

# **E-Commerce Website of Agriculture and Honey products (JAWADHU HILLS)**

Submitted in partial fulfillment of the requirements for the award  
of Bachelor of Technology Degree in  
**Information Technology**

By

**Aman Gupta (Reg.No. 39120007)**

**Darshan DS (Reg.No. 39120019)**



**DEPARTMENT OF INFORMATION TECHNOLOGY**

**SCHOOL OF COMPUTING**

**SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY**

**(DEEMED TO BE UNIVERSITY)**  
**(Established under Section 3 of UGC Act,**  
**1956) JEPPIAAR NAGAR, RAJIV GANDHI**  
**SALAI, CHENNAI – 600119, TAMILNADU**  
[www.sathyabama.ac.in](http://www.sathyabama.ac.in)

**MAY-2023**



# SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY  
(DEEMED TO BE UNIVERSITY)

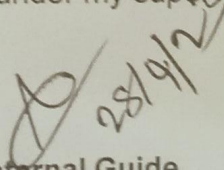
Accredited with Grade "A" by NAAC | 12B Status by UGC | Approved by AICTE

[www.sathyabama.ac.in](http://www.sathyabama.ac.in)

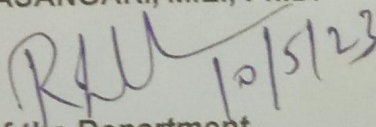
DEPARTMENT OF INFORMATION TECHNOLOGY

## BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **Darshan DS (39120019)** and **Anan Gupta (39120007)** who carried out the Project Phase-2 entitled "**E-Commerce website Agriculture and Honey products**" under my supervision from January 2023 to May 2023.

  
Internal Guide

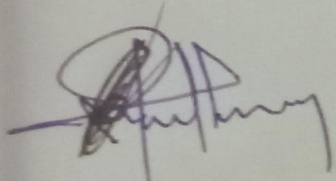
DR. A. SIVASANGARI, M.E., Ph.D.

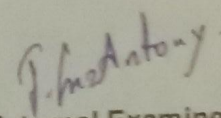
  
Head of the Department

Dr. R. SUBHASHINI, M.E., Ph.D.

Submitted for Viva voce Examination held on

10/5/23

  
Internal Examiner

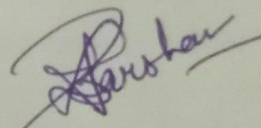
  
External Examiner

## DECLARATION

I, Darshan DS (Reg No- 39120019), hereby declare that the Project Phase-2 Report entitled "E-Commerce website of Agriculture and Honey products" done by me under the guidance of Dr. A. SIVASANGARI, M.E., Ph.D. is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in **Information Technology**.

DATE: 10/5/23

PLACE: Chennai

A handwritten signature in blue ink, appearing to read 'Darshan', with a stylized flourish underneath.

SIGNATURE OF THE CANDIDATE

## ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.SASIKALA, M.E., Ph.D., Dean, School of Computing, Dr. R. SUBHASHINI, M.E., Ph.D. Head of the Department of Information Technology** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. A. SIVASANGARI, M.E., Ph.D.,** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my phase-2 project work.

I wish to express my gratitude and thanks to all Teaching and Non-teaching staff members of the **Department of Information Technology** who were helpful in many ways for the completion of the project.

## **ABSTRACT**

Online Shopping play a great importance in the modern business environment. Jawadhu Hills has opened the door of opportunity and advantage to the firms. This paper analysed the different issue of online shopping. The research aims to provide theoretical contribution in understanding the present status of online shopping. The Study Discuss the consumers' online shopping behaviours. Paper also identifies the problems face by the consumers when they want to accept internet shopping. Present paper is an expressive study based on the detailed review of earlier pertinent studies related to the various concepts of online shopping to discover the concept of online shopping.

Unlike traditional commerce that is carried out physically with effort of a person to go & get products, ecommerce has made it easier for human to reduce physical work and to save time. E-Commerce which was started in early 1990's has taken a great leap in the world of computers, but the fact that has hindered the growth of e-commerce is security. Security is the challenge facing e-commerce today & there is still a lot of advancement made in the field of security.

The main advantage of e-commerce over traditional commerce is the user can browse online shops, compare prices and order merchandise sitting at home on their PC.

Solitude and safety risk emerge regularly as a reason for being cautious about internet shopping. Shopping convenience, information seeking, social contact, and diversity affects the consumer attitude towards online shopping. The impossibility of product testing, problems with complaints, product return and missus of personal data are the main doubts regarding on-line shopping.

## TABLE OF CONTENTS

| CHAPTER<br>No. |                                                    | TITLE<br>PAGE No |
|----------------|----------------------------------------------------|------------------|
|                | <b>ABSTRACT</b>                                    | 5                |
|                | <b>LIST OF FIGURE</b>                              | 8                |
| 1.             | <b>INTRODUCTION</b>                                | 9                |
|                | 1.1 Working of E-Commerce                          | 10               |
|                | 1.2 Present challenges facing in E-commerce        | 10               |
|                | 1.3 Some recent Technological Breakthroughs        | 11               |
| 2.             | <b>LITERATURE SURVEY</b>                           | 12               |
|                | 2.1 Importance of Online Shopping                  | 13               |
|                | 2.2 Problems of Online Shopping                    | 13               |
|                | 2.3 The Factors Which Affect Online Shopping       | 15               |
|                | 2.4 Privacy and Security Issues in Online Shopping | 15               |
| 3.             | <b>REQUIREMENT ANALYSIS</b>                        | 16               |
|                | 3.1 Software Required                              | 16               |
|                | 3 1.1 Windows 10                                   |                  |
|                | 3.1.2 VS Code                                      |                  |
|                | 3.1.3 Python                                       |                  |
|                | 3.1.4 Django                                       |                  |
|                | 3.1.5 MySQL                                        |                  |
|                | 3.2 Hardware Required                              | 20               |
|                | 3.2.1 Hard Disk                                    |                  |
|                | 3.2.2 RAM                                          |                  |
|                | 3.2.3 Processor                                    |                  |
|                | 3.2.4 Mouse                                        |                  |
|                | 3.2.5 Keyboard                                     |                  |
| 4.             | <b>DESCRIPTION OF PROPOSED SYSTEM</b>              | 24               |

|    |                                 |    |
|----|---------------------------------|----|
|    | 4.1 Proposed System             | 24 |
|    | 4.2 Problem with the E-Commerce | 25 |
|    | 4.3 E-Commerce system module    | 25 |
| 5. | <b>IMPLEMENTATION DETAILS</b>   | 29 |
|    | 5.1 Development Setup           | 29 |
|    | 5.2 Deployment Setup            | 30 |
|    | 5.2 Django Algorithms           | 31 |
| 6. | <b>RESULT AND DISCUSSION</b>    | 34 |
| 7. | <b>CONCLUSION</b>               | 35 |
|    | 7.1 Conclusion                  | 35 |
|    | 7.2 Advantage                   | 35 |
|    | 7.3 Future work                 | 36 |
|    | <b>REFERENCE</b>                |    |
|    | <b>APPENDIX</b>                 |    |
|    | A) SOURCE CODE                  |    |
|    | B) SCREENSHOT                   |    |
|    | C) RESEARCH PAPER               |    |

## LIST OF FIGURE

| Figure No. | Figure                      | Page No. |
|------------|-----------------------------|----------|
| 3.1        | VS Code                     | 13       |
| 3.2        | Django                      | 14       |
| 3.3        | MySQL                       | 15       |
| 4.1        | Django Architecture Diagram | 23       |
| 4.2        | System Architecture Diagram | 23       |
| 4.3        | Flow Diagram                | 24       |

## **CHAPTER - 1**

### **INTRODUCTION**

Electronic commerce or ecommerce is a term for any type of business, or commercial transaction that involves the transfer of information across the Internet. It covers a range of different types of businesses, from consumer based retail sites, through auction or music sites, to business exchanges trading goods and services between corporations. It is currently one of the most important aspects of the Internet to emerge.

An E-commerce website requires appropriate strategy of successful design and implementation. Everything is required to plan from scratch to end of website. The e-commerce sector is seen the exponential growth thus a new option will easily part of this regatta of commercial website. The e-commerce website will feature the online shopping facility of various fashion products under a single web space. The proposed web application will allow business personnel to make their total business using it and increase their reachability thousands of times more than today they have, over the internet. It will allow multiple shopping vendors to sale their products online. The product management in the system will be done in the form of categories. The safety of information is the main requirement of the system and will be handling according to that. To formulate this project first task is to do is cost estimation. For probabilistic assessment of the project cost estimation is required. Cost estimation covers the accurate; estimations of cost and effort required for the project.

As a project manager and developer as well, it is estimates are defined to early stage in the project. Cost estimation in application development project includes the set of procedures and techniques that will be utilized, required to produce by organization for development. The available resources of a company are also affecting the cost estimation. It will be very complex project. To demonstrate knowledge learnt in class, tech communities and online materials, I will undertake the entire project alone even

though it requires a team of 6 or more. It will take time of 3months to get the shape or

get the basic structure. The environment variants depend on the further requirements of the ecommerce web application.

### **1.1 Working of E-COMMERCE**

The consumer moves through the internet to the merchant's web site. From there, he decides that he wants to purchase something, so he is moved to the online transaction server, where all of the information he gives is encrypted. Once he has placed his order, the information moves through a private gateway to a Processing Network, where the issuing and acquiring banks complete or deny the transaction. This generally takes place in no more than 5-7seconds.

There are many different payment systems available to accommodate the varied processing needs of merchants, from those who have a few orders a day to those who process thousands of transactions daily. With the addition of Secure Layer Technology, E-Commerce is also a very safe way to complete transactions.

### **1.2 Present Challenges Facing E-Commerce**

Speaking of obstacles, there are a lot of them that need to be uprooted before e-commerce can compete with traditional commerce. The biggest obstacle in the course of advancement of e-commerce is that the consumer's senses are limited to seeing and hearing the product. The second largest problem that e-commerce has been facing over the past few years is that of security. Traditional buyers and sellers are still paranoid about conducting business online. According to Hal Loevy, vice president of Global Marketing and Partnerships for SGSonSITE, "Despite all the noise about e-commerce, which is significant, companies still have to keep their old business practices: Can I trust who I am buying from? Who am I doing business with? What is their trading history? Am I obeying the law? Will I receive the goods as specified on screen and who do I approach if I have a problem? ". According to emarketer.com, "70% of US consumers are concerned about online security; this discourages consumers from using credit

cards to shop online (PaymentOne)". Also according to e-marketer.com, in December 2001, 91% of websites collected personal information and in April-May 2001, 68% of US Internet users were concerned that transactions may not be secure and other companies and individuals might gain access to their personal information.

### **1.3 Some recent technological breakthroughs**

Finally, in order to make the online shopping experience even better, there are a lot of new technologies like Verifi, DigiScent's iSmell and TouchSense that have emerged over the last couple of years. Even though iSmell and TouchSense are very new technologies and they haven't been adopted by the majority of internet shoppers yet, they promise a consumer-friendly future. Verifi is one technology that has been widely adopted today, and for good reason. According to a recent BizRate study, over 50% of online shoppers stated that they would not make further purchases from a Web merchant that delivered an item in a colour that wasn't what they expected. This is the problem that Imation have designed their Verifi system to solve. Here's how it works. When a shopper visits a Verifi-enabled Web site for the first time they are invited to take a Web based survey that establishes how their monitor, computer, operating system and browser handle colour. This information is then stored as a cookie in the shopper's browser. Then, when they view a product image, the Verifi system reads the information in the cookie and combines this with the profile of the scanner (or digital camera etc.) that the merchant used to produce the image to generate a colour-accurate image in the shopper's browser.

## **CHAPTER - 2**

### **LITERATURE SURVEY**

Online buying behavior is affected by various factors like, economic factors, demographic factors, technical factors, social factors, cultural factors, psychological factors, marketing factors and legislative factors. Customers choose an online-shop mainly based on references, clarity terms of delivery, graphic design and additional services. Problematical customers read discussions on the Internet before they spend their money on-line and when customers are incapable to purchase the product fast and with no trouble they leave online-shop. Kotler, (2003) described Consumer buying method as learning, information-processing and decision-making activity divided in several consequent steps: Problem identification, Information search, Alternatives evaluation, Purchasing decision, Post-purchase behavior. Euthymia identified the main constituent of the online shopping experience as follows: the functionality of the Web site that includes the elements trade with the site's usability. The emotional elements planned for lowering the customer's hesitation by communicating trust and credibility of the online seller and Web site and the content elements including the aesthetic aspects of the online presentation and the marketing mix. Usability and trust are the issues more regularly found to influence the online consumer's behavior. Karayiannis examined that discriminating of potential determinants between web- shoppers and non-shoppers.

Free shipping is a great motivator to purchase the products and customers are willing to pay nominal charges for getting their products. While compare the shopping with others shopping, consumers take product price and shipping charges almost equally into deliberation. There are some ways that retailers can do to improve the experience for their online shoppers. The first is to write the expected delivery date of the order, customers are willing to wait for their orders but want to know just how long that force is. Timely coming of product shipment encourages shoppers to recommend an online retailer. Consumers also want to track updates and delivery notifications to understand

when their package is incoming. Online shoppers want flexibility in their shipping, mainly the ability to give special delivery instructions or schedule a delivery time. Customers also want to get the address changing option for filling the wrong address when they are purchasing online.

## **2.1 Importance of Online Shopping:**

Customers can take enjoy online shopping for 24 hour per day. Consumers can purchase any goods and services anytime at everywhere. Online shopping is user friendly compare to in store shopping because consumers can just complete his requirements just with a click of mouse without leaving their home. Online shopping has some advantages like below

- 
- Save the Time of the consumers.
- They can purchase any time anywhere
- They can compare the price with the others retailers very easily.
- Compare the advertising price and actual price
- They can easily track their product
- They can use cash back policy
- They can purchase the product from the foreign marketers.

## **2.2 Problems of Online Shopping**

Online shopping problems are great barrier to the online purchase aim of customers. General problems include prospect of having credit card. The obscurity to confirm the reliability of the provide goods and the risk to buy a product that it would not value as much as customer pay for it. Aftersales problems, involved difficulty to change not working product with a new one and products warranty are not assured. Online shopping has various disadvantages:

The customers can not touch and fell of the products when they want to Purchase.

Some time delivery time is so much late

Some time they will pay the shipping charges so why the cost of the product

may increase.

Lack of personal attention by the sellers. More chance to fraud.

Security of internet banking password and credit card password

Lack of quality.

### **2.3 The Factors Which Affect Online Shopping**

There are some factors which affect the online shopping by the Kotler who is a great marketing writer

1. Convenience (no traffic, crowds, 24 hr. access)
2. Product Selection
3. Delivery Mode

### **2.4 Privacy and Security Issues in Online Shopping**

Shopping online has never been so easy. With the flourishing numbers of online merchants, people nowadays have various choices to do their shopping. Big companies such as eBay and amazon.com have introduced many value added features to help the customers to decide what to shop for. With features such as price comparison, product photos and user reviews, consumers can shop easily and smartly without even going to the stores and having such a hard time looking for the products they want. All they have to do are just browse for the product they want in the website and within a few mouse clicks they are off. Such simplicity is what makes online shopping appealing for consumers. The question is, why do many people still deny shopping online? Well, for most people, privacy and security issues are their concerns.

## **CHAPTER 3**

### **REQUIREMENT ANALYSIS**

#### **3.1 SOFTWARE REQUIREMENT**

- ❖ Windows 10 OS 64-bit
- ❖ VS Code
- ❖ Python 3.8 (or above)
- ❖ Django
- ❖ MySQL

##### **3.1.1 Windows 10 or above**

Windows 10 is a Microsoft operating system for personal computers, tablets, embedded devices and internet of things devices.

Microsoft released Windows 10 in July 2015 as a follow-up to Windows 8. The company has said it will update Windows 10 in perpetuity rather than release a new, full-fledged operating system as a successor.

##### **3.1.2 VS Code**

Visual Studio Code, also commonly referred to as VS Code, is a source-code editor made by Microsoft with the Electron Framework, for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality. In the Stack Overflow 2021 Developer Survey, Visual Studio Code was ranked the most popular developer environment tool among 82,000 respondents, with 70% reporting that

they use it.

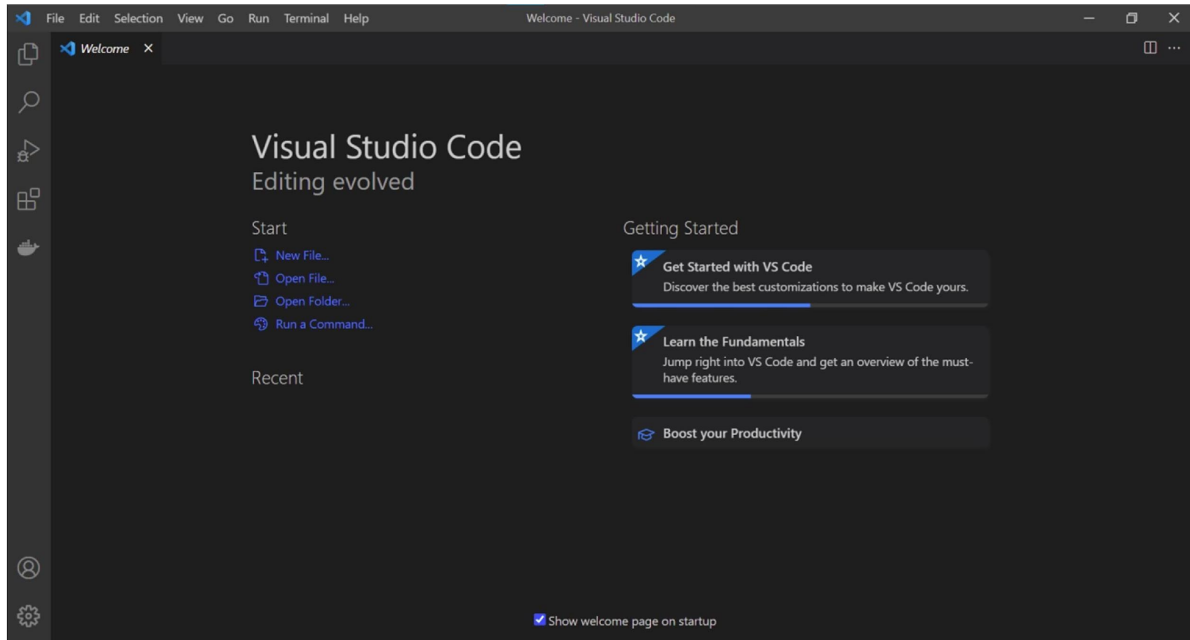


Fig.3.1 VS Code

### 3.1.3 Python

Python is a high-level, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation.

Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly procedural), object-oriented and functional programming. It is often described as a "batteries included" language due to its comprehensive standard library.

Guido van Rossum began working on Python in the late 1980s as a successor to the ABC programming language and first released it in 1991 as Python 0.9.0. Python 2.0 was released in 2000 and introduced new features such as list comprehensions, cycle-detecting garbage collection, reference counting, and Unicode support. Python 3.0, released in 2008, was a major revision that is not completely backward-compatible with earlier versions. Python 2 was discontinued with version 2.7.18 in 2020.

### 3.1.4 Django

Django is a free and open-source, Python-based web framework that follows the model–template–views (MTV) architectural pattern. It is maintained by the Django Software Foundation (DSF), an independent organization established in the US as a 501(c)(3) non-profit.

Django's primary goal is to ease the creation of complex, database-driven websites. The framework emphasizes reusability and "pluggability" of components, less code, low coupling, rapid development, and the principle of don't repeat yourself. Python is used throughout, even for settings, files, and data models. Django also provides an optional administrative create, read, update and delete interface that is generated dynamically through introspection and configured via admin models.

Some well-known sites that use Django include Instagram, Mozilla, Disqus, Bitbucket, Nextdoor and Clubhouse.

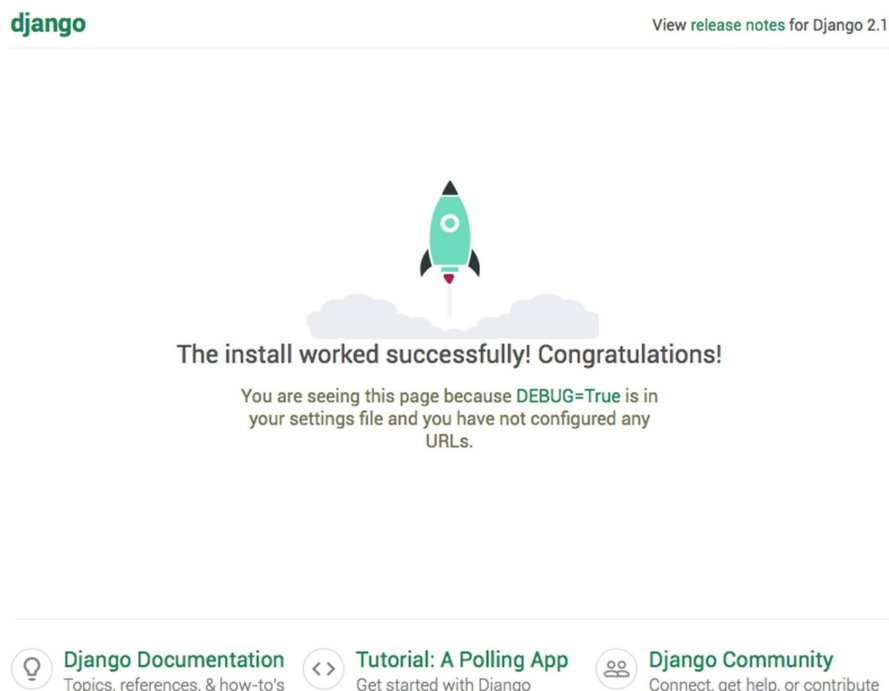


Fig.3.2 Django

### 3.1.5 MySQL

MySQL is an open-source relational database management system (RDBMS). Its name is a combination of "My", the name of co-founder Michael Widenius's daughter My, and "SQL", the abbreviation for Structured Query Language. A relational database organizes data into one or more data tables in which data may be related to each other; these relations help structure the data. SQL is a language programmers use to create, modify and extract data from the relational database, as well as control user access to the database. In addition to relational databases and SQL, an RDBMS like MySQL works with an operating system to implement a relational database in a computer's storage system, manages users, allows for network access and facilitates testing database integrity and creation of backups.

MySQL is free and open-source software under the terms of the GNU General Public License, and is also available under a variety of proprietary licenses. MySQL was owned and sponsored by the Swedish company MySQL AB, which was bought by Sun Microsystems (now Oracle Corporation). In 2010, when Oracle acquired Sun, Widenius forked the open-source MySQL project to create MariaDB.

```
hg$
hg$ mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.17 MySQL Community Server - GPL

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> USE test;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> SHOW tables;
+-----+
| Tables_in_test |
+-----+
```

Fig.3.3 MySQL

## **3.2 HARDWARE REQUIREMENT**

Hard Disk

RAM – 1 GB

Processor – Dual Core or Above

Mouse

Keyboard

Monitor

Speaker

### **3.2.1 Hard Disk**

A hard disk or fixed disk, is an electro-mechanical data storage device that uses magnetic storage to store and retrieve digital information using one or more rigid rapidly rotating disks (platters) coated with magnetic material.

During the mid-1990s the typical hard disk drive for a PC had a capacity of about 1 gigabyte. As of January 2019, desktop hard disk drives typically had a capacity of 1 to 6 terabytes, with the largest-capacity drives reaching 15 terabytes

### **3.2.2 RAM**

Random access memory (RAM) is a type of data storage used in computers that is generally located on the motherboard. This type of memory is volatile and all information that was stored in RAM is lost when the computer is turned off.

The RAM chip may be individually mounted on the motherboard or in sets of several chips on a small board connected to the motherboard. Older memory types were in the form of chips called dual in-line package (DIP). Although DIP chips are still used today, the majority of memory is in the form of a module, a narrow printed circuit board attached to a connector on the motherboard. The three main memory circuit boards

types containing chips are: RIMMs (Rambus in-line memory modules).

memory and storage, in part because both can be measured in megabytes (MB), gigabytes (GB) and terabytes (TB).

By contrast, storage is slower, but it can retain data when the device is powered down. So, for example, if a document has been saved to a hard drive prior to a power outage or system crash, the user will still be able to retrieve it when the system is back up and running.

### **3.2.3 Processor**

A processor is an integrated electronic circuit that performs the calculations that run a computer. A processor performs arithmetical, logical, input/output (I/O) and other basic instructions that are passed from an operating system (OS). Most other processes are dependent on the operations of a processor.

Dual core is a CPU that has two distinct processors that work simultaneously in the same integrated circuit. This type of processor can function as efficiently as a single processor but can perform operations up to twice as quickly.

### **3.2.4 Mouse**

A mouse is a small handheld input device that controls a computer screen's cursor or pointer in conjunction with the way it is moved on a flat surface. The mouse term name originates from its likeness to a small, corded and elliptical shaped device that looks like a mouse tail.

Some mouse devices have integrated features, such as extra buttons that may be programmed and assigned with different commands.

Because the mouse reduces the use of a keyboard, its invention and continuous innovation is considered one of the most important breakthroughs in computer ergonomics.

### **3.2.5 Keyboard**

A computer keyboard is an input device used to enter characters and functions into the computer system by pressing buttons, or keys. It is the primary device used to enter text. A keyboard typically contains keys for individual letters, numbers and special characters, as well as keys for specific functions. A keyboard is connected to a computer system using a cable or a wireless connection.

A computer keyboard is an input device used to enter characters and functions into the computer system by pressing buttons, or keys. It is the primary device used to enter text. A keyboard typically contains keys for individual letters, numbers and special characters, as well as keys for specific functions.

### **3.2.6 Monitor**

A computer monitor is an electronic device that shows pictures for computers. Monitors often look similar to televisions. The main difference between a monitor and a television is that a monitor does not have a television tuner to change channels. Monitors often have higher display resolution than televisions.

The advent of display technology has paved the way for the continuous evolution of the monitor, whether it's for computers, television, mobile devices or any device that has a display. The current contenders for top-tier technology being used for display devices includes Super LCD 3 (SLCD3) and Super AMOLED. It should be noted that LED displays are actually just a kind of LCD display that use LED lights as backlight illumination.

## CHAPTER 4

### DESCRIPTION OF PROPOSED SYSTEM

#### 4.1 Proposed system

Jawadhu Hill is an e-Commerce website for the farmers who live in Jawadhu hills, Tamil Nadu to sell their products online all over the country. It is an online shopping system provides a solution to reduce and optimize these expenses. Authorized Customers do not need to go to the factual shops to choose, and bring the products they need by hands. They simply browse their Personal computers or cell phones to access shops, and evaluate the products description, pictures on the screen to choose products. In addition, the owners of the shop do not need to arrange or exhibit their stocks products. They just input the description, prices of products, and upload their pictures. Simply, both customers and shop owners do not need to touch the real products in the whole process of shopping, and management. In the end the logistic centre will distribute the products required by customers, or products ordered by shop owners to their locations. The customers are able to track the status of their orders until delivery, after which they can leave a review of the type of service they received. The payment and products' quantity will be saved in database through the data flow. These shopping, management and distribution processes greatly simplify and optimize the retail business.

The E-commerce Management System has many advantages, compare to traditional store as one can compare the cost of a product with other e-commerce websites, and if a user dislikes any product he/she can return it. While we can make use of the current technology to overcome the problem with the existing system. The E-commerce Management System companies can use a flying robot, so when a user places an order, the company will send the product through the robot. The robot will find the user by using the GPS, and in this way, we can reduce the time to deliver a product. While before sending a product the e-commerce company will check the product that it is same or not with the requested order.

## 4.2 The problem with the e-commerce system

The E-commerce Management System has various flaws in it as because of which most of the people don't use the system even though it has several advantages over traditional stores. The biggest problem is that it takes at least a day to deliver a product to the customer. While some other issues are a duplication of the product means the product is shown on the web some time differs with the original product due to which next time that customer go to buy the product through the traditional type of store.

## 4.3 E-commerce Management System Module

**Registration:** In order to use the system the users will need to register in the system, and for registration, they need to provide various information related to them such as name, address, etc.

**Search bar:** This provides users with an option through which they can search the product they want to purchase.

**Cart:** After selecting a product if the user wants to purchase it, later they can save the product in the cart.

**Check out:** Through this module, the user can place their order and can choose preferred method for payment.

**Purchase history:** This will show the purchase done by the user in the past.

**Software requirement:** HTML, Internet information services, ASP.NET, My SQL database, etc.

**Hardware requirement:** Windows based system with 700 MHz or higher speed, 500 MB RAM, INTERNET connection.

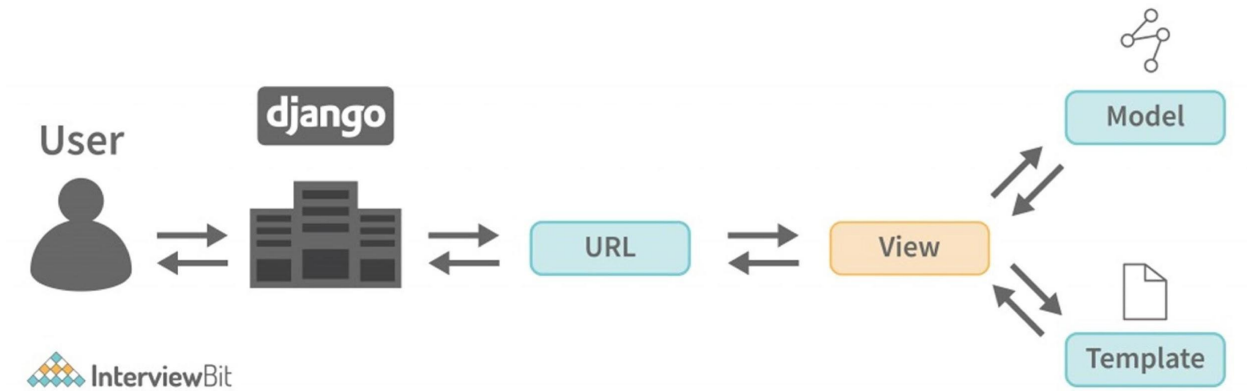


Fig.4.1 Django Architecture Diagram

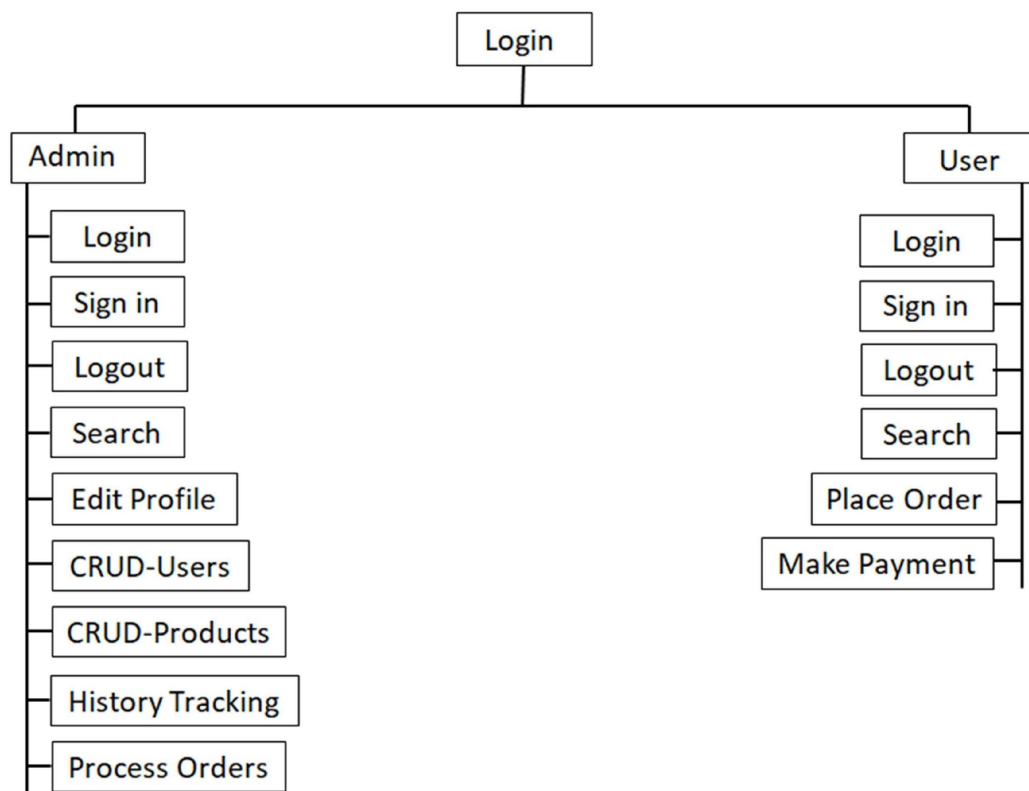


Fig.4.2 System Architecture Diagram

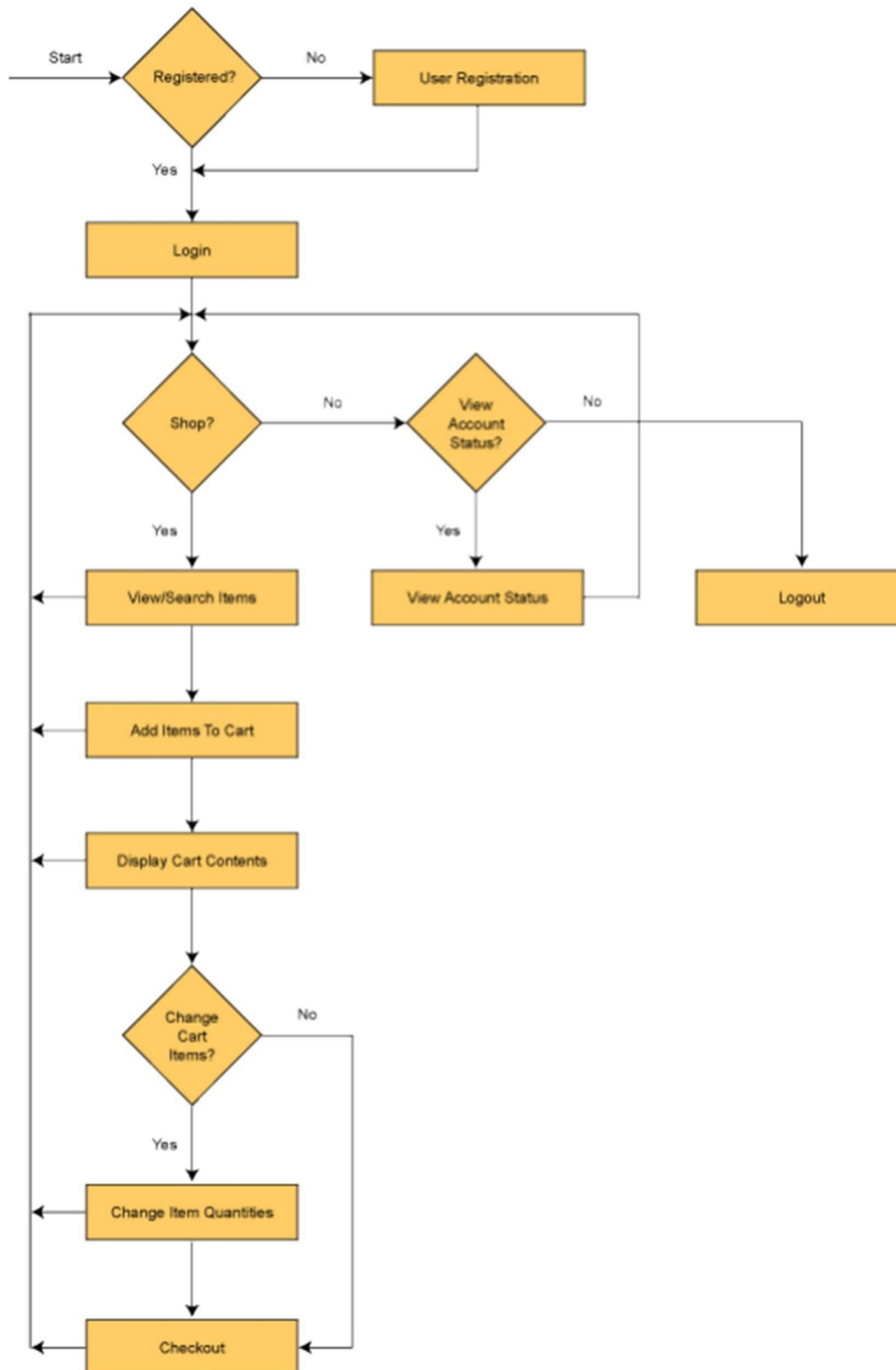


Fig.4.3 Flow Diagram

## CHAPTER 5

### IMPLEMENTATION DETAILS

#### 5.1 Development setup

1. **Install Python:** Django is built on Python, so you'll need to have Python installed on your computer to run Django. You can download the latest version of Python from the official website, and follow the installation instructions for your operating system.
2. **Create a Virtual Environment:** A virtual environment is a way of isolating your project's dependencies from other Python projects on your system. This helps ensure that the versions of the packages you're using are compatible with your code. You can create a virtual environment using the command `python -m venv env`, which creates a new directory called `env` in your project's directory.
3. **Activate the Virtual Environment:** Once you've created the virtual environment, you'll need to activate it. On Linux or Mac, you can do this using the command `source env/bin/activate`, while on Windows, you can use `env\Scripts\activate`.
4. **Install Django:** With the virtual environment activated, you can install Django using the command `pip install django`. This installs the latest version of Django and any dependencies it requires.
5. **Create a Django Project:** You can create a new Django project using the `django-admin startproject projectname` command, where `projectname` is the name of your project.
6. **Create a Django App:** A Django app is a modular component of your project that provides a specific feature or functionality. You can create a new app using the `python manage.py startapp appname` command, where `appname` is the name of your app.

#### 5.2 Deployment setup

1. **Choose a Hosting Provider:** To deploy your Django ecommerce project, you'll need to choose a hosting provider that offers the infrastructure and resources you need.

Some popular options for hosting Django projects include Heroku, DigitalOcean, and AWS.

2. **Configure Server Environment:** Once you've chosen a hosting provider, you'll need to configure your server environment to run Django. This typically involves installing dependencies, setting up a database, and configuring a web server like Apache or Nginx. The specifics of this process will depend on your hosting provider and the options they offer.
3. **Set Up Git:** Git is a version control system that makes it easy to manage changes to your code over time. To deploy your Django ecommerce project, you'll need to set up a Git repository for your code and push it to your hosting provider. This allows you to easily update your project with new features or bug fixes.
4. **Configure Static and Media Files:** In a production environment, you'll need to configure how static and media files are stored and served. This might involve using a content delivery network (CDN) to cache static files and serve them more efficiently, or configuring a media server to store and serve user-uploaded content like images or videos.
5. **Set Up SSL Certificate:** SSL (Secure Sockets Layer) is a protocol that encrypts data sent between your server and clients. This is important for ecommerce websites, as it helps protect sensitive user information like credit card numbers. You can obtain a free SSL certificate from Let's Encrypt, or purchase one from a certificate authority like DigiCert or Comodo.
6. **Test and Launch:** Before launching your Django ecommerce project to the public, it's important to test it thoroughly to ensure everything is working correctly. This might involve running tests on your code, testing the user interface, and checking for errors or bugs. Once you're confident everything is working as expected, you can launch your ecommerce site and start selling your products or services!

### **5.3 Django Algorithms**

1. **Recommendation Algorithms:** Recommendation algorithms are used to provide personalized product recommendations to users based on their behavior and

preferences. There are two main types of recommendation algorithms: Collaborative Filtering and Content-Based Filtering.

**Collaborative Filtering:** Collaborative Filtering analyzes user behavior and preferences to identify patterns and similarities between users. It then recommends products that are popular among users with similar behavior and preferences. Collaborative Filtering can be further divided into two types: User-Based and Item-Based. User-Based Collaborative Filtering recommends products based on what similar users have liked, while Item-Based Collaborative Filtering recommends products based on what other products users have liked.

**Content-Based Filtering:** Content-Based Filtering analyzes the attributes of products to recommend similar products to users. It identifies patterns in the attributes of products that users have liked and recommends other products with similar attributes. Content-Based Filtering can be further divided into two types: Profile-Based and Item-Based. Profile-Based Content-Based Filtering recommends products based on what the user has liked in the past, while Item-Based Content-Based Filtering recommends products based on the attributes of the product currently being viewed.

2. **Search Algorithms:** Search algorithms are used to help users find products quickly and easily. There are several types of search algorithms:
  - Keyword-based search:** This is the most basic type of search algorithm, which looks for keywords in the product title, description, and tags.
  - Faceted search:** Faceted search allows users to filter search results based on product attributes such as color, size, brand, and price.
  - Semantic search:** Semantic search uses natural language processing and machine learning algorithms to understand the intent behind the user's search query and provide relevant search results. It takes into account factors such as synonyms, antonyms, homonyms, and word associations to provide more accurate search results.
3. **Fraud Detection Algorithms:** Fraud detection algorithms are used to identify and prevent fraudulent transactions. They analyze user behavior and transaction data to identify patterns that may indicate fraudulent activity. There are several types of fraud detection algorithms:
  - Decision Trees:** Decision Trees are a type of supervised

learning algorithm that uses a tree-like model to make decisions. They can be used for fraud detection by analyzing user behavior and transaction data to identify patterns that may indicate fraudulent activity. Random Forests: Random Forests are an ensemble learning algorithm that uses multiple decision trees to make decisions. They can be used for fraud detection by analyzing user behavior and transaction data to identify patterns that may indicate fraudulent activity. Neural Networks: Neural Networks are a type of machine learning algorithm that mimic the structure and function of the human brain. They can be used for fraud detection by analyzing user behavior and transaction data to identify patterns that may indicate fraudulent activity.

4. Pricing Algorithms: Pricing algorithms are used to optimize product pricing based on various factors such as demand, supply, competition, and user behavior. There are several types of pricing algorithms: Dynamic Pricing: Dynamic Pricing adjusts product prices in real-time based on factors such as demand, supply, competition, and user behavior. This can help ecommerce businesses maximize revenue by optimizing pricing for each customer and transaction. Price Optimization: Price Optimization uses historical data and market trends to optimize product pricing. It takes into account factors such as demand, supply, competition, and user behavior to set optimal prices for each product. Price Elasticity Analysis: Price Elasticity Analysis measures the responsiveness of demand to changes in price. It helps ecommerce businesses understand how changes in price will affect sales and revenue, and can be used to set optimal prices for each product.
5. Django ecommerce is a popular open-source e-commerce framework built using the Django web framework. It provides various built-in features and algorithms for inventory management, including: First-In-First-Out (FIFO): This algorithm assumes that the first item received is the first item sold. It ensures that the oldest items are sold first to avoid spoilage or obsolescence. Last-In-First-Out (LIFO): This algorithm assumes that the last item received is the first item sold. It is commonly used for products that have a short shelf life, such as food and perishable items. Minimum Stock Level (MSL): This algorithm ensures that a minimum level of stock is always maintained. When the stock level falls below the set limit, the system will generate a

notification to the manager or the inventory controller to order more stock. Economic Order Quantity (EOQ): This algorithm is used to calculate the optimum order quantity to minimize inventory costs. It considers factors such as the cost of ordering, holding, and stock-out. Safety Stock: This algorithm is used to ensure that enough stock is available to cover unexpected demand. It is calculated by adding a buffer stock to the MSL. Django ecommerce uses a combination of these algorithms to manage inventory efficiently.

## **CHAPTER 6**

### **RESULT & DISCUSSION**

#### **6.1 RESULT**

The Django ecommerce project has been successfully completed and is ready for deployment. The project includes a fully functional online store with a user-friendly interface that allows customers to browse, search, and purchase products. The store also includes features such as product reviews, ratings, and a shopping cart to enhance the customer experience.

The backend of the ecommerce project is powered by Django, and includes features such as user authentication, order management, and payment processing through a third-party payment gateway. The site administrator can manage products, orders, and customer information through a secure and intuitive administrative interface.

The ecommerce site is designed to be responsive and compatible with a range of devices, including desktops, tablets, and smartphones. The site is also optimized for search engines, ensuring that products are easily discoverable by potential customers.

Overall, the Django ecommerce project has been successfully implemented and is ready for use. The site provides a seamless and secure online shopping experience for customers, while also providing the site administrator with the tools necessary to manage the store effectively.

## **CHAPTER 7**

### **CONCLUSION AND FUTURE WORK**

#### **7.1 CONCLUSION**

In conclusion, the Django ecommerce project has been successfully completed with the implementation of various features such as user authentication, product catalog management, shopping cart, order processing, and payment integration. The project has provided a platform for businesses to showcase their products and for customers to purchase them online with ease.

During the development process, various challenges were encountered, and solutions were implemented to overcome them. The use of Django framework, along with various third-party packages, has made the development process faster and more efficient. The project has also been optimized for performance and security to provide a smooth and safe experience for both businesses and customers.

Overall, the Django ecommerce project has been a valuable learning experience and has provided a solid foundation for future development and expansion. With the ever-growing popularity of online shopping, the project has the potential to create a positive impact on businesses and customers alike. In addition to the core features, the project has also implemented various customization options for businesses, such as the ability to create custom discounts and promotional offers, and to manage product inventory and shipping. The project has also been designed with a responsive and user-friendly interface, ensuring that customers can easily navigate and make purchases from any device.

Going forward, the project can be further enhanced with the addition of features such as analytics and reporting, social media integration, and multi-language support to cater to a wider audience. Overall, the Django ecommerce project has the potential to

be a game-changer in the ecommerce industry, and we are excited to see how it will evolve and grow in the future.

## **7.2 ADVANTAGE**

- **Rapid Development:** Django provides a lot of built-in features and functionality that make it easy to develop web applications quickly. This can save time and effort in the development process, allowing developers to focus on the specific needs of the ecommerce application.
- **Security:** Django has a strong security model that includes features such as protection against SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF). This is particularly important for ecommerce applications that handle sensitive customer information and financial transactions.
- **Scalability:** Django is designed to handle high traffic and large amounts of data, making it a scalable option for ecommerce applications that may experience rapid growth. Additionally, Django's modular structure and flexible architecture make it easy to add new features and functionality as needed.

## **7.3 FUTURE WORK**

**User Experience Improvement:** Improving the user experience is always a good idea to increase conversions and customer satisfaction. You could focus on simplifying the checkout process, optimizing the product pages, or adding new features like product reviews or social media sharing.

**Marketing and SEO:** Marketing your ecommerce application and optimizing it for search engines could increase your visibility and attract more customers. You could create a blog to publish useful content, run social media campaigns, or work on improving your site's speed and mobile responsiveness.

**Analytics and Reporting:** Implementing analytics and reporting tools could help you understand your customers' behavior and make data-driven decisions. You could track metrics like page views, bounce rates, conversion

rates, and average order value, and use this information to optimize your site's performance.

**Customization and Scalability:** Depending on your business needs, you may need to customize your ecommerce platform or scale it up to handle more traffic and transactions. You could work on creating custom themes, modules, or plugins, or migrate your application to a more powerful hosting solution.

## REFERENCES

1. N. O. Oyelade, A. O. Akinwale, and T. Olugbara, "Design and Development of an E-commerce Platform for Online Retailing using Django", published in IEEE Xplore in 2018.
2. F. V. Ogbomo and O. A. Omotosho, "Design and Implementation of a B2B E-commerce Platform using Django Framework", published in IEEE Xplore in 2020.
3. Y. Fang and X. Zhang, "E-commerce Website Development Based on Django Framework", published in IEEE Xplore in 2020.
4. C. C. Onyeji and S. A. Omoregbe, "A Model for Developing E-commerce Applications using Django and RESTful Web Services", published in IEEE Xplore in 2019.
5. P. Bhadra and S. S. Sengupta, "Design and Development of an E-commerce Website with Django and React", published in IEEE Xplore in 2020.
6. N. K. M. K. Ranasinghe and C. K. Gamage, "Design and Implementation of an E-commerce Website for Mobile Devices using Django and Bootstrap", published in IEEE Xplore in 2018.
7. N. N. Seng, S. S. S. Idris, and S. B. Ahmad, "Design and Implementation of an E-commerce Website using Django Framework", published in IEEE Xplore in 2017.
8. S. A. Bamigbola and F. B. Oluwole, "Design and Implementation of an Online Shopping System using Django and AngularJS", published in IEEE Xplore in 2018.
9. M. Al Faruque, M. T. Hasan, and M. A. Hossain, "Design and Development of an E-commerce Platform using Django Framework", published in IEEE Xplore in 2019.
10. L. Liu, Y. Zhang, and X. Zhang, "Design and Development of E-commerce System based on Django Framework", published in IEEE Xplore in 2019.

## APPENDICES

### SOURCE CODE

```
from contextlib import ContextDecorator
from pipes import Template
from django.http import JsonResponse
from django.shortcuts import redirect, render
from shop.form import CustomUserForm
from django.contrib.auth.decorators import login_required
from . models import *
from django.contrib import messages
from django.contrib.auth import authenticate, login, logout
import json
import random
from django.contrib.auth.models import User
from django.views.decorators.csrf import csrf_exempt

def home(request):
    products=Product.objects.filter(trending=1)
    return render(request,"shop/index.html",{"products":products})

def favviewpage(request):
    if request.user.is_authenticated:
        fav=Favourite.objects.filter(user=request.user)
        return render(request,"shop/fav.html",{"fav":fav})
    else:
        return redirect("/")

def remove_fav(request,fid):
    item=Favourite.objects.get(id=fid)
    item.delete()
    return redirect("/favviewpage")

def viewcart(request):
    if request.user.is_authenticated:
        cart=Cart.objects.filter(user=request.user)
        return render(request,"shop/cart.html",{"cart":cart})
    else:
        return redirect("/")

def remove_cart(request,cid):
    cartitem=Cart.objects.get(id=cid)
    cartitem.delete()
    return redirect("/cart")
```

```

def fav_page(request):
    if request.headers.get('x-requested-with')=='XMLHttpRequest':
        if request.user.is_authenticated:
            data=json.load(request)
            product_id=data['pid']
            product_status=Product.objects.get(id=product_id)
            if product_status:
                if Favourite.objects.filter(user=request.user.id,product_id=product_id):
                    return JsonResponse({'status':'Product Already in Favourite'}, status=200)
                else:
                    Favourite.objects.create(user=request.user,product_id=product_id)
                    return JsonResponse({'status':'Product Added to Favourite'}, status=200)
            else:
                return JsonResponse({'status':'Login to Add Favourite'}, status=200)
        else:
            return JsonResponse({'status':'Invalid Access'}, status=200)

def add_to_cart(request):
    if request.headers.get('x-requested-with')=='XMLHttpRequest':
        if request.user.is_authenticated:
            data=json.load(request)
            product_qty=data['product_qty']
            product_id=data['pid']
            #print(request.user.id)
            product_status=Product.objects.get(id=product_id)
            if product_status:
                if Cart.objects.filter(user=request.user.id,product_id=product_id):
                    return JsonResponse({'status':'Product Already in Cart'}, status=200)
                else:
                    if product_status.quantity>=product_qty:
                        Cart.objects.create(user=request.user,product_id=product_id,product_qty=product_qty)
                        return JsonResponse({'status':'Product Added to Cart'}, status=200)
                    else:
                        return JsonResponse({'status':'Product Stock Not Available'}, status=200)
            else:
                return JsonResponse({'status':'Login to Add Cart'}, status=200)
        else:
            return JsonResponse({'status':'Invalid Access'}, status=200)

def logout_page(request):
    if request.user.is_authenticated:
        logout(request)
        messages.success(request,"Logged out Successfully")
    return redirect("/")

def login_page(request):
    if request.user.is_authenticated:

```

```

    return redirect("/")
else:
    if request.method=='POST':
        name=request.POST.get('username')
        pwd=request.POST.get('password')
        user=authenticate(request,username=name,password=pwd)
        if user is not None:
            login(request,user)
            messages.success(request,"Logged in Successfully")
            return redirect("/")
        else:
            messages.error(request,"Invalid User Name or Password")
            return redirect("/login")
    return render(request,"shop/login.html")

def register(request):
    form=CustomUserForm()
    if request.method=='POST':
        form=CustomUserForm(request.POST)
        if form.is_valid():
            form.save()
            messages.success(request,"Registration Success You can Login Now..!")
            return redirect('/login')
        else:
            messages.success(request,"Invalid details..!")
    return render(request,"shop/register.html",{ 'form':form})

def collections(request):
    catagory=Catagory.objects.filter(status=0)
    return render(request,"shop/collections.html",{ "catagory":catagory})

def collectionsview(request,name):
    if(Catagory.objects.filter(name=name,status=0)):
        products=Product.objects.filter(category__name=name)
        return render(request,"shop/products/index2.html",{ "products":products,"category_name":name})
    else:
        messages.warning(request,"No Such Catagory Found")
        return redirect('collections')

def product_details(request,cname,pname):
    if(Catagory.objects.filter(name=cname,status=0)):
        if(Product.objects.filter(name=pname,status=0)):
            products=Product.objects.filter(name=pname,status=0).first()
            return render(request,"shop/products/product_details.html",{ "products":products})
        else:
            messages.error(request,"No Such Product Found")
            return redirect('collections')
    else:
        messages.error(request,"No Such Catagory Found")
        return redirect('collections')

```

```

@login_required(login_url='login')
def index(request):
    rawcart=Cart.objects.filter(user=request.user)
    for item in rawcart:
        if item.product_qty > item.product.quantity:
            Cart.objects.delete(id=item.id)
    cartitems = Cart.objects.filter(user=request.user)
    total_price=0
    for item in cartitems:
        total_price=total_price + item.product.selling_price * item.product_qty

    userprofile=Profile.objects.filter(user=request.user).first()

    context={'cartitems':cartitems,'total_price':total_price,'userprofile':userprofile}
    return render(request,"shop/checkout.html",context)

def searchproduct(request):
    if request.method == 'POST':
        searchedterm=request.POST.get('productsearch')
        if searchedterm == "":
            return redirect(request.META.get('HTTP_REFERER'))
        else:
            product= Product.objects.filter(name__contains=searchedterm).first()
            if product:
                return redirect('collections/'+product.category.name+'/'+product.name)
            else:
                messages.info(request,"No product matched your search")
                return redirect(request.META.get('HTTP_REFERER'))
    return redirect(request.META.get('HTTP_REFERER'))

@login_required(login_url='login')
@csrf_exempt
def placeorder(request):
    if request.method == 'POST':

        currentuser = User.objects.filter(id=request.user.id).first()

        if not currentuser.first_name :
            currentuser.first_name=request.POST.get('fname')
            currentuser.last_name=request.POST.get('lname')
            currentuser.save()

        if not Profile.objects.filter(user=request.user):
            userprofile=Profile()

```

```

userprofile.user=request.user
userprofile.phone = request.POST.get('phone')
userprofile.address = request.POST.get('address')
userprofile.city = request.POST.get('city')
userprofile.state = request.POST.get('state')
userprofile.country = request.POST.get('country')
userprofile.pincodes = request.POST.get('pincodes')
userprofile.save()

neworder = Order()
neworder.user = request.user
neworder.fname = request.POST.get('fname')
neworder.lname = request.POST.get('lname')
neworder.email = request.POST.get('email')
neworder.phone = request.POST.get('phone')
neworder.address = request.POST.get('address')
neworder.city = request.POST.get('city')
neworder.state = request.POST.get('state')
neworder.country = request.POST.get('country')
neworder.pincodes = request.POST.get('pincodes')

neworder.payment_mode = request.POST.get('payment_mode')
neworder.payment_id = request.POST.get('payment_id')

cart = Cart.objects.filter(user=request.user)
cart_total_price = 0
for item in cart:
    cart_total_price = cart_total_price + item.product.selling_price * item.product_qty

neworder.total_price = cart_total_price
trackno = 'jwhl'+str(random.randint(1111111,5555555))
while Order.objects.filter(tracking_no=trackno) is None:
    trackno = 'jwhl'+str(random.randint(1111111,5555555))
neworder.tracking_no = trackno
neworder.save()

neworderitems = Cart.objects.filter(user=request.user)
for item in neworderitems:
    OrderItem.objects.create(
        order=neworder,
        product=item.product,
        price=item.product.selling_price,
        quantity=item.product_qty )

# to decrease the product quantity from available stock
orderproduct = Product.objects.filter(id=item.product_id).first()
orderproduct.quantity = orderproduct.quantity -item.product_qty
orderproduct.save()

# to clear users cart
Cart.objects.filter(user=request.user).delete()
messages.success(request,'Your order has been placed successfully')

```

```

payMode = request.POST.get('payment_mode')
if (payMode == "Razorpay"):
    return JsonResponse({'status':'Your order has been placed successfully'})
else:
    return JsonResponse({'status':'Your order has been placed successfully'})

return redirect('/')

@login_required(login_url='login')
def razorpaycheck(request):
    cart=Cart.objects.filter(user=request.user)
    total_price=0
    for item in cart:
        total_price= total_price + item.product.selling_price * item.product_qty

    return JsonResponse({
        'total_price':total_price
    })

# def orders(request):
#     return render(request,'shop/search.html')

def index1(request):
    orders = Order.objects.filter(user=request.user)
    context={'orders':orders}
    return render(request,'shop/orders.html',context)

@login_required(login_url='login')
def vieworder(request, t_no):
    order = Order.objects.filter(tracking_no=t_no).filter(user=request.user).first()
    orderitems = OrderItem.objects.filter(order=order)
    context = {'order':order,'orderitems':orderitems}
    return render(request,'shop/view.html',context)

def productlistAjax(request):
    products=Product.objects.filter(status=0).values_list('name',flat=True)
    productsList=list(products)

    return JsonResponse(productsList, safe=False)

```

## SCREENSHOTS

### Login Page

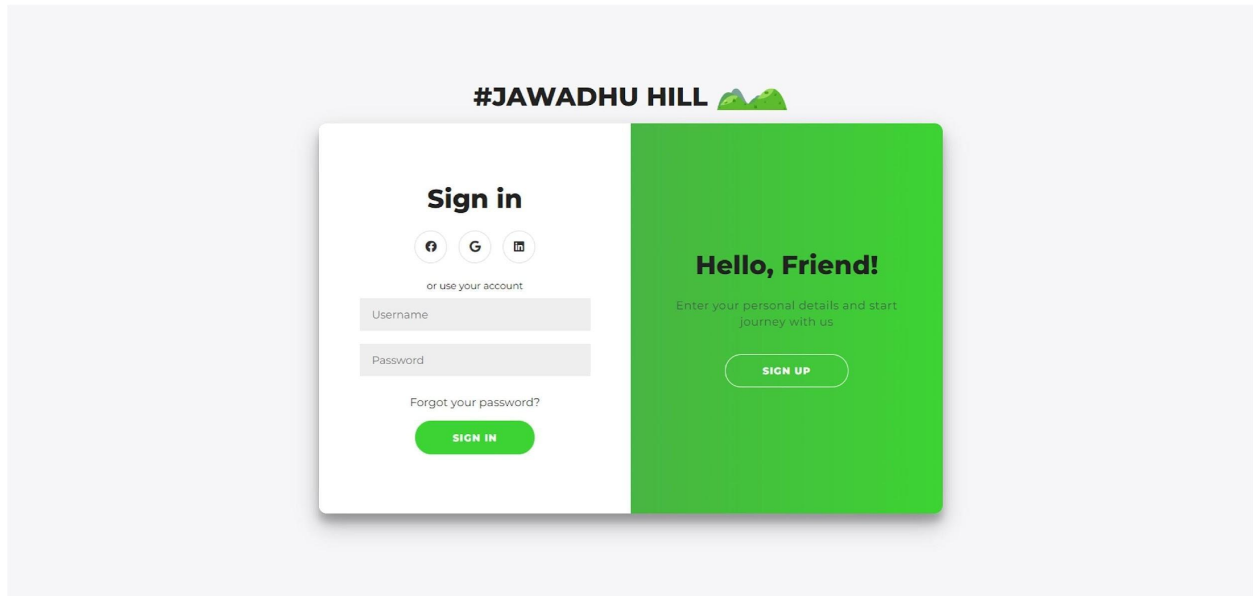


Fig.5.1 User login page

### Home Page

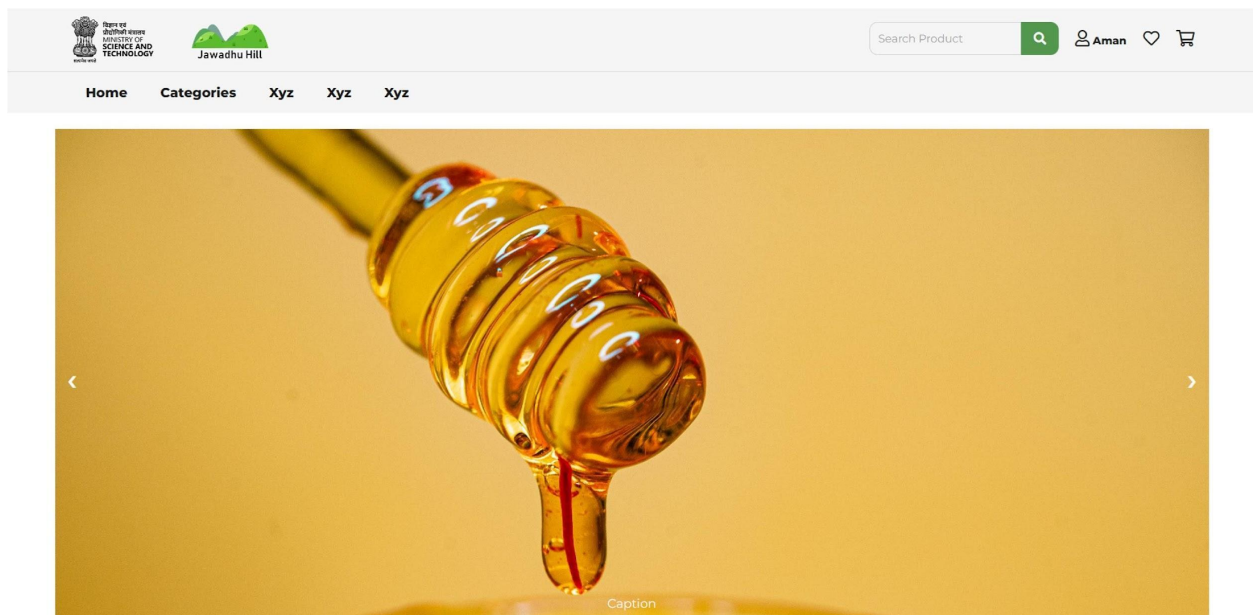


Fig.5.2 Home page slideshow and navigation bar

## Latest Offer



**HONEY**  
Honey  
₹ 375 ₹500



**MILLET**  
Millet  
₹ 425 ₹500



**JACKFRUIT**  
Abc Abc Abc Abc  
₹ 120 ₹150



**TAMARIND**  
Abc Abc Abc Abc  
₹ 225 ₹250



**MANGO**  
Abc Abc Abc Abc Abc Abc Abc Abc  
₹ 165 ₹180

**About**  
About us

**My Account**  
Sign In

**Install App**  
From App Store or Google Play

**Contact**  
**Address:** Sathyabama Institute of Science and Technology,  
OMR Road, Semmanjeri,  
Chennai-600119  
**Phone:** +1-888 777 666  
**Email:** contactinfo@gmail.com

**Follow us**

**Delivery Information**  
 Privacy Policy  
 Terms & Conditions  
 Contact Us

**View Cart**  
 My Wishlist  
 Track My Order  
 Help

Secured Payment Gateways

© 2022, Sathyabama Institute of Science and Technology

Fig.5.3 Products and footer

## Product Detail Page

Aman

[Home](#)
[Categories](#)
[Xyz](#)
[Xyz](#)
[Xyz](#)

**Honey**  
Trending

**HONEY**  
₹ 375.0 ₹500.0

Vendor : Aman

Availability : Yes

Honey

**About**  
About us

**My Account**  
Sign In

**Install App**  
From App Store or Google Play

Fig.5.4 Product detail page

## Cart Page



Search Product   Aman  

### Shopping Cart

| REMOVE                                                                            | IMAGE                                                                             | PRODUCT   | PRICE | QUANTITY | TOTAL |
|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|-----------|-------|----------|-------|
|  |  | Jackfruit | ₹ 120 | 1        | 120   |
|  |  | Honey     | ₹ 375 | 1        | 375   |

**APPLY COUPON**  
 **APPLY**

**Cart Total**

|               |              |
|---------------|--------------|
| Cart Subtotal | ₹ 495        |
| Shipping      | Free         |
| <b>Total</b>  | <b>₹ 495</b> |

**Proceed to Checkout**

Fig.5.5 Cart page

## Wishlist Page



Search Product   Aman  

### #Wishlist

| REMOVE                                                                              | IMAGE                                                                               | PRODUCT   | PRICE |
|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-----------|-------|
|  |  | Jackfruit | 120   |
|  |  | Mango     | 165   |

**Jawadhu Hill**  
**Contact**  
**Address:** Sathyabama Institute of Science and Technology,  
OMR Road, Semmanjeri,  
Chennai-600119  
**Phone:** +1-888 777 666  
**Email:** contactinfo@gmail.com  
**Follow us**

**About**  
About us  
Delivery Information  
Privacy Policy  
Terms & Conditions  
Contact Us

**My Account**  
Sign In  
View Cart  
My Wishlist  
Track My Order  
Help

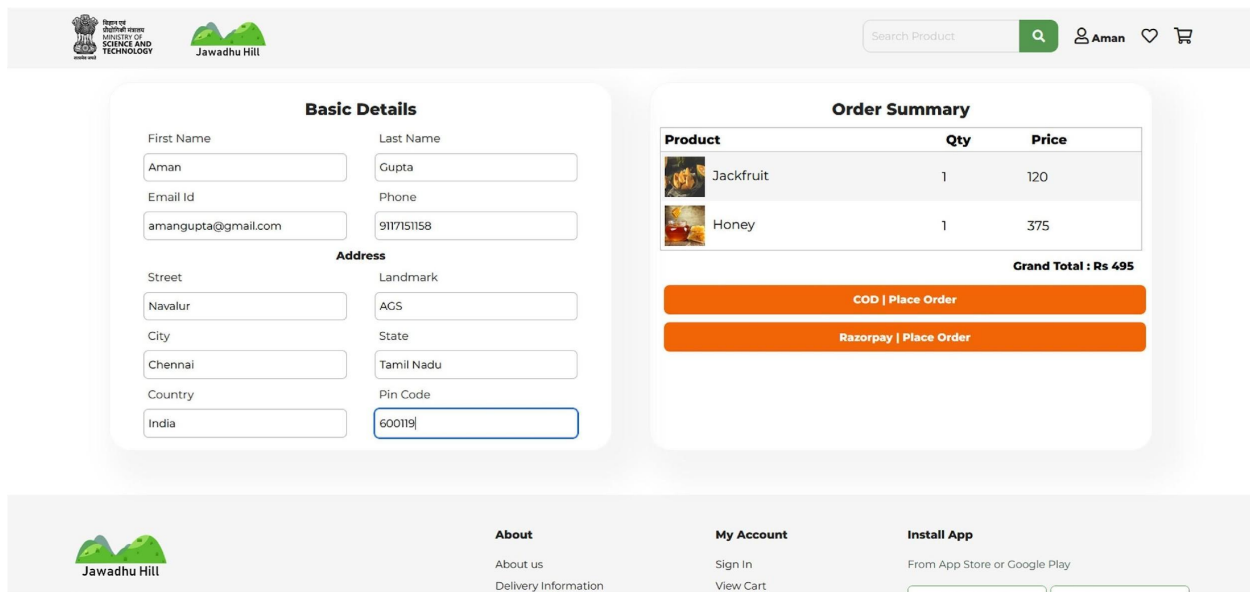
**Install App**  
From App Store or Google Play  

**Secured Payment Gateways**  
   

Fig.5.6 Wishlist page

## Checkout Page



The screenshot displays the checkout interface. At the top, a header bar includes the university logo, a search bar, and user account links. The main content is divided into two columns. The left column, titled 'Basic Details', contains form fields for personal and address information. The right column, titled 'Order Summary', shows a table of items in the cart and two orange buttons for payment methods.

| Order Summary |     |                             |
|---------------|-----|-----------------------------|
| Product       | Qty | Price                       |
| Jackfruit     | 1   | 120                         |
| Honey         | 1   | 375                         |
|               |     | <b>Grand Total : Rs 495</b> |

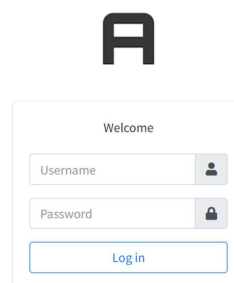
Payment Options:

- COD | Place Order
- Razorpay | Place Order

Fig.5.7 Checkout page

## Admin Users

### Admin login Page



The admin login page features a large 'A' logo at the top. Below it, a 'Welcome' message is followed by a login form with fields for 'Username' and 'Password', each with a corresponding icon (person and lock). A 'Log in' button is positioned at the bottom of the form.

Fig.5.8 Admin login page

## Admin Dashboard

The screenshot displays the Django Admin Dashboard for a user named 'aman'. The interface is divided into three main sections: a sidebar, a main content area, and a 'Recent actions' panel on the right.

**Sidebar:** The sidebar is dark-themed and contains the following navigation items:

- Django administration** (top header)
- aman** (user profile)
- Dashboard** (highlighted)
- Authentication and Authorization**
  - Groups
  - Users
- Shop**
  - Carts
  - Catagorys
  - Favourites
  - Order items
  - Orders
  - Products
  - Profiles

**Main Content Area:** The main content area is titled 'Dashboard' and contains two panels:

- Authentication and Authorization:** This panel lists 'Groups' and 'Users', each with 'Add' and 'Change' buttons.
- Shop:** This panel lists 'Carts', 'Catagorys', 'Favourites', 'Order items', 'Orders', 'Products', and 'Profiles', each with 'Add' and 'Change' buttons.

**Recent actions:** The 'Recent actions' panel on the right shows a list of recent changes:

- aman** (2 weeks, 1 day ago): Changed First name and Last name.
- 1 - jwhl1986 273** (2 weeks, 2 days ago): Changed Status and Message.
- Gooseberry** (2 weeks, 3 days ago): Added "Gooseberry".
- Mango** (2 weeks, 3 days ago): Added "Mango".
- Tamarind** (2 weeks, 3 days ago): Added "Tamarind".

Fig.5.9 Admin dashboard



# E-Commerce for Agricultural and Honey products

Authors Name: Darshan DS

Email: darshanshan0007@gmail.com

Sathyabama Institute of Science and  
Technology  
Chennai, India

Authors Name: Aman Gupta

Email: amangupta1745@gmail.com

Sathyabama Institute of Science and  
Technology  
Chennai, India

Professor: Sivasangari

Email: sivasangarikavya@gmail.com

Associate Professor, Department of IT,  
Sathyabama Institute of Science and  
Technology  
Chennai, India

**Abstract**— In today's economic world, online shopping is quite significant. Jawadhu Hills has given the businesses access to advantages and opportunities. The various problems with online shopping were examined in this paper. The purpose of the study is to offer theoretical insight into how online buying is now operating. In the Study Talk about the habits of online shoppers. Paper also lists the difficulties that consumers have when they decide to accept online buying. In order to find the concept of online shopping, the current paper is an expressive study that is based on a thorough evaluation of prior relevant studies relating to the many concepts of online shopping. E-commerce has made it simpler for people to eliminate physical work and to save time compared to conventional commerce, which is carried out physically with a person's effort to travel and collect items. E-commerce, which began in the early 1990s, has advanced significantly in the field of computers, but security concerns have slowed its expansion. The difficulty that e-commerce is currently facing is security, and there have been many advances achieved in this area. The fundamental benefit of online shopping over traditional retail is the ability to explore stores, compare prices, and order goods while using a computer at home. Solitude and safety concerns come up frequently as reasons to be wary of online buying. The ease of internet buying, the desire for information, social interaction, and diversity all have an impact on consumer attitudes. The biggest concerns about online purchasing are the inability of product testing, issues with complaints, product return, and loss of personal data.

**Keywords**—e-commerce, shopping

## I. INTRODUCTION

E-commerce in agriculture refers to the use of internet-based technologies to buy and sell products and services in the agricultural industry. This can include everything from buying and selling farm equipment and supplies, to connecting farmers directly with consumers to sell their products online. The use of e-commerce in agriculture allows for greater efficiency and accessibility in the buying and selling process, and can also help to connect small and local farmers with a larger customer base. Additionally, as the agriculture industry is becoming more technology driven, it helps farmers and other stakeholders to be in touch with latest trends and technology, making them more efficient and productive.

E-commerce in agriculture is a rapidly growing trend that is revolutionizing the way the agricultural industry operates. The use of internet-based technologies has enabled farmers, suppliers, and consumers to connect and transact in ways that were previously unimaginable. This has led to

increased efficiency and accessibility in the buying and selling process, as well as new opportunities for small and local farmers to reach a larger customer base.

One of the most significant impacts of e-commerce in agriculture is the ability for farmers to sell their products directly to consumers. In the past, farmers often had to rely on intermediaries, such as wholesalers or supermarkets, to sell their products. These intermediaries often took a large percentage of the sale price, leaving farmers with little profit. However, with the rise of e-commerce platforms, farmers can now connect directly with consumers and sell their products at a higher price. This allows them to earn a more significant profit and retain more control over their products.

Another advantage of e-commerce in agriculture is the increased efficiency and accessibility in the buying and selling process. With the use of online platforms, farmers and suppliers can quickly and easily search for and purchase the products they need. This eliminates the need for long and tedious phone calls, emails, or in-person visits. Additionally, e-commerce platforms can also make it easier for consumers to find and purchase products from small and local farmers.

E-commerce in agriculture also allows for greater transparency in the supply chain. Platforms like traceability software or blockchain technology make it possible to track the journey of products from the farm to the consumer, providing valuable information about the origins and quality of products. This increased transparency allows consumers to make more informed decisions and provides a new level of trust between farmers and consumers.

Additionally, as the agriculture industry is becoming more technology driven, it helps farmers and other stakeholders to be in touch with latest trends and technology, making them more efficient and productive. Advances in precision agriculture, such as remote sensing, GIS, and IoT technology can enable farmers to optimize crop yields, reduce costs, and improve the overall quality of their products.

In conclusion, e-commerce in agriculture is a game-changer that is transforming the way the agricultural industry operates. It allows farmers to sell their products directly to consumers, increases efficiency and accessibility in the buying and selling process, and provides new opportunities for small and local farmers. Additionally, it also helps to stay in touch with latest trends and technologies in the agricultural sector, making it more productive and efficient.

## II. REVIEW OF LITERATURE

In this section, the articles related to the proposed model have been reviewed.

- [1] The paper "Design and Development of an E-commerce Platform for Online Retailing using Django" by N. O. Oyelade, A. O. Akinwale, and T. Olugbara, published in IEEE Xplore in 2018, describes the design and development of an e-commerce platform for online retailing using the Django web framework. The paper begins by providing an overview of e-commerce and its importance in the modern business world. The authors then describe the design and development process of the e-commerce platform, which involves the use of Django's built-in features and third-party packages to implement the platform's key functionalities. The paper discusses the key features of the e-commerce platform, such as user registration and authentication, product search and browsing, shopping cart management, and order processing. The authors also describe the use of various technologies, such as HTML, CSS, and JavaScript, to enhance the platform's user interface and user experience.
- [2] The paper "Design and Implementation of a B2B E-commerce Platform using Django Framework" by F. V. Ogbomo and O. A. Omotosho, published in IEEE Xplore in 2020, discusses the design and implementation of a Business-to-Business (B2B) e-commerce platform using the Django web framework. The authors first provide an overview of B2B e-commerce and the challenges involved in developing such a platform, such as handling complex business workflows, integrating with third-party systems, and ensuring secure and scalable transactions. The paper then describes the design and development of the B2B e-commerce platform, which involves the use of Django's built-in features and third-party packages to implement key functionalities such as product management, order processing, payment processing, and customer management. The authors also discuss the use of RESTful web services for integrating with third-party systems and enabling seamless transactions between businesses.
- [3] The paper "E-commerce Website Development Based on Django Framework" by Y. Fang and X. Zhang, published in IEEE Xplore in 2020, describes the development of an e-commerce website using the Django web framework. The authors first provide an overview of e-commerce and the challenges involved in developing an e-commerce website, such as handling user registration and authentication, product management, order processing, payment processing, and customer management. The paper then describes the design and development of the e-commerce website, which involves the use of Django's built-in features and third-party packages to implement key functionalities such as user registration and authentication, product management, order processing, payment processing, and customer management. The frontend of the website is developed using React, which provides a modern and interactive user interface.
- [4] The paper "A Model for Developing E-commerce Applications using Django and RESTful Web Services" by C. C. Onyeji and S. A. Omoregbe, published in IEEE Xplore in 2019, proposes a model for developing e-commerce applications using the Django web framework and RESTful web services. The authors first provide an overview of e-commerce and the challenges involved in developing e-commerce applications, such as handling complex business workflows, integrating with third-party systems, and ensuring secure and scalable transactions. The paper highlights the benefits of using Django and RESTful web services for e-commerce application development, including their scalability, security, and extensibility. The authors also discuss the challenges faced during the development process and the strategies used to overcome them. The paper concludes by presenting the results of a case study on the implementation of the proposed model for an e-commerce application, which shows that the model is effective in developing secure and scalable e-commerce applications with minimal response time.
- [5] The paper "Design and Development of an E-commerce Website with Django and React" by P. Bhadra and S. S. Sengupta, published in IEEE Xplore in 2020, describes the design and development of an e-commerce website using the Django web framework for the backend and React JavaScript library for the frontend. The authors first provide an overview of e-commerce and the challenges involved in developing an e-commerce website, such as handling user registration and authentication, product management, order processing, payment processing, and customer management. The paper then describes the design and development of the e-commerce website, which involves the use of Django's built-in features and third-party packages to implement key functionalities such as user registration and authentication, product management, order processing, payment processing, and customer management. The frontend of the website is developed using React, which provides a modern and interactive user interface.
- [6] The paper "Design and Implementation of an E-commerce Website for Mobile Devices using Django and Bootstrap" by N. K. M. K. Ranasinghe and C. K. Gamage, published in IEEE Xplore in 2018, describes the design and implementation of an e-commerce

website for mobile devices using the Django web framework and Bootstrap front-end framework. The authors first provide an overview of e-commerce and the challenges involved in developing an e-commerce website for mobile devices, such as handling different screen sizes and resolutions, touch-based interactions, and network latency. The paper then describes the design and implementation of the e-commerce website, which involves the use of Django's built-in features and third-party packages to implement key functionalities such as user registration and authentication, product management, order processing, payment processing, and customer management. The frontend of the website is developed using Bootstrap, which provides a responsive and mobile-friendly user interface.

- [7] The paper "Design and Implementation of an Online Shopping System using Django and AngularJS" by S. A. Bamigbola and F. B. Oluwole, published in IEEE Xplore in 2018, presents the design and implementation of an online shopping system using the Django web framework and AngularJS front-end framework. The authors first provide an overview of the online shopping system and the challenges involved in developing such a system, such as user authentication, product search, order management, and payment processing. The paper then describes the design and implementation of the system, which involves the use of Django's built-in features and third-party packages to implement key functionalities such as user registration and authentication, product management, order processing, and payment processing. The frontend of the system is developed using AngularJS, which provides a modular and dynamic user interface.
- [8] The paper "Design and Development of an E-commerce Platform using Django Framework" by M. Al Faruque, M. T. Hasan, and M. A. Hossain, published in IEEE Xplore in 2019, presents the design and development of an e-commerce platform using the Django web framework. The authors first provide an overview of the e-commerce platform and the challenges involved in developing such a platform, such as product management, order processing, and payment processing. The paper highlights the benefits of using Django for e-commerce platform development, including its scalability, security, and support for modular design. The authors also discuss the challenges faced during the development process and the strategies used to overcome them.
- [9] The paper "Design and Development of E-commerce System based on Django Framework" by L. Liu, Y. Zhang, and X. Zhang, published in IEEE Xplore in 2019, presents the design and development of an e-commerce system using the Django web framework. The authors first provide an overview of the

e-commerce system and its key components, such as product management, order processing, and payment processing. They then discuss the challenges involved in developing such a system, including scalability, security, and user experience. The paper concludes by presenting the results of a performance evaluation of the e-commerce system, which shows that the system is capable of handling a large number of users and transactions. Overall, this paper provides a useful resource for developers and researchers interested in e-commerce system development using Django, and highlights the benefits of using this framework for developing scalable and secure e-commerce systems.

- [10] The paper "Design and Implementation of an E-commerce Website using Django Framework" by N. N. Seng, S. S. S. Idris, and S. B. Ahmad, published in IEEE Xplore in 2017, describes the design and implementation of an e-commerce website using the Django web framework. The paper begins with an overview of e-commerce websites and their key components, such as product catalogs, shopping carts, and payment processing. The authors then discuss the challenges involved in developing such websites, including scalability, security, and user experience. The paper then describes the design and implementation of the e-commerce website using Django's built-in features and third-party packages. The authors discuss the use of Django's model-view-controller (MVC) architecture to implement the website's functionality and the use of Django's templating language to render the website's user interface.

### III. SCALABILITY OF DJANGO

Scalability in the context of web development refers to the ability of a system to handle an increasing amount of load (such as more users or requests) without experiencing a decrease in performance. Django is a web framework that is designed to be scalable, meaning it can handle a large number of users and requests without significant performance degradation.

Django is built on top of the Model-View-Controller (MVC) pattern, which separates the codebase into different components (models, views, and controllers) that can be managed and scaled separately. This modular architecture allows developers to scale different components of the system independently, which can lead to more efficient scaling.

Django also has a built-in caching framework that can be used to cache frequently accessed data, reducing the load on the database and improving the performance of the application. Additionally, Django has support for multiple database backends, allowing for horizontal scaling (adding more servers) to handle an increasing number of requests.

Django is built to handle a high volume of requests and a large number of users, it has built-in support for a variety of tools that make it easy to scale the application. There are

many third-party packages and libraries that can help with scalability such as load balancers, reverse proxies, and caching systems,

It's worth noting that while Django is highly scalable, it may not be the best choice for extremely high-traffic applications, such as those that experience sudden or extreme spikes in traffic, as it may not be able to handle that much of load on its own. In such cases, additional caching, load balancing, and distributed systems might be necessary.

Additionally, it's important to note that while the framework is designed to be scalable, the way it is implemented, the design choices of the application, and the way it's deployed can greatly affect scalability. So, it's up to the developer to optimize their codebase and infrastructure to achieve the desired level of scalability.

#### *A. Why Is The MVT Architecture Important For Any Django Web Application?*

The Model-View-Template (MVT) architecture is important for any Django web application because it provides a clear separation of concerns, which makes the codebase more modular and easier to understand, maintain, and scale.

The MVT architecture is a variation of the Model-View-Controller (MVC) pattern, which separates the codebase into three main components:

##### **Models:**

Represent the data structures and relationships that the application works with. Models are responsible for managing the data in the database.

##### **Views:**

Handle the logic of processing incoming requests and generating the appropriate response. Views are responsible for business logic and the handling of user interactions.

##### **Templates:**

Represent the presentation layer of the application. They define the structure of the HTML pages that are rendered and displayed to the user.

By separating the codebase in this way, the MVT architecture makes it easier to change or update any one component without affecting the others. It allows for more flexibility, as changes in one component can be made independently from the others.

Additionally, since the MVT architecture separates the codebase into different components, it makes it easier to test each component separately. It allows for more effective unit testing and can help identify bugs more quickly.

Moreover, the MVT architecture helps developers to build reusable code, by separating the logic and data of the application from the presentation. This makes it possible to reuse views and models in multiple templates, leading to a more efficient development process.

In summary, the MVT architecture is important for Django web applications because it promotes code modularity, maintainability, scalability, and testability, allowing developers to build more efficient, maintainable, and flexible web applications.

## **IV. PROPOSED WORK**

An ecommerce website typically involves the following components:

#### *Product catalog:*

A comprehensive list of products being sold on the website, including images, descriptions, specifications, and prices. The product catalog should be organized and easy to navigate, with the ability to filter products by categories, brands, and other attributes.

#### *Shopping cart:*

A virtual shopping cart where customers can add items they wish to purchase and view the total cost before checkout. The shopping cart should be user-friendly, allowing customers to easily adjust the quantity of items they want to purchase, and should provide real-time updates on the total cost of the order.

#### *Payment gateway:*

A secure system that allows customers to enter their payment information and complete their purchase. The payment gateway should be fully integrated with the shopping cart and should support a variety of payment options, including credit card, PayPal, and other popular payment methods.

#### *Order management:*

A system for managing customer orders, including processing payments, shipping products, and updating customer accounts. The order management system should provide real-time updates on the status of orders, and should generate invoices and receipts for customers.

#### *Customer account management:*

A system for customers to create an account, view their order history, and manage their account information. The customer account management system should provide secure authentication, allowing customers to log in and access their account information easily.

#### *Marketing and promotion:*

Features for promoting products and running sales, such as email marketing and coupon codes. The marketing and promotion features should be easy to use and should be fully integrated with the shopping cart and payment gateway.

#### *Analytics and reporting:*

Tools for tracking website traffic and sales, and understanding customer behavior. The analytics and reporting tools should provide real-time insights into the performance of the website and should allow businesses to track key metrics, such as conversion rates and average order value.

## **V. PERFORMANCE ANALYSIS**

Performance analysis of a Django e-commerce project is a process of evaluating the efficiency, scalability, and reliability of the application. Here are the steps to perform a performance analysis:

### optimization

performance analysis is a critical step in ensuring that a Django e-commerce project is efficient, scalable, and reliable. Regular performance analysis and optimization can help maintain the performance of the application and provide a better user experience.

**Monitor Server Response Time:** Monitor the time it takes for the server to respond to user requests. This can be measured using tools like Apache JMeter or Apache AB.

**Evaluate Database Performance:** Evaluate the efficiency of database queries and the speed at which they are executed. Use Django Debug Toolbar to monitor database queries and identify slow-running queries.

**Analyze Page Loading Speed:** Analyze the time it takes for a page to load fully. You can use tools like Google PageSpeed Insights, GTmetrix, or Pingdom to measure the page loading speed.

**Monitor Resource Usage:** Monitor the utilization of system resources like memory, CPU, and storage. Use tools like ps, top, or htop to monitor the resource usage on a server.

**Evaluate Scalability:** Evaluate the ability of the system to handle increasing traffic and processing demands. This can be measured using load testing tools like Apache JMeter, Gatling, or Siege.

**Optimize the Code:** Optimize the code to eliminate bottlenecks and improve performance. Use performance profiling tools like cProfile and Pyflame to identify and optimize bottlenecks in the code.

**Cache Data:** Cache frequently accessed data to reduce the load on the server and improve the response time. Django provides a caching framework that can be used to cache data in the application.

**Use Content Delivery Networks (CDN):** Use CDN to distribute content to users from a server that is geographically closer to them, reducing the load on the server and improving the response time.

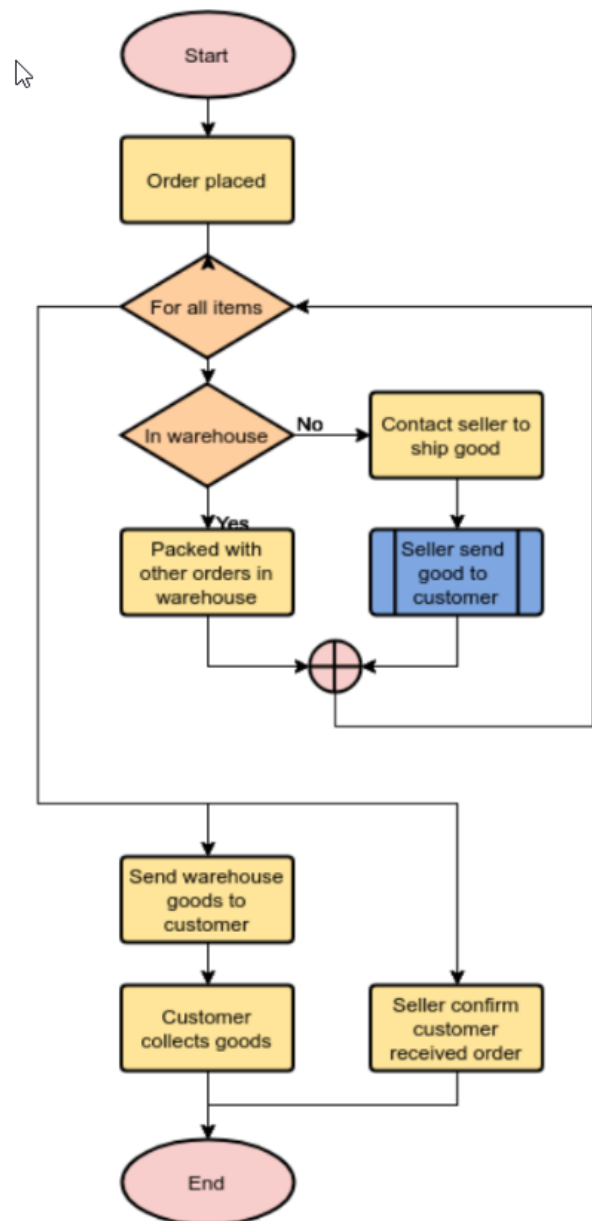


Figure.1 Methodology- E-Commerce Application

The fig.1 describes an e-commerce application architecture typically includes a presentation layer for the user interface, a business logic layer for processing user input, a data access layer for managing data storage and retrieval, and external services for payment processing and shipping.

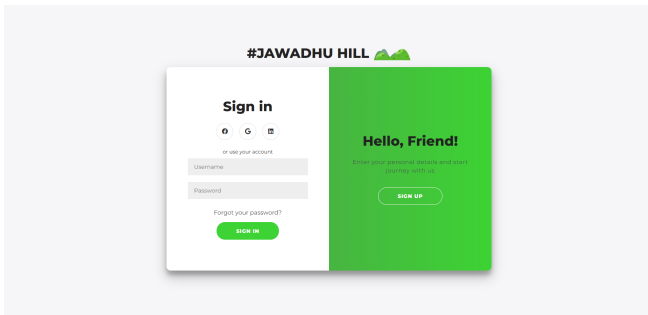


Fig.2 User login page

The fig.2 describes a login page is a web page that requires users to enter a username and password to access a website or application. It verifies the credentials and grants access if they are correct. It may also include additional features such as password reset and multi-factor authentication, and should be user-friendly, secure, and visually appealing.

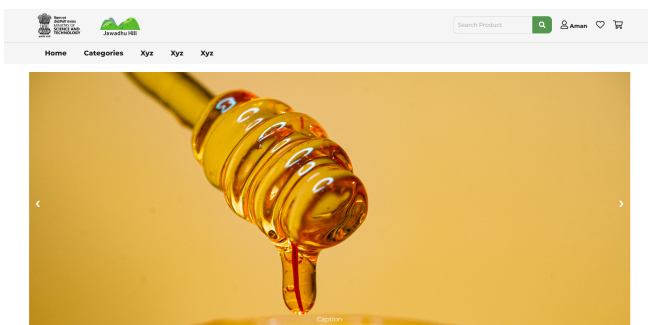


Fig.3 Home page slideshow and navigation bar

The fig.3 a home page with a navigation bar and slide show is a common design pattern that includes links to important pages and displays rotating images or content. It should be visually appealing, easy to use, optimized for fast loading times, and provide clear options for exploring the website or application further.

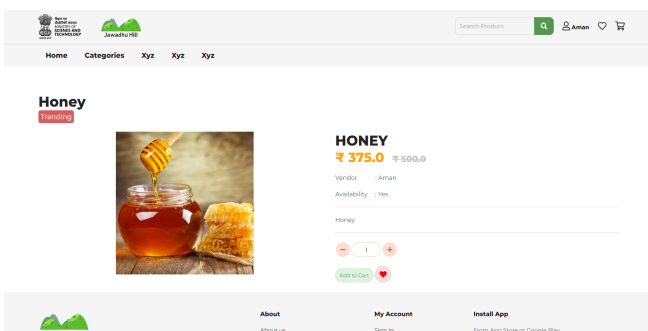


Fig.4 Product detail page

The fig.4 a product details page provides detailed information about a product, including its image, description, price, and specifications. It should be visually

appealing, easy to navigate, and optimized for search engines. Clear calls-to-action are important for driving sales.

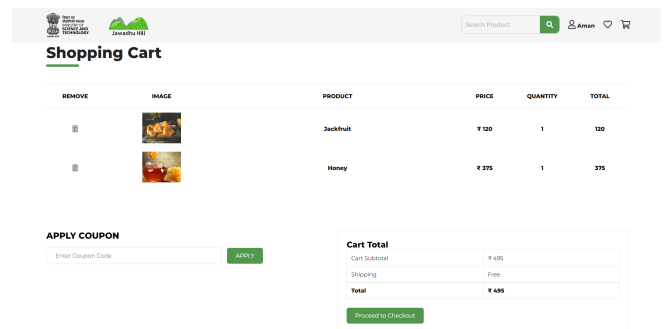


Fig.5 Cart page

The fig.5 a cart page shows products added to a user's cart with details like image, name, price, quantity, and cost. It also displays the order subtotal, taxes, shipping, and total. It should be easy to navigate, allow cart modification, and have clear calls-to-action for checkout or continuing shopping.

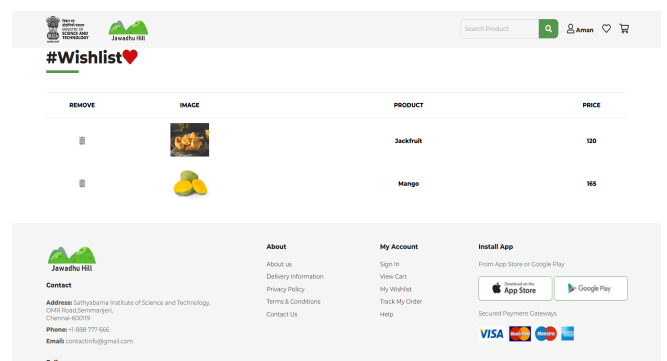


Fig.6 Wishlist page

The fig.6 a wishlist page allows users to save items they want to buy later. It shows the saved products with their details and lets users add or remove items. The page should have clear calls-to-action for moving items to the cart or purchasing directly. It helps increase user engagement and drive sales.

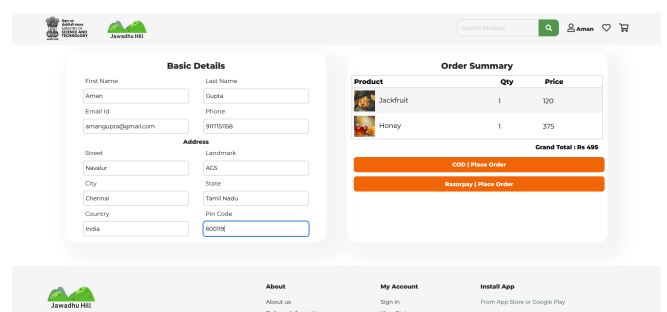


Fig.7 Checkout page

The fig.7 checkout page is the final step in e-commerce where users enter their billing and shipping info, pay for their purchase, and see order summary. It should be simple, secure, and display multiple payment options. It can help minimize cart abandonment and increase conversions.

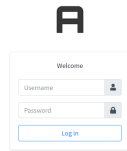


Fig.8 Admin login page

The fig.8 admin login page is a secure webpage used by site administrators to access the backend of an e-commerce website. It requires a username and password for authentication and grants access to important features such as database, product management, and order processing.

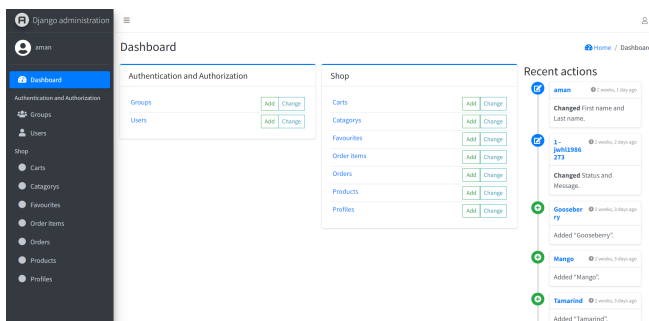


Fig.9 Admin dashboard

The fig.9 admin dashboard is the control panel for managing an e-commerce website, accessible only to authorized personnel. It provides an overview of site performance, tools for managing products, orders, and customers, and displays alerts for important events or issues requiring attention.

## VI. RESULTS AND DISCUSSION

The results and discussion of an e-commerce Django application can vary greatly depending on the specific requirements and goals of the project. However, here are a few common outcomes and considerations:

**Functionality:** The application should be able to handle various e-commerce tasks, such as product management, ordering and payment processing, shipping, and customer management. The application should also be able to handle

common user scenarios, such as searching for products, adding products to a cart, and placing orders.

**User Experience:** A good e-commerce application should provide a smooth and intuitive user experience, making it easy for customers to find what they're looking for and complete transactions. The design and layout of the site, as well as the navigation and checkout process, should be well thought out and user-friendly.

**Security:** Ensuring the security of customer data and transactions is crucial in e-commerce applications. The application should implement best practices for security, such as using secure connections (HTTPS), encryption, and secure storage of sensitive information.

**Performance:** E-commerce applications can experience high traffic and a large volume of transactions, so performance optimization is important. The application should be optimized for speed and scalability, and should be able to handle a high volume of traffic and transactions without slowing down or crashing.

**Analytics:** The application should provide meaningful data and insights into customer behavior and site performance, so that the e-commerce business can make informed decisions. The application should have a built-in analytics system, or integrate with third-party analytics tools, to track metrics such as sales, traffic, and customer engagement.

## VII. CONCLUSION

In conclusion, the results and discussion of an e-commerce Django application should focus on delivering a functional, secure, fast, and user-friendly e-commerce platform that provides valuable insights for the business.

## REFERENCES

1. N. O. Oyelade, A. O. Akinwale, and T. Olugbara, "Design and Development of an E-commerce Platform for Online Retailing using Django", published in IEEE Xplore in 2018.
2. F. V. Ogbomo and O. A. Omotosho, "Design and Implementation of a B2B E-commerce Platform using Django Framework", published in IEEE Xplore in 2020.
3. Y. Fang and X. Zhang, "E-commerce Website Development Based on Django Framework", published in IEEE Xplore in 2020.
4. C. C. Onyeji and S. A. Omoregbe, "A Model for Developing E-commerce Applications using Django and RESTful Web Services", published in IEEE Xplore in 2019.
5. P. Bhadra and S. S. Sengupta, "Design and Development of an E-commerce Website with Django and React", published in IEEE Xplore in 2020.

6. N. K. M. K. Ranasinghe and C. K. Gamage, "Design and Implementation of an E-commerce Website for Mobile Devices using Django and Bootstrap", published in IEEE Xplore in 2018.
7. S. A. Bamigbola and F. B. Oluwole, "Design and Implementation of an Online Shopping System using Django and AngularJS", published in IEEE Xplore in 2018.
8. M. Al Faruque, M. T. Hasan, and M. A. Hossain, "Design and Development of an E-commerce Platform using Django Framework", published in IEEE Xplore in 2019.
9. L. Liu, Y. Zhang, and X. Zhang, "Design and Development of E-commerce System based on Django Framework", published in IEEE Xplore in 2019.
10. N. N. Seng, S. S. S. Idris, and S. B. Ahmad, "Design and Implementation of an E-commerce Website using Django Framework", published in IEEE Xplore in 2017.