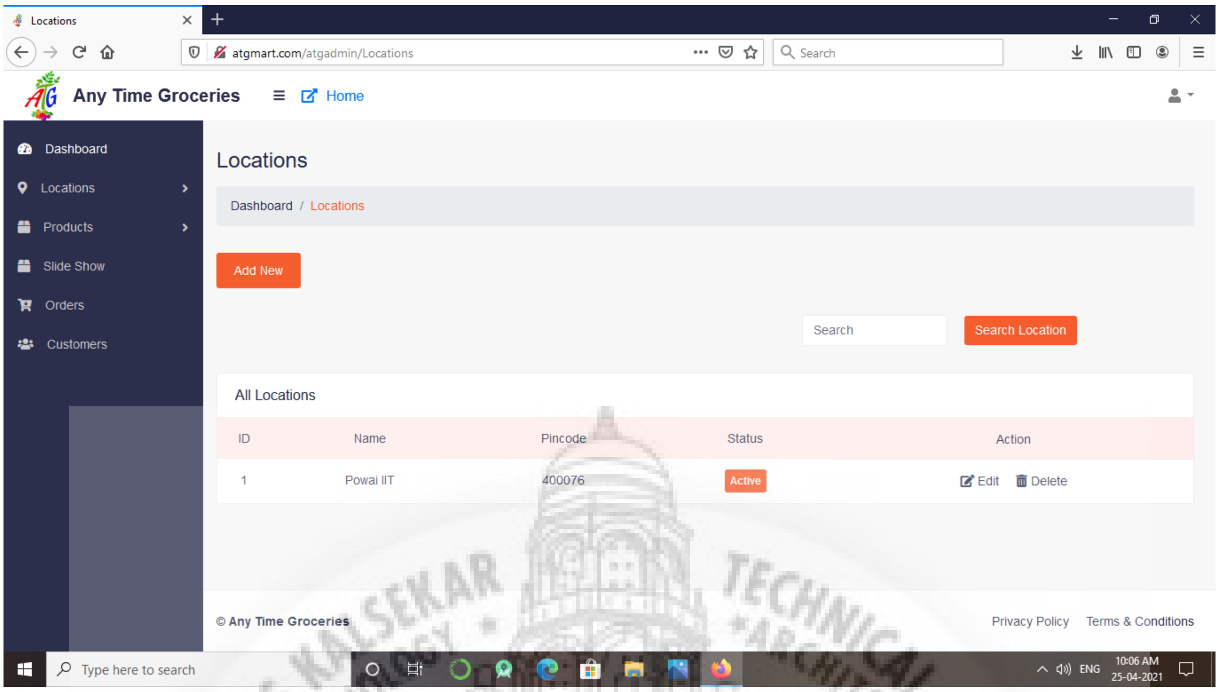


Figure 8.28: Delete Location



Locations

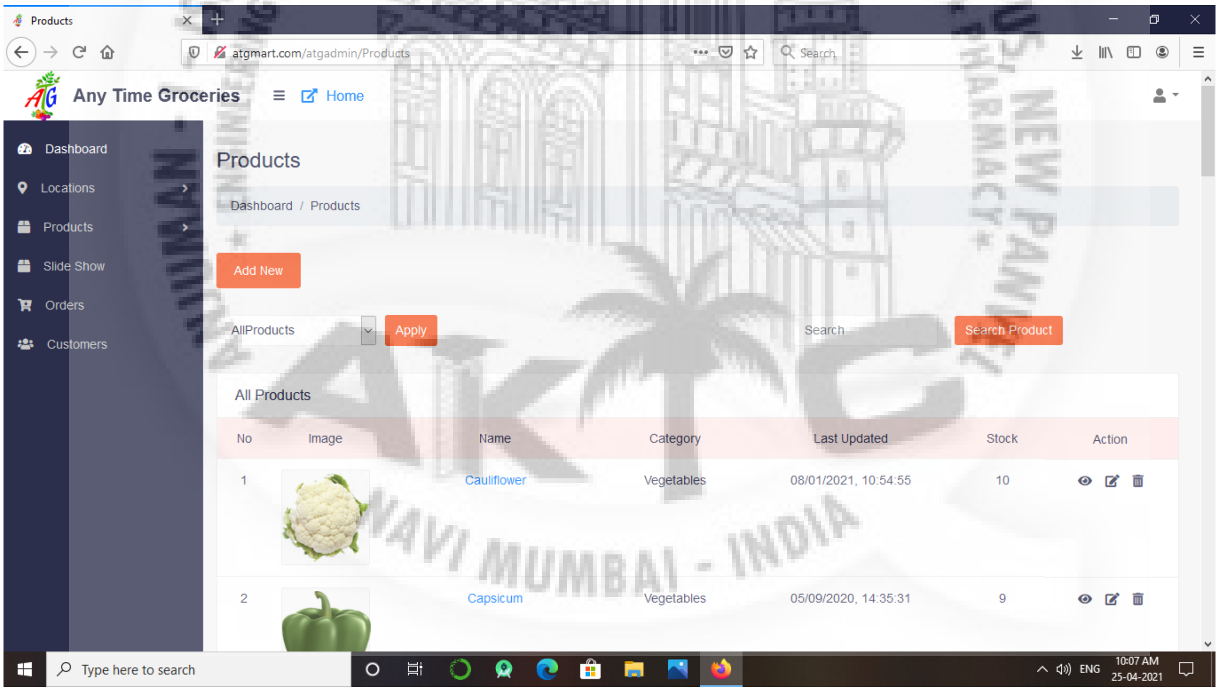
Dashboard / Locations

All Locations

ID	Name	Pincode	Status	Action
1	Powai IIT	400076	Active	Edit Delete

© Any Time Groceries Privacy Policy Terms & Conditions

Figure 8.29: Deleted Successfully. Showing All remaining locations



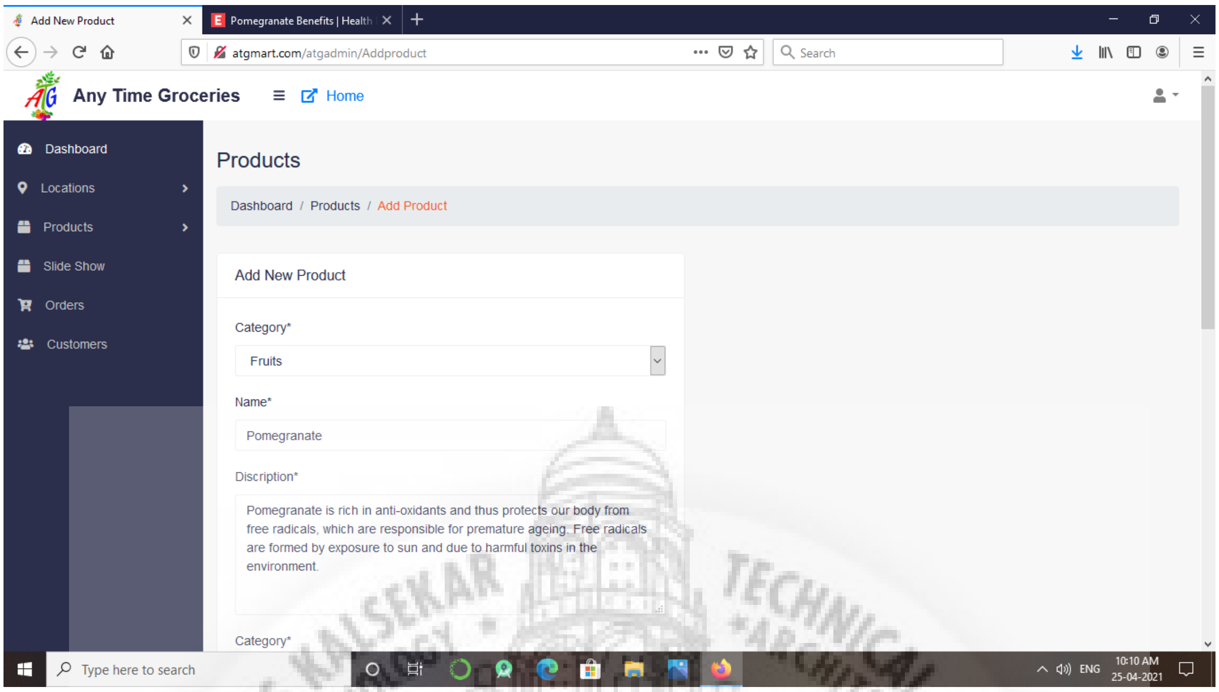
Products

Dashboard / Products

All Products

No	Image	Name	Category	Last Updated	Stock	Action
1		Cauliflower	Vegetables	08/01/2021, 10:54:55	10	
2		Capsicum	Vegetables	05/09/2020, 14:35:31	9	

Figure 8.30: All Products



Products

Dashboard / Products / Add Product

Add New Product

Category*

Fruits

Name*

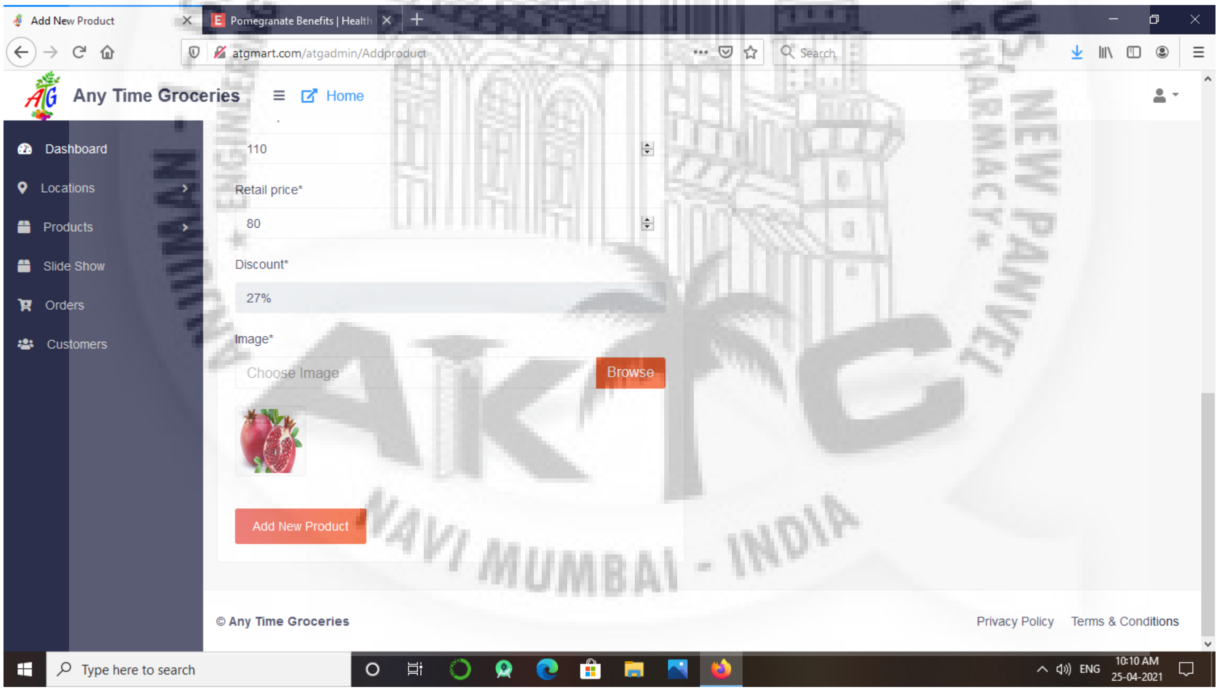
Pomegranate

Discription*

Pomegranate is rich in anti-oxidants and thus protects our body from free radicals, which are responsible for premature ageing. Free radicals are formed by exposure to sun and due to harmful toxins in the environment.

Category*

Figure 8.31: Add new products



110

Retail price*

80

Discount*

27%

Image*

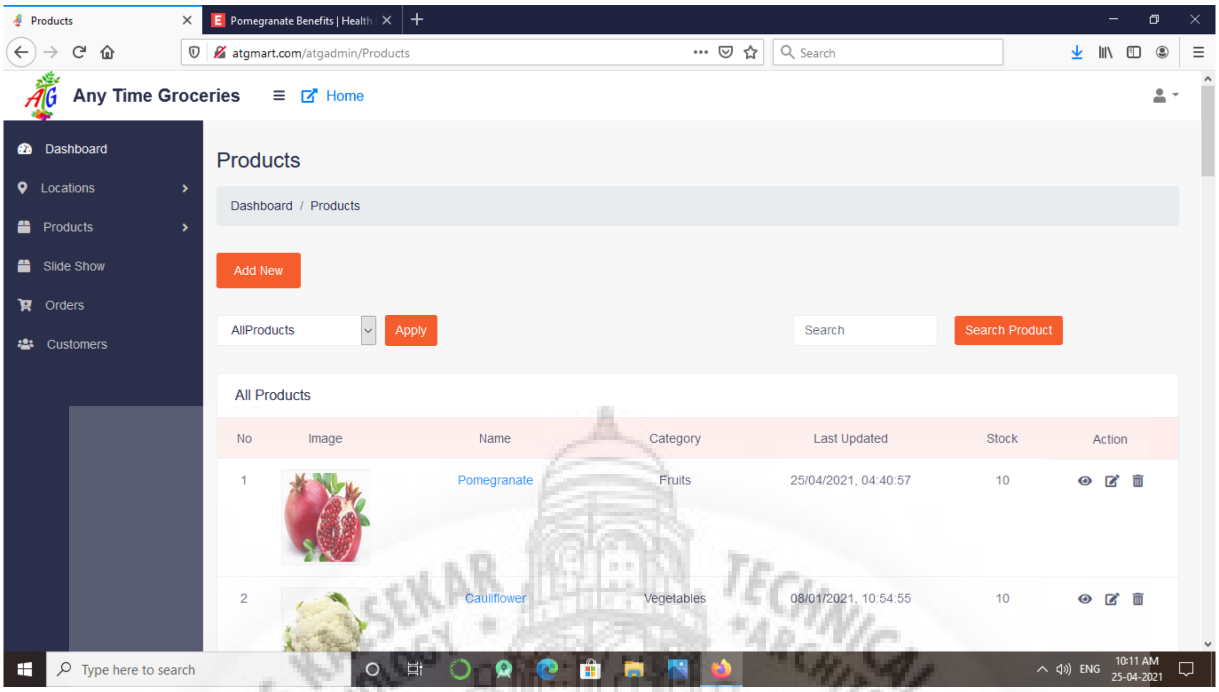
Choose Image Browse

Add New Product

© Any Time Groceries

Privacy Policy Terms & Conditions

Figure 8.32: Adding new product



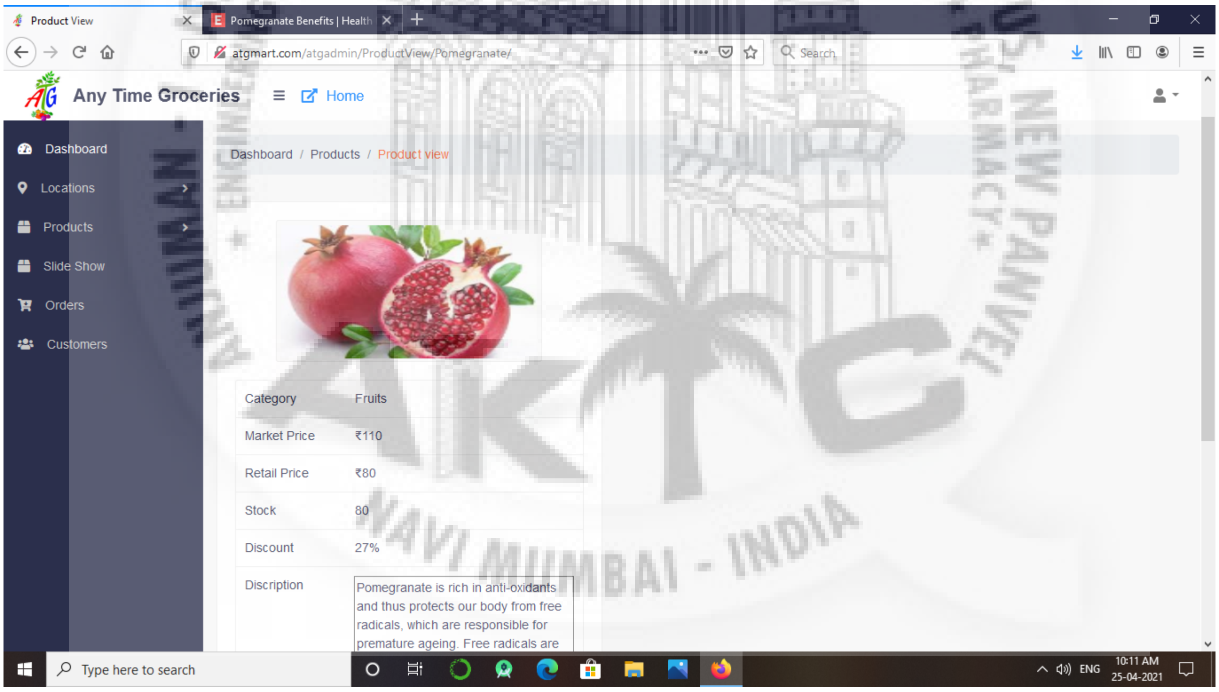
Products

Dashboard / Products

All Products

No	Image	Name	Category	Last Updated	Stock	Action
1		Pomegranate	Fruits	25/04/2021, 04:40:57	10	
2		Cauliflower	Vegetables	08/01/2021, 10:54:55	10	

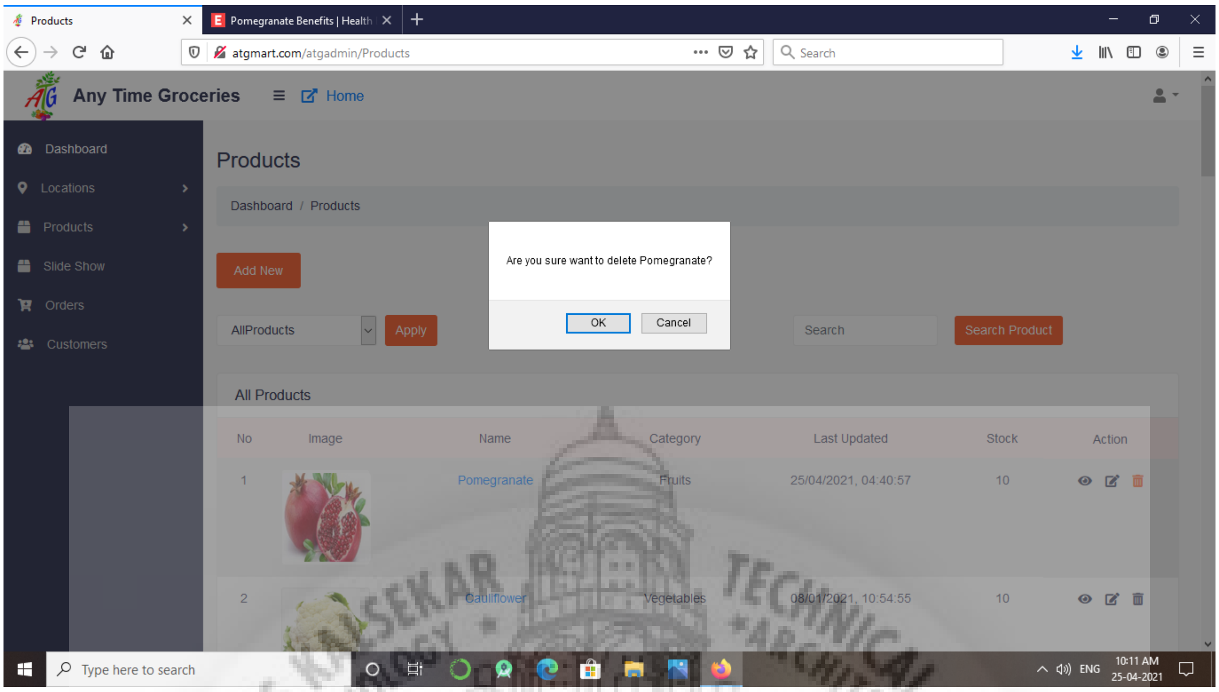
Figure 8.33: Product added successfully



Dashboard / Products / Product view

Category	Fruits
Market Price	₹110
Retail Price	₹80
Stock	80
Discount	27%
Discription	Pomegranate is rich in anti-oxidants and thus protects our body from free radicals, which are responsible for premature ageing. Free radicals are

Figure 8.34: Product details



Products

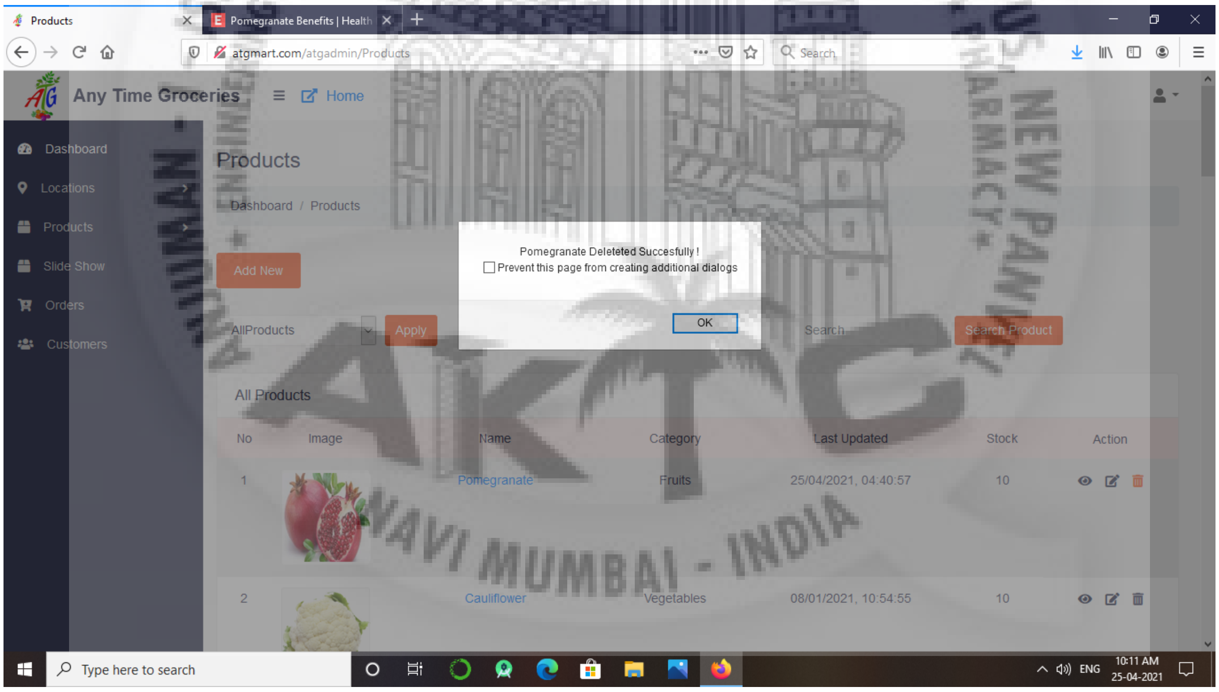
Dashboard / Products

Are you sure want to delete Pomegranate?

All Products

No	Image	Name	Category	Last Updated	Stock	Action
1		Pomegranate	Fruits	25/04/2021, 04:40:57	10	
2		Cauliflower	Vegetables	08/01/2021, 10:54:55	10	

Figure 8.35: Delete Product



Products

Dashboard / Products

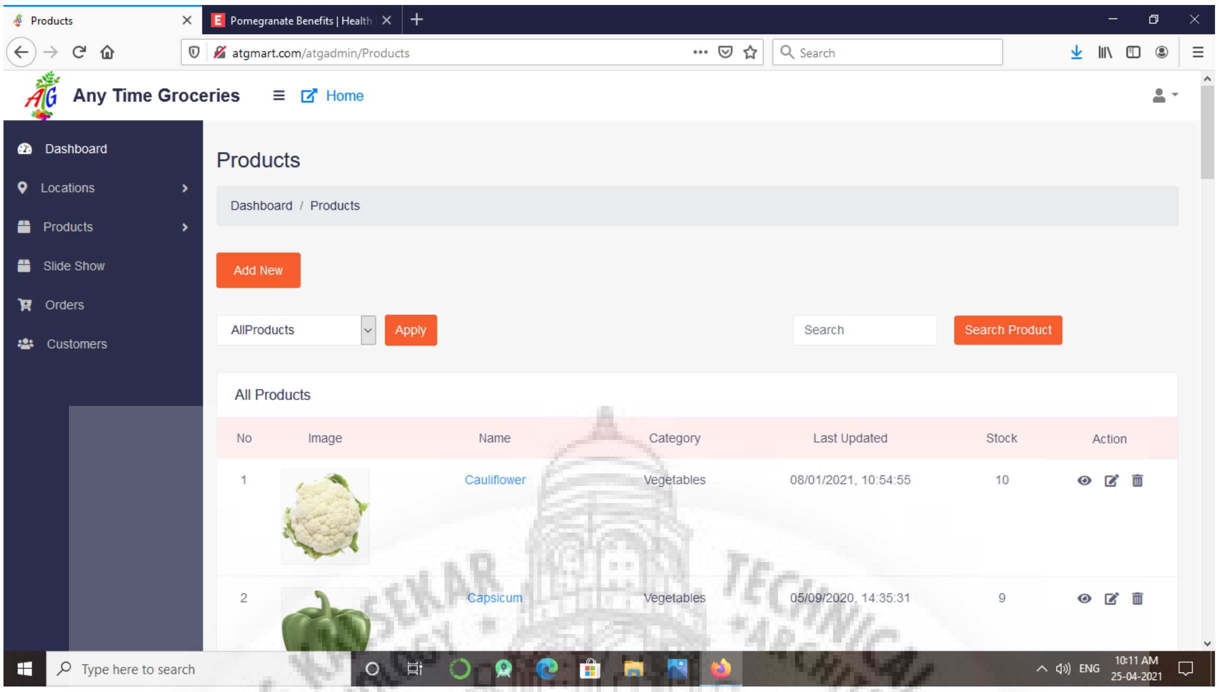
Pomegranate Deleted Succesfully !

☐ Prevent this page from creating additional dialogs

All Products

No	Image	Name	Category	Last Updated	Stock	Action
1		Pomegranate	Fruits	25/04/2021, 04:40:57	10	
2		Cauliflower	Vegetables	08/01/2021, 10:54:55	10	

Figure 8.36: Product deleted successfully


Any Time Groceries · Home

Products

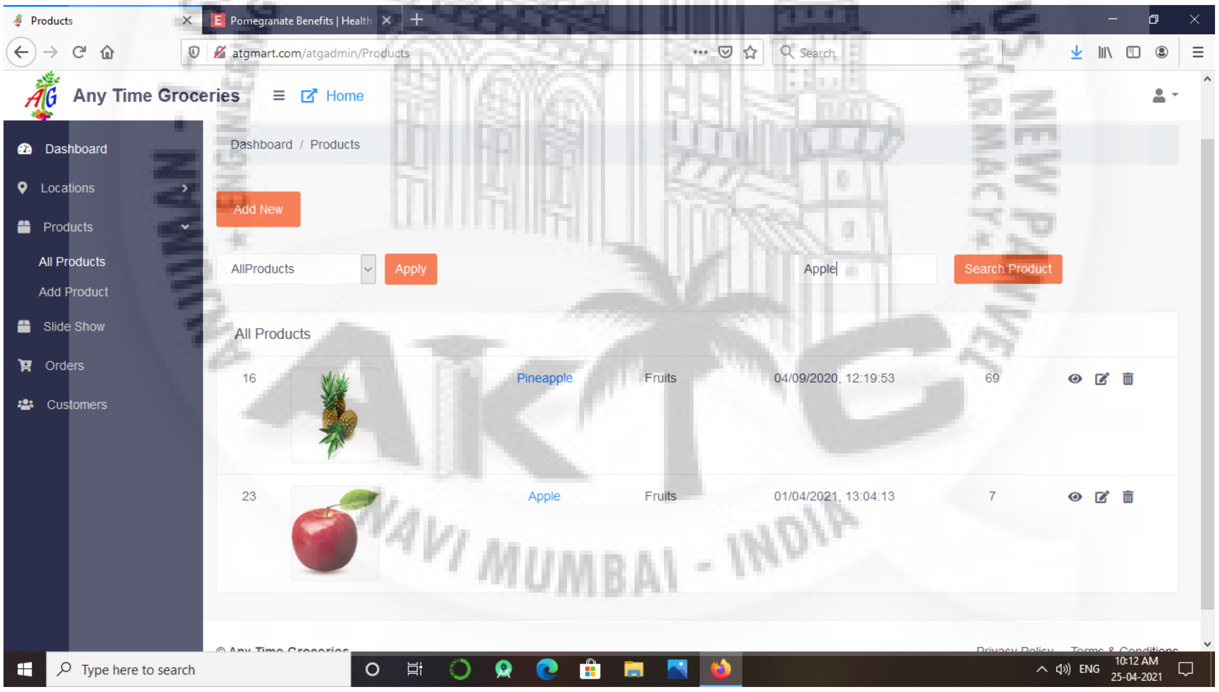
Dashboard / Products

Add New · AllProducts · Apply · Search · Search Product

All Products

No	Image	Name	Category	Last Updated	Stock	Action
1		Cauliflower	Vegetables	08/01/2021, 10:54:55	10	
2		Capsicum	Vegetables	05/09/2020, 14:35:31	9	

Figure 8.37: Product list after deleting the product


Any Time Groceries · Home

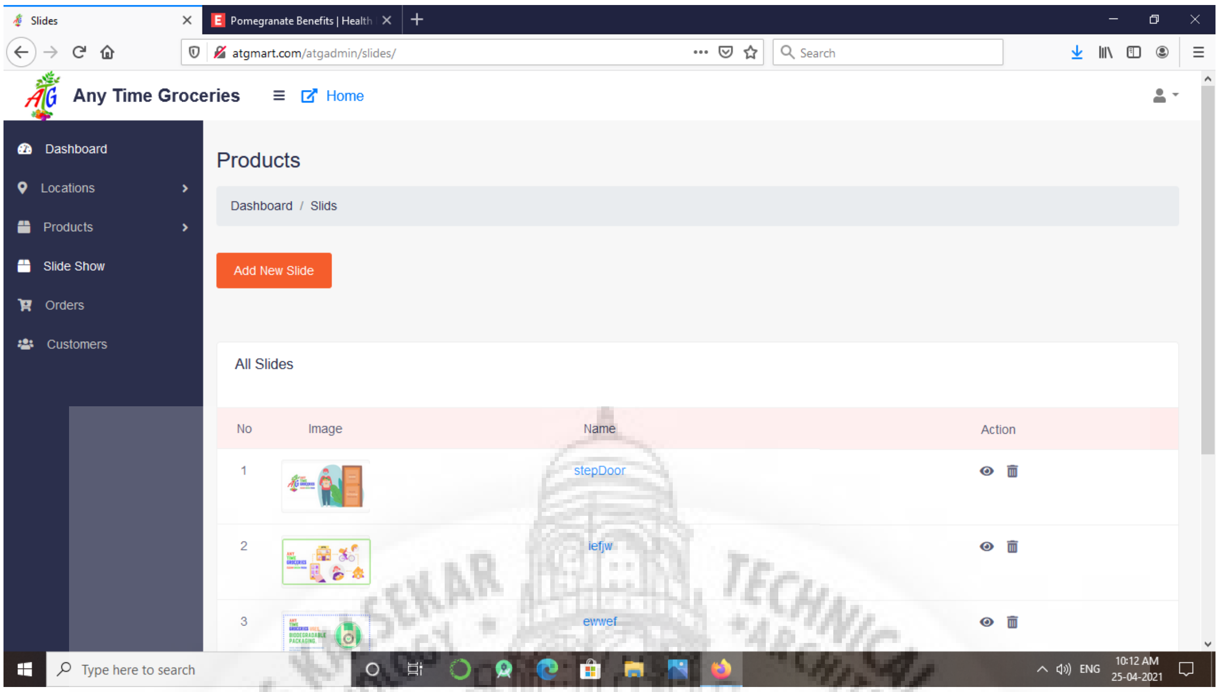
Dashboard / Products

Add New · AllProducts · Apply · Apple · Search Product

All Products

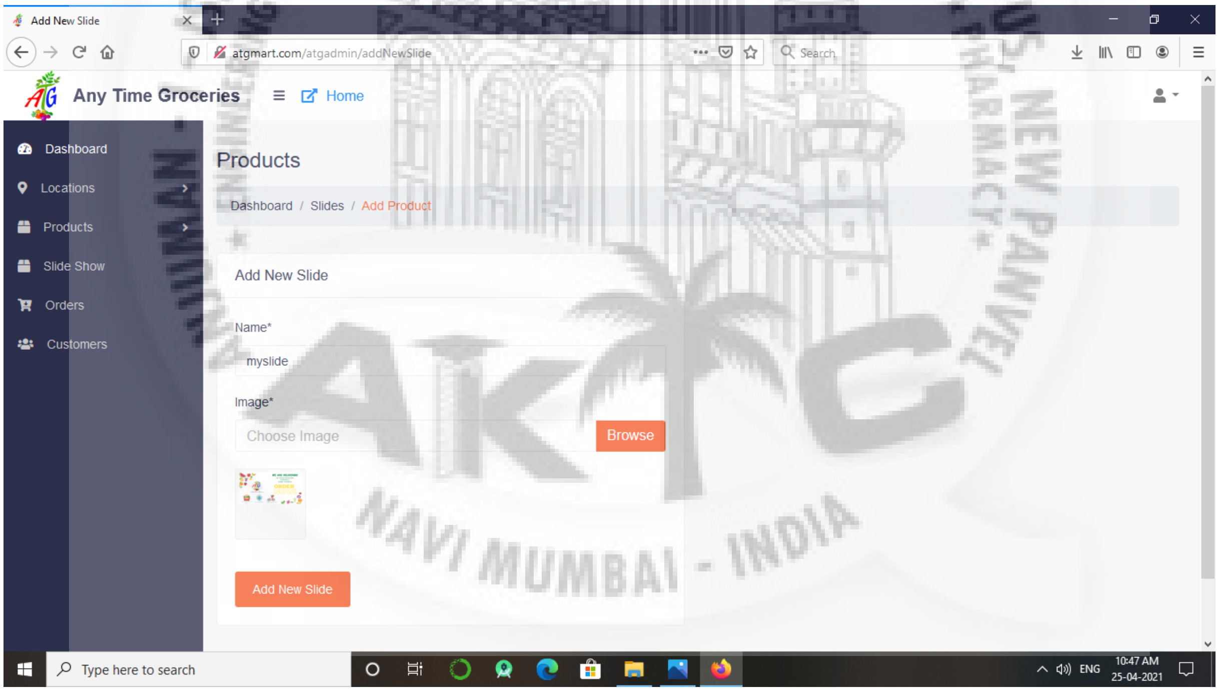
16		Pineapple	Fruits	04/09/2020, 12:19:53	69
23		Apple	Fruits	01/04/2021, 13:04:13	7

Figure 8.38: Search products



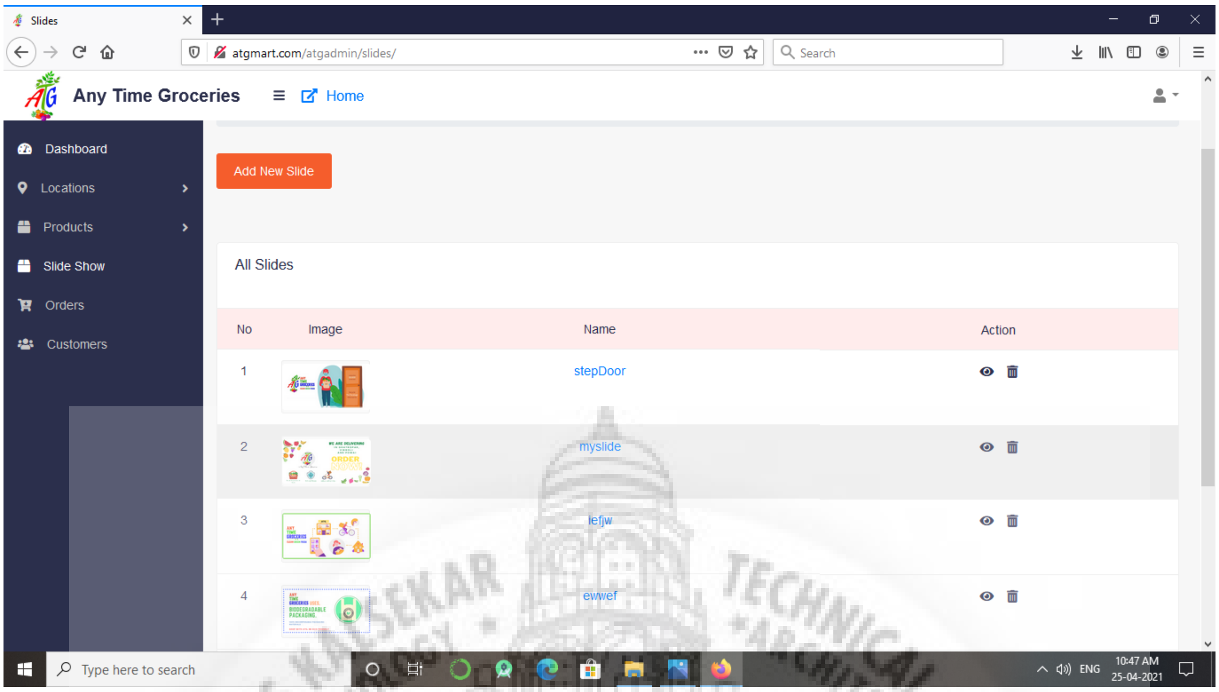
No	Image	Name	Action
1		stepDoor	
2		iefjw	
3		ewwef	

Figure 8.39: Slideshow management



Name*
myslide
Image*
Choose Image

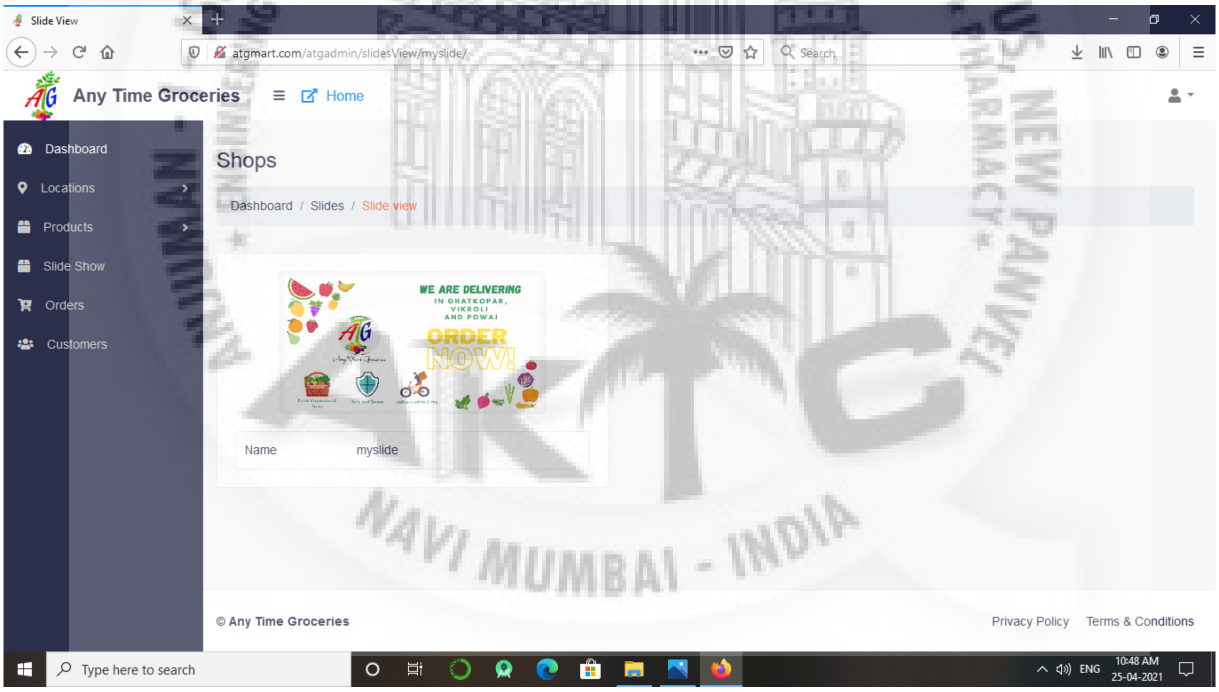
Figure 8.40: Adding a new slide



All Slides

No	Image	Name	Action
1		stepDoor	
2		myslide	
3		iefjw	
4		ewwef	

Figure 8.41: New slide added successfully



Name	myslide

Figure 8.42: Slide details

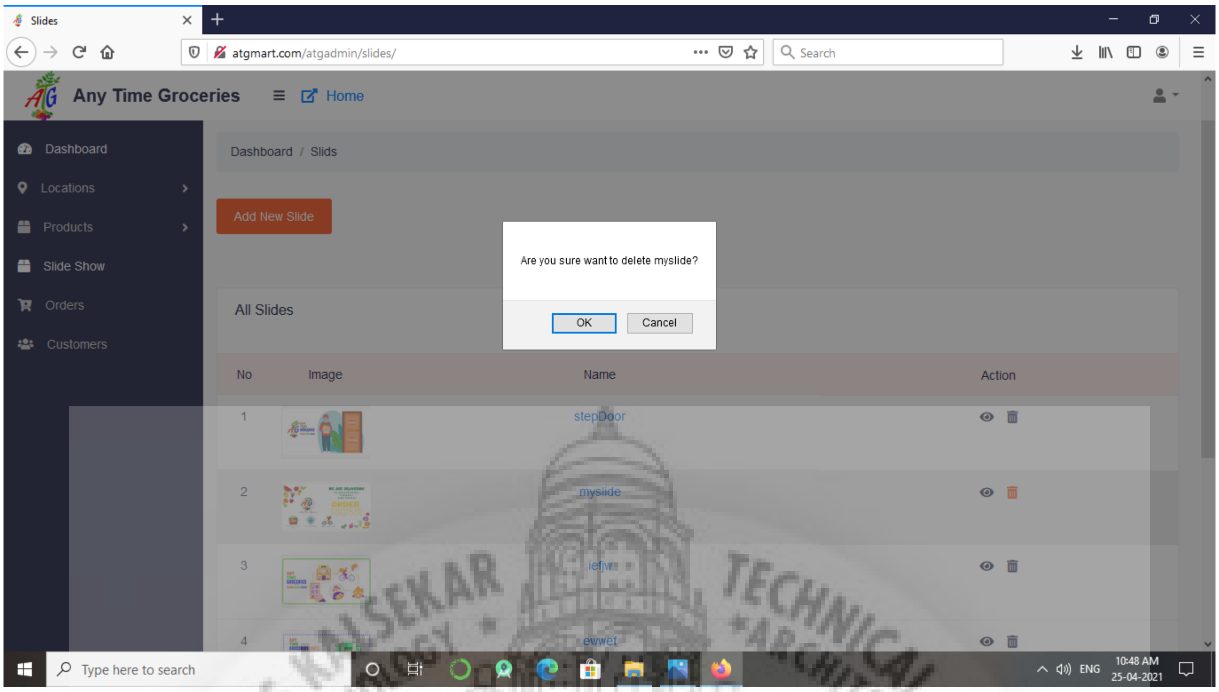


Figure 8.43: Delete a slide

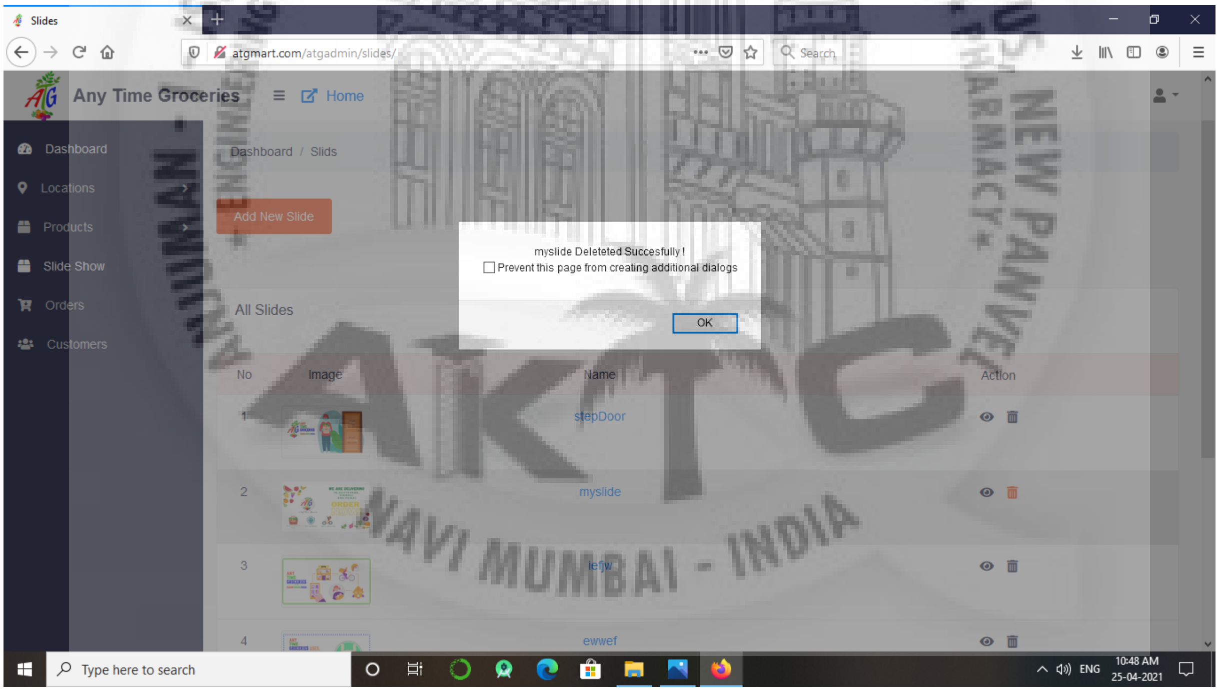
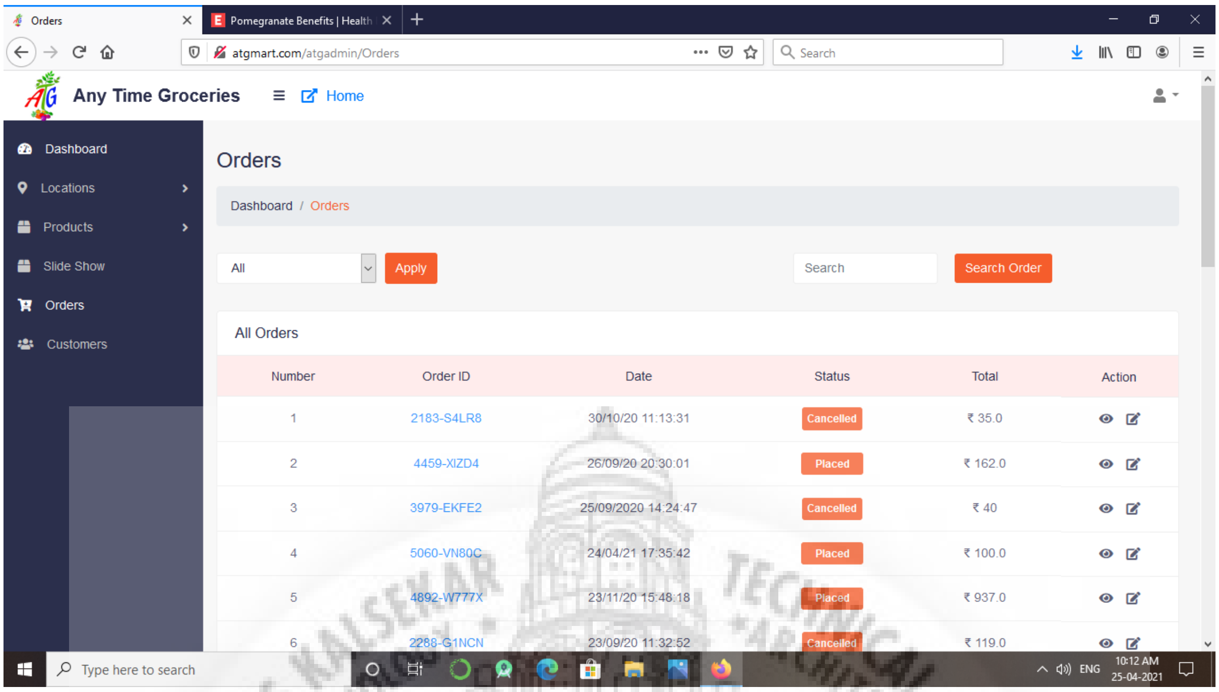


Figure 8.44: Slide deleted successfully



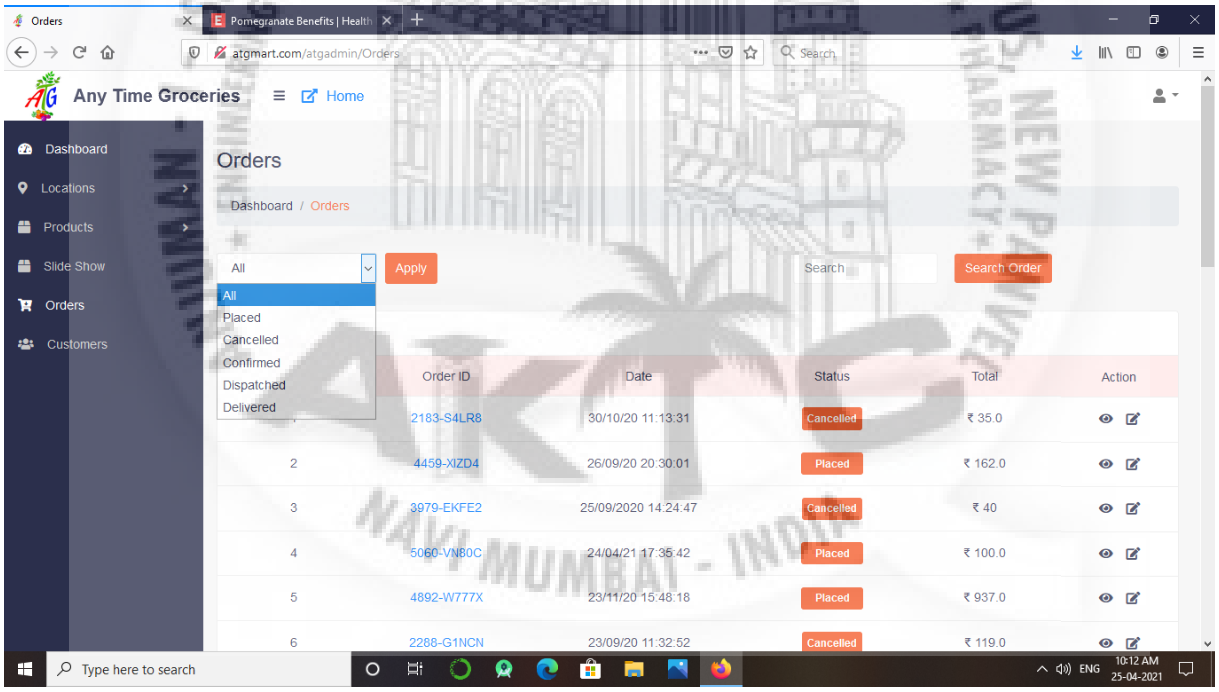
Orders

Dashboard / Orders

All Orders

Number	Order ID	Date	Status	Total	Action
1	2183-S4LR8	30/10/20 11:13:31	Cancelled	₹ 35.0	
2	4459-XIZD4	26/09/20 20:30:01	Placed	₹ 162.0	
3	3979-EKFE2	25/09/2020 14:24:47	Cancelled	₹ 40	
4	5060-VN80C	24/04/21 17:35:42	Placed	₹ 100.0	
5	4892-W777X	23/11/20 15:48:18	Placed	₹ 937.0	
6	2288-G1NCN	23/09/20 11:32:52	Cancelled	₹ 119.0	

Figure 8.45: All orders management

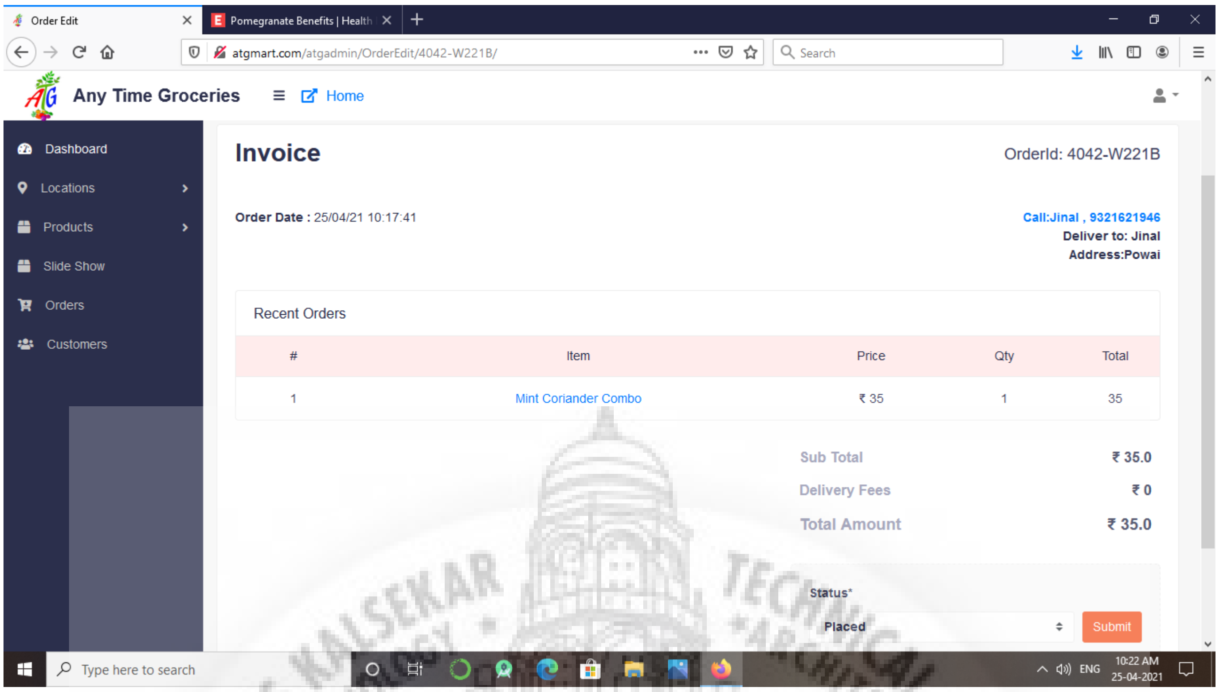


Orders

Dashboard / Orders

Number	Order ID	Date	Status	Total	Action
1	2183-S4LR8	30/10/20 11:13:31	Cancelled	₹ 35.0	
2	4459-XIZD4	26/09/20 20:30:01	Placed	₹ 162.0	
3	3979-EKFE2	25/09/2020 14:24:47	Cancelled	₹ 40	
4	5060-VN80C	24/04/21 17:35:42	Placed	₹ 100.0	
5	4892-W777X	23/11/20 15:48:18	Placed	₹ 937.0	
6	2288-G1NCN	23/09/20 11:32:52	Cancelled	₹ 119.0	

Figure 8.46: Filtering the orders



Invoice

OrderId: 4042-W221B

Order Date : 25/04/21 10:17:41

Call:Jinal , 9321621946
Deliver to: Jinal
Address:Powai

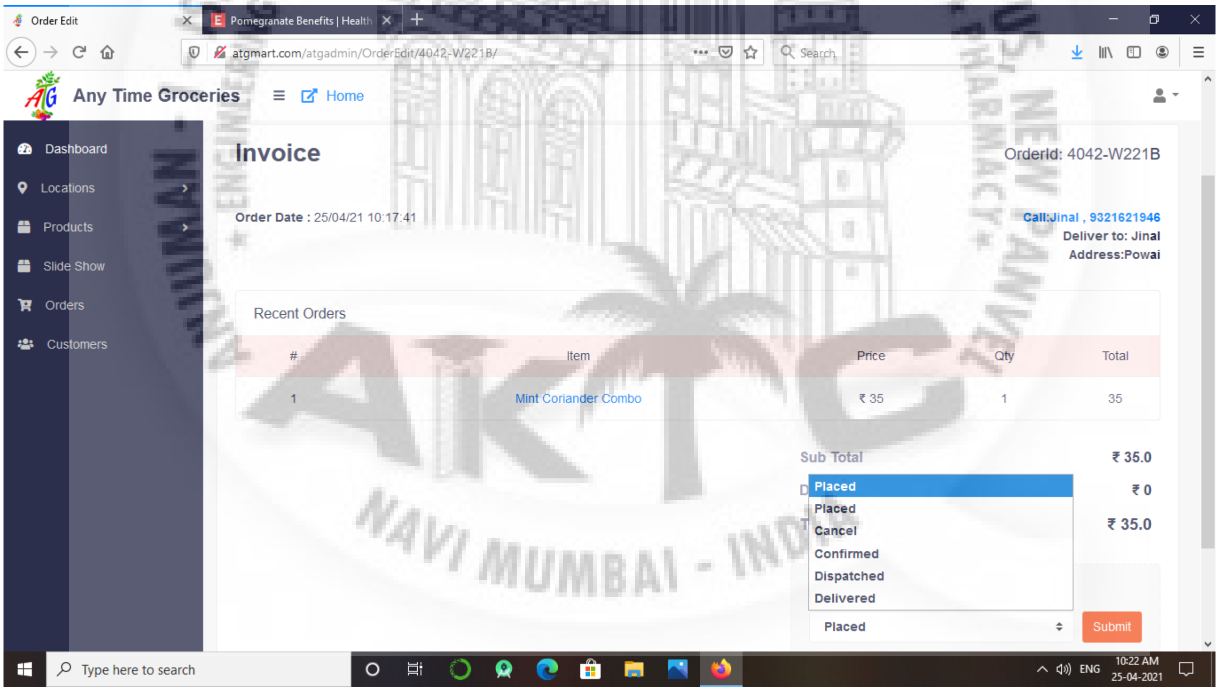
Recent Orders

#	Item	Price	Qty	Total
1	Mint Coriander Combo	₹ 35	1	35

Sub Total	₹ 35.0
Delivery Fees	₹ 0
Total Amount	₹ 35.0

Status*
Placed

Figure 8.47: Order details



Invoice

OrderId: 4042-W221B

Order Date : 25/04/21 10:17:41

Call:Jinal , 9321621946
Deliver to: Jinal
Address:Powai

Recent Orders

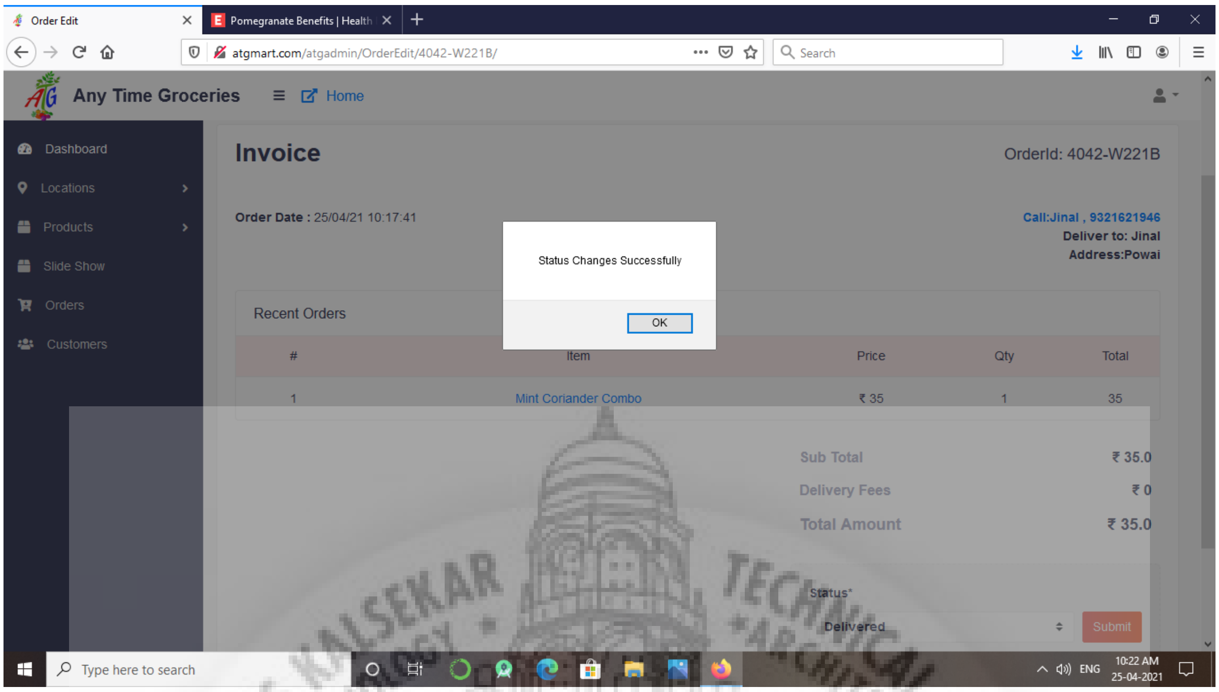
#	Item	Price	Qty	Total
1	Mint Coriander Combo	₹ 35	1	35

Sub Total ₹ 35.0
₹ 0
₹ 35.0

Placed
Placed
Cancel
Confirmed
Dispatched
Delivered

Placed

Figure 8.48: Changing status of order



Invoice

OrderId: 4042-W221B

Order Date : 25/04/21 10:17:41

Call:Jinal , 9321621946
Deliver to: Jinal
Address:Powai

Status Changes Successfully

OK

Recent Orders

#	Item	Price	Qty	Total
1	Mint Coriander Combo	₹ 35	1	35

Sub Total	₹ 35.0
Delivery Fees	₹ 0
Total Amount	₹ 35.0

Status*

Delivered

Submit

Figure 8.49: Status of order changed successfully

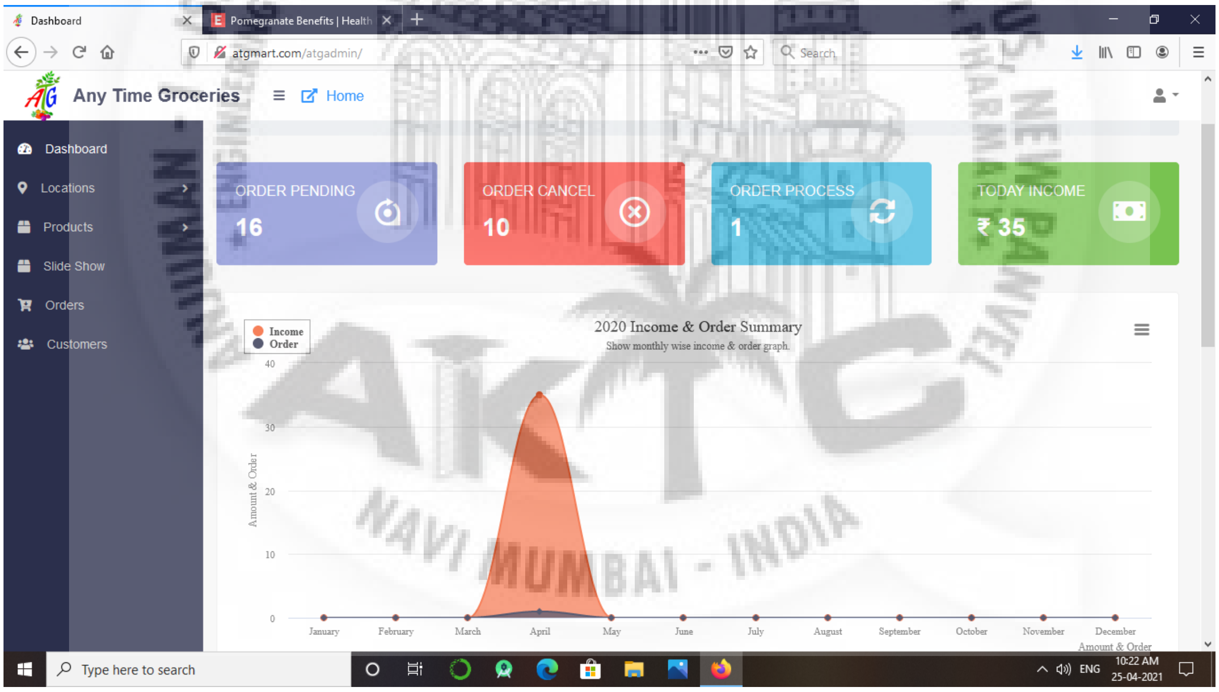


Figure 8.50: Income and order summary

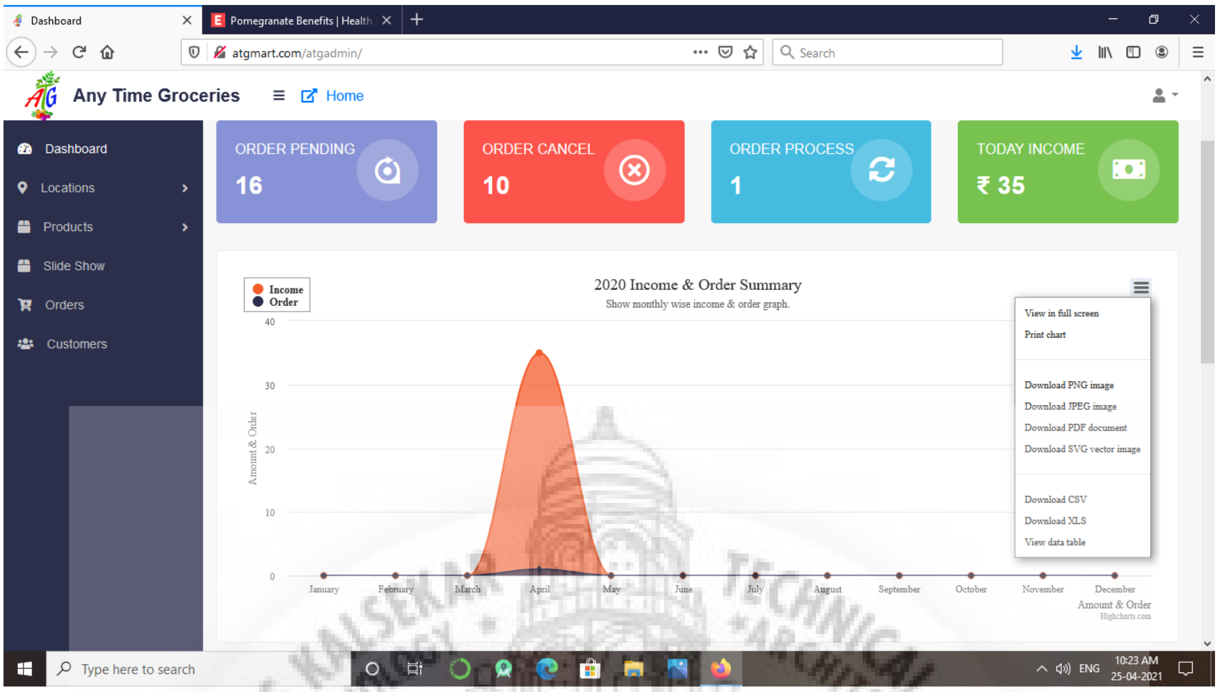
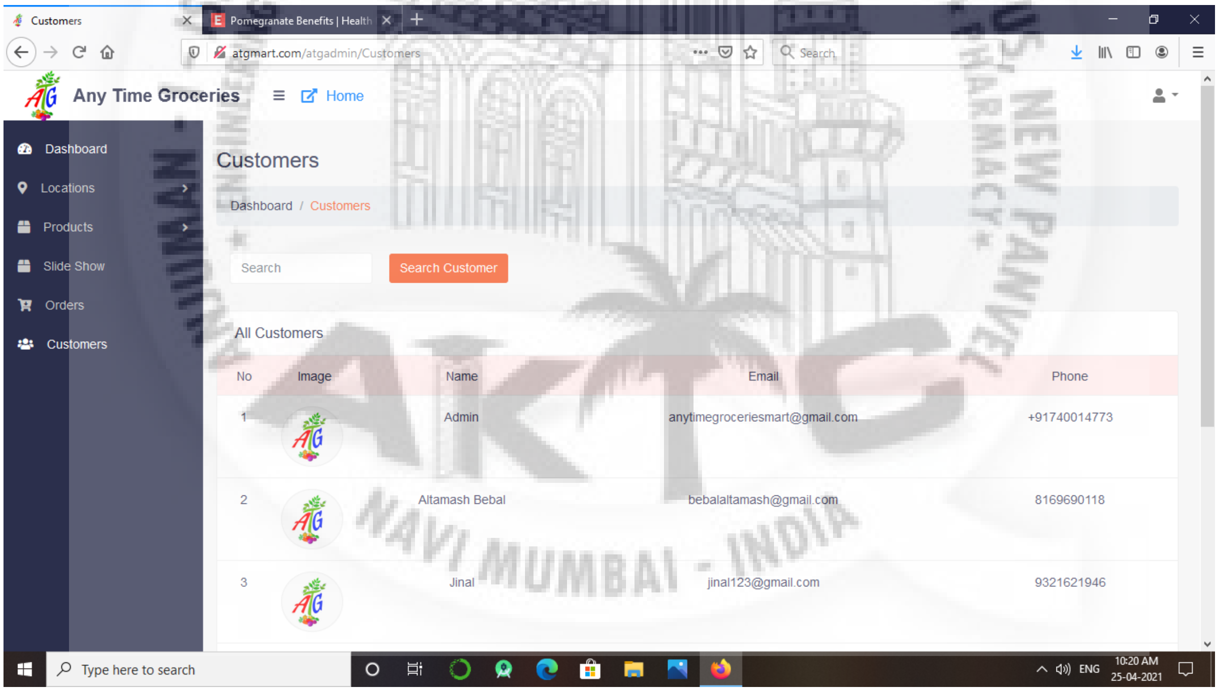
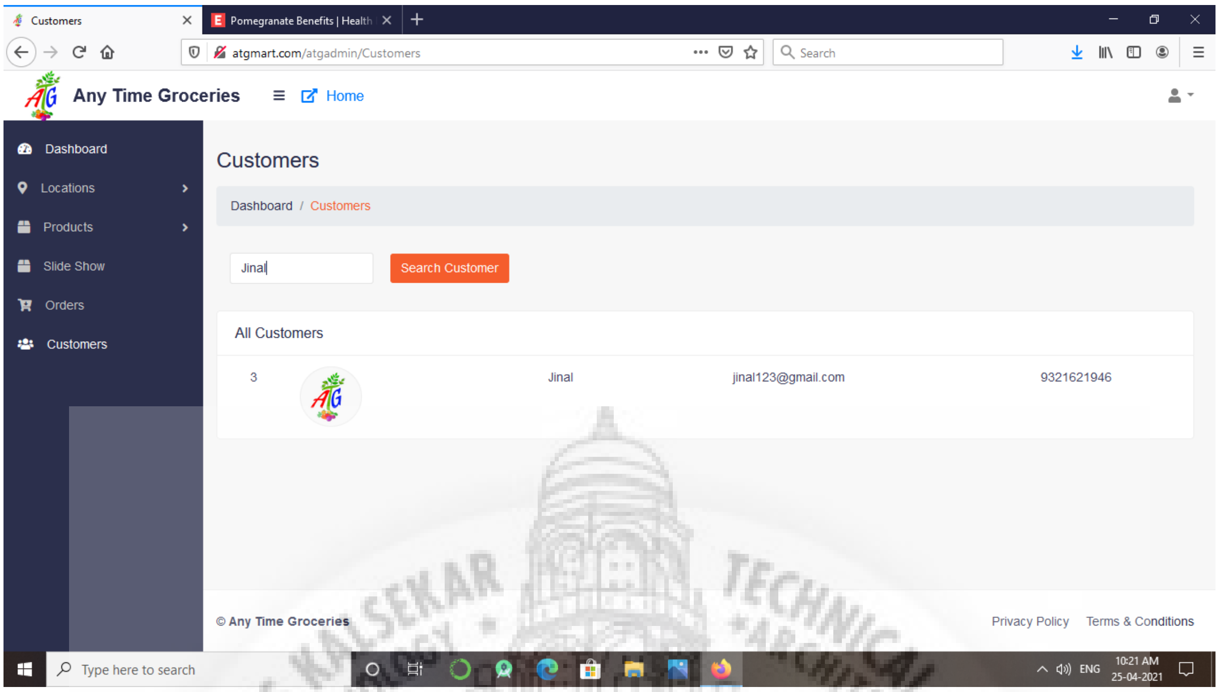


Figure 8.51: Income and order summary download options



No	Image	Name	Email	Phone
1		Admin	anytimegroceriesmart@gmail.com	+91740014773
2		Altamash Bebal	bebalaltamash@gmail.com	8169690118
3		Jinal	jinal123@gmail.com	9321621946

Figure 8.52: All customers management



Customers

Dashboard / Customers

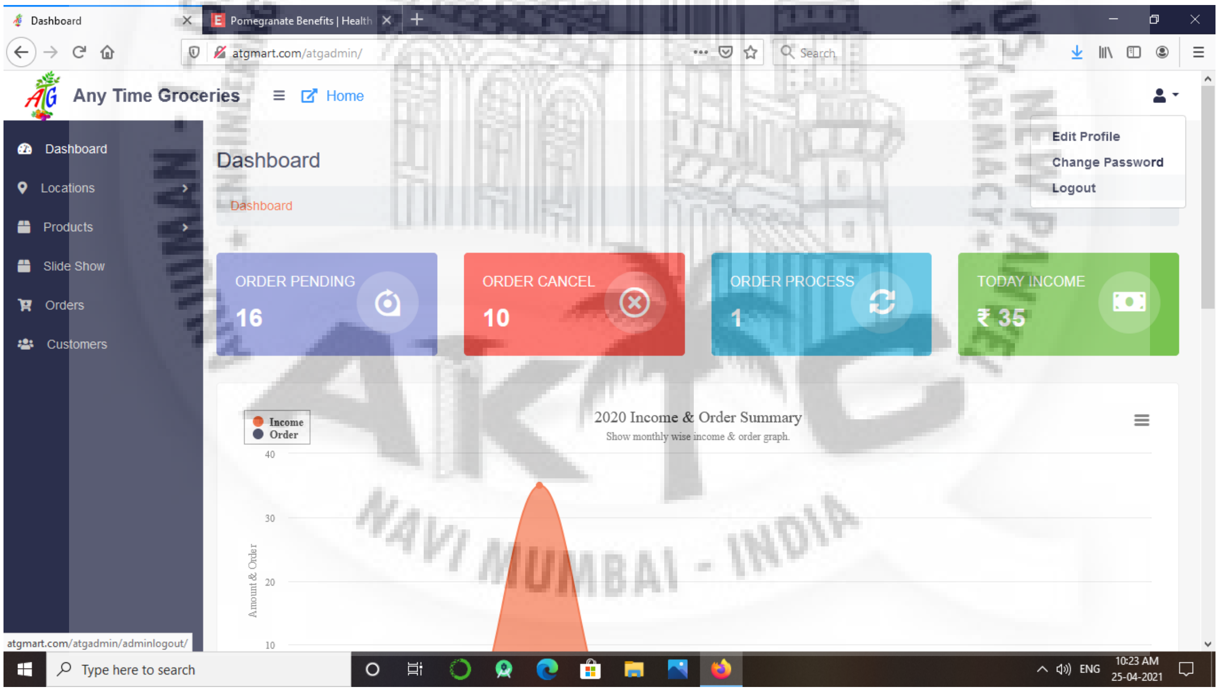
Jinal | Search Customer

All Customers

3		Jinal	jinal123@gmail.com	9321621946

© Any Time Groceries   Privacy Policy   Terms & Conditions

Figure 8.53: Searching a customer



Dashboard

Dashboard

ORDER PENDING 16 | ORDER CANCEL 10 | ORDER PROCESS 1 | TODAY INCOME ₹ 35

2020 Income & Order Summary

Show monthly wise income & order graph.

Figure 8.54: Admin Logout

Chapter 9

Conclusion and Future Scope

9.1 Conclusion

The project introduces an android application through which user is able to shop for daily essentials in pandemic where people required to maintain social distancing and sanitization measures. The application has made user to satisfy the need for daily essentials like vegetables, fruits, herbs etc. With the help of the application user is able to shop without waiting for time slots to be available to him/her which was the most used way of other shopping applications. User got the ordered product on time without any delay along with proper care and safety. Also admin panel is made in such way that admin is able to do all the product management and order management with ease. Which helps in managing the system very efficiently.

9.2 Future Scope

- Recommendation model: We plan to develop recommendation model for user.
- User can get the recommendation based on its past purchases, frequently visited products etc.
- Use of Ml algorithm.
- Wallet implementation.

References

[1] Jasper Grashuis, Theodoros Skevas and Michelle S. Segovia, *Grocery Shopping Preferences during the COVID-19 Pandemic*, Faculty of Management and Commerce,Division of Applied Social Sciences, University of Missouri, Columbia, MO 65211, USA, Received: 17 June 2020; Accepted: 30 June 2020; Published: 2 July 2020.

[2] Mrs.Panuganti Jayasree, Assistant professor, *Consumer behavior-Online grocery shopping in India: An overview*, International Journal of Advance Research in Science and Engineering, August 2019.

[3] Chandini A. V. and Nagendra, Assistant professor, *An Exploratory Study on Consumer Attitude Towards On-line Grocery shopping*, Faculty of Management and Commerce, Ramaiah University of Applied Sciences, Bangalore 560 054.

[4] Django Guide,Django docs: https://docs.djangoproject.com/en/3.2/

[5] Android Developers Guide: https://developer.android.com/guide

[6] COVID-19 Pandemic: https://en.wikipedia.org/wiki/COVID-19-pandemic/

Achievements