# Chapter 1: Introduction, Problem Definition/Statement and Proposed Model

## 1.1 INTRODUCTION

New technology and techniques are implemented in Ecommerce websites to visually present information due to significant growth in shopping worldwide. A web application that also allows visually impaired and blind users to access and shop. The system gives voice instructions with welcome messages which assist users to use the website and an input is requested from the user. The user is required to hover over the entire page using the cursor to access different voice messages for various fields. The user provides input using the keyboard keys thus taking the process forward. The website also provides recommendations based on customer selection using collaborative filtering algorithms. It is a process of filtering information or patterns using techniques involving collaboration of data gathered from different agents and data sources. The system processes the voice output, and asks for the input by pressing the button of their desired choices.

This will help them to buy their desired things online. Every kind of instruction will give them in mode of audio they need to hear and follow the instructions.

## 1.2 PROBLEM DEFINITION/STATEMENT

According to WHO, globally 1 billion people suffer from partial or severe vision impairment or blindness. When the number is so huge, it is needless to say that your e-commerce store design should consider web accessibility for shoppers with partial or complete blindness. Consider a blind individual who lives alone, but wants to make a purchase from an ecommerce site. He doesn't have anybody there by his side, at the moment, to either guide him on what to do on the site, or make the purchase for him. Traditionally users type in a query into a search box to fetch answers this will some time take more time and not too friendly like consumers to shop when they're cooking, multitasking, or even driving.

It is also difficult for blind people to choose clothes with different colors or they find it difficult to shop online. People with complete blindness use screen readers to access the web content. Screen readers are software that allows people with blindness to read the web content either with a Braille display or a sound synthesizer. Now it often happens that the screen readers don't work properly due to browser compatibility issues. People with visual impairments find it difficult to differentiate between the foreground and background when they encounter low-contrast color schemes. With this in mind, how can any seller make sure that e-commerce store stands out from the crowded competition? Further to it, to make online shopping more convenient, voice-enabled searches are becoming immensely popular and are being widely adopted.

## 1.3 BACKGROUND STUDY

E-commerce is fast gaining ground as an accepted and used business paradigm. More and more business houses are implementing web sites providing functionality for performing commercial transactions over the web. It is reasonable to say that the process of shopping on the web is becoming commonplace. The objective of this project is to develop a general-purpose e-commerce store where any product (such as books, groceries, cloths, computers, mobile phones, electronic items, and home appliances) can be bought from the comfort of home through the Internet. However, for implementation purposes, this paper will deal with an online ecommerce store.

An online store is a virtual store on the Internet where customers can browse the catalog and select products of interest. The selected items may be collected in a shopping cart. At checkout time, the items in the shopping cart will be presented as an order. At that time, more information will be needed to complete the transaction.

Usually, the customer will be asked to fill or select a billing address, a shipping address, a shipping option, and payment information such as a credit card number but in our case we have one added feature of voice based module which can help to do shopping of the product till adding to cart. An email notification is sent to the customer as soon as the order is placed.

## 1.4 PROPOSED SOLUTION

A system is developed to assist people by automatically recognizing product patterns and colors. The system processes the voice output, and asks for the input by pressing the button of their desired choices. This will help them to buy their desired things online. Every kind of instruction will give them in mode of audio they need to hear and follow the instructions. Just try to keep our online store website as simple and minimalistic as possible. In short, our website is less flashy in terms of graphics. We wil try to use high-contrast color schemes when choosing colors for your online store. We can take the help of tools like Wave and CheckMyColours to figure which component of your online store is high contrast or low contrast. A rise in adoption of smart speakers by consumers, plus the growing usage in voice-enabled searches, combined with an increasing inclination to use voice assistants for shopping, voice-based eCommerce will be the next major disruption for online retail. This is just the right time for retailers to focus on voice search optimization strategy in their eCommerce business to stay ahead of the curve. When users can do pretty much everything by their voice, they are more likely to spend less time on device keyboards. From how the things are shaping currently, voice commerce is all poised to gain more traction.

## 1.5 MODULE DESCRIPTION

**Products module:** The shop owner has the authority to add or delete items from the web mall. Shop owner also has the right to modify those details. This includes the information

about a particular product, such as product number, item, name, category, images of products, description, features, constraints of products, product price which are to be displayed on the website.

**Orders Module:** When a customer goes through checkout, the information on their order is automatically transferred to the Orders section for you to keep track of. In the administration, you can view all of the orders made on their site, manually add orders, or edit the details of existing orders. The Orders section is located under Sales > Orders. On this page, every order ever made from the store is listed in detail.

**Categories Module:** In the default theme of the store front, parent categories are listed in the top menu of the home page, and on the left side of product pages. This navigational feature is used to guide customers to similar products within the same category. Exposing customers to different products within a category lets the customer compare the similarities and differences between products to make the most informed purchase. When adding products to the store, you will be asked for a product category to sort them in. It is a good idea to establish these categories before adding products, to save yourself the trouble of adding the category name to the products later.

**Shopping module:** The customer can select the items and put it into a cart and as soon as he clicks on finish, the total amount to be paid is displayed and also provides a discount if any. The customer can pay cash on delivery or else he can pay by entering his credit card number and address to which the items have to be shipped.

**Customer Module:** Shop owners should know who their customers are and how to manage their information. In the administration, customer information will need to be stored efficiently to remember any transactions made with their account. To access customer information, you can log into the administration panel of the store. The Customer management sections are located under Sales> Customers.

**Billing Module:** Here customers can view billing reports and order reports. Admin module: Administrator has full permission to access this web site. Administrators or employees can handle customer details, and he can communicate with customers through mail. If a customer forgets his password, the employee can reset his password.

**Report module:** This module deals with report management of the entire system.

**Coupons Module:** You may create and designate coupons to specific products or product categories under Sales > Coupons. The Coupons page will display a list of all the coupons created in the administration.

## 1.6 PURPOSES

The project is about to handle all the information of the shop regarding members. Also it manages resources which were managed and handled by manpower previously. The

main purpose of the project is to integrate distinct sections of the shop into consistent manner so that complex functions can be handled smoothly. The project aims at the following matters:

- Automation of product manipulation.
- Buying products.
- To manage information of different types of items.
- Consistently update information of all the item.
- Managing security by providing authorized email & password. Manages database efficiently

# Chapter 2: Objective, Scope & Limitations

## 2.1 Objective

The main objective of the Speak and Shop Portal is to manage the details of Products, Customer, Shipping, Payment, and Category with voice command. It manages all the information about Products, Sales, Category, and Products. The project is totally built at administrative end and thus only the administrator is guaranteed the access. The purpose of the project is to build an application program to reduce the manual work with the help of voice oriented system for managing the Products, Customer, Sales, and Shipping. It tracks all the details about the Shipping, Payment, and Category.

## 2.2 Scope of Project

This website is created so that it helps visually impaired and blind people to use the e-commerce websites with ease. This will also help other normal users to shop easily. In future with the new studies additional technologies can be integrated with the existing website which further prove to be helpful for the audience in better navigation. In the future the website can also be converted to a mobile application which can help users navigate with the help of gestures. With the increasing efforts by major companies like Google, Netflix, Uber, Apple, and Samsung to incorporate accessibility, in future the technologies used in the website can also act as an extension to other e-commerce websites so that it increases target audience on a global level.

## 2.3 Limitations

**Security:** One of the main limitations of e-commerce is security. In most cases, people are hesitant to provide their personal and financial details in spite of advanced data encryption security systems in place. Moreover, there are some websites that do not have the capability and features installed to authenticate transactions. As such, there are instances of fraudulent activities. The fear of providing financial information like credit card details hinders the growth of e-commerce.

**Lack of Privacy:** To some extent, the privacy of a customer is compromised in e-commerce. You need to provide your personal details, such as an address, telephone number, and so on to the seller. There are still lots of sites that do not have the advanced technology to protect sensitive information. Moreover, there are also sites that illegally collect consumer statistics without permission. This is one reason why people get skeptical while using e-commerce.

**Product Suitability:** As already mentioned, it is not possible for people to physically examine The product in e-commerce. In many cases, the original product may not match the picture or specifications in the e-commerce site. This absence of 'touch and feel' creates a discouraging effect.

**Technical Limitations:** e-commerce requires advanced technology platforms for better performance. Some limitations, such as lack of proper domain, network and software issues, and so on can affect the seamless performance of an e-commerce site.

**Delivery Guarantee:** Many people fear that their product might not be shipped or the website might be a fraud. Businesses need to work to build customer trust with reviews, testimonials, etc. to add more value to their website.

**Legal Issues:** A lot of legal compliances and cyber laws need to be taken care of in an e-commerce business. These regulations may vary from country to country. All these reasons deter businesses from going electronic.

## 2.4 Planning

**1.** Competitor analysis. In the early stages of the project, we conduct in-depth competitor and market research. It helps us to identify the weaknesses and strengths of the market leaders. Moreover, we receive insights on the mistakes we can avoid and the best practices we can adopt.

**2.** Feature list. This stage includes selecting the features we will integrate into the e-commerce website or the future online platform. The list of features has an impact on the user experience of your future customers.

**3.** Shipment and payment providers. Our team carefully selects the payment and shipment providers. The choice has a direct impact on the number of future sales, expenses, and the online shop's trustworthiness.

**4.** Design. Our team could recommend using some particular layouts or design elements distinguished as your competitor's best practices. The team also offers several options of themes from websites. In addition, we can develop a custom e-commerce design according to your business needs.

**5.** Data migration. In some cases, a customer already has an online shop with databases to import into the future project. We include the database migration to the workflow since it has an impact on the cost scope and the time of the online shop development.

**6.** The e-commerce platform. We offer several options of the e-commerce platforms depending on the customer business needs, goals, budget, and other factors. In some cases, we develop custom ecommerce platforms.

Then, to keep the project progress transparent and measurable, we use six main deliverables.

## 2.5 Feasibility

Whatever we think need not be feasible. It is wise to think about the feasibility of any problem we undertake. Feasibility is the study of impact, which happens in the organization by the development of a system. The impact can be either positive or negative. When the positives nominate the negatives, then the system is considered feasible. Here the feasibility study can be performed in two ways such as technical feasibility and Economical Feasibility.

**Technical Feasibility**

It is technically feasible, since there will not be much difficulty in getting required resources for the development and maintaining the system as well. All the resources needed for the development of the software as well as the maintenance.

**Economical Feasibility**

Development of this application is highly economically feasible .The organization needed not spend much m one for the development of the system already available. The only thing is to be done is making an environment for the development with an effective supervision. I f we are doing so , we can attain the maximum usability of the corresponding resources .Even after the development , the organization will not be in a condition to invest more in the organization .Therefore , the system is economically feasible.

## 2.6 Hardware & Software Requirements

### 2.6.1 Client Side

|  | DESCRIPTION | ALTERNATIVE (if any) |
|---|---|---|
| **HARDWARE** | PC with 20 GB hard-disk and minimum 512 MB RAM with Internet facility. | Not Applicable |
| **SOFTWARE** | Internet Explorer | Mozilla Firefox / Google Chrome / Apple Safari |

Table 2.1: Client Side Requirements

### 2.6.2 Server Side

|  | DESCRIPTION | ALTERNATIVE (if any) |
|---|---|---|
| **HARDWARE** | PC with 1 TB hard-disk and minimum 2 GB RAM with Internet facility. | AWS(or any other cloud platform) |
| **SOFTWARE** | Python & MySQL Server | Not Applicable |

Table 2.1 : Server Side Requirements

# Chapter 3: Planning and Feasibility Analysis

## 3.1 Planning

Planning for every project is very individual and specific. However, several major steps are going to be similar for many projects of the businesses. These major steps are described in the next paragraphs.

### STEP #1: CONSIDER ECOMMERCE WEBSITE SCOPE OF WORK

The first step to start with is to craft a scope statement and a general outline for the project. Creating the outline and the scope is usually the first step towards writing technical documentation and specs. Certainly, documentation is going to be changed in the scope of developing the project but initially, it is created by stakeholders and a project manager. The first step also contains developing the purpose and objectives of the project as well as the business justification for the project and is practically a basis for the entire future work.

### STEP #2: ORGANIZE DELIVERABLES INTO A WORK BREAKDOWN STRUCTURE

When it comes to implementing project ideas effectively and promptly, there is a need to prepare deliverables. In turn, making deliverables is a part of the WBS that is the work breakdown structure for the project. A manager is the one to create the WBS according to the requirements of the customer, which need to include project scope, project tasks, and tasks split into simpler subtasks. Such an approach helps to make one step at a time and build the project quicker. Thus, the WBS is necessary as a project roadmap.

### STEP #3: GET TO KNOW YOUR DEPENDENCIES

When you already have the parts of your project to do with the WBS, it is time to create a to-do list for every subtask. At this point, many managers often ignore dependencies and this might create bottlenecks in the project. To avoid bottlenecks, there is a need to plan tasks, taking in mind their dependencies on other tasks and subtasks. It is important to know whether there are subtasks that need to be completed before another task or at least initiated. With this knowledge, it is easier to create a smooth working process. It is a duty of a project manager to specify dependencies in the WBS and to make sure these dependencies are taken into account when the work is planned.

### STEP #4: CREATE A TIMELINE

Have you heard about a Gantt chart? It is a very convenient tool to use when it comes to planning the milestones and the dependencies. With a Gantt chart, you can plan time and tasks in a way that at any given moment you can see the status your project is in. The example of a Gantt chart is represented in the picture below. Usually, a Gantt chart is created in Excel but you can generally use whatever you find comfortable. The main idea is to be specific enough, to keep planning close to the needs of your project, and to make sure that all the dependencies are taken into account when planning those steps.

*STEP #5 PLAN BUDGET*

It goes without saying, before starting the project, there is a need to estimate the total cost that you can afford. Ideally, there should be several estimations, depending on the timeline and features added to the project. For example, you might want to have the lowest and the highest total cost of your project at hand so you can estimate the project from several points of view. Potentially, this helps to be more realistic with the budget, and determine the features that are a must for the project and the ones that are auxiliary but desired.

| Week | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| Planning | ■ | ■ |  |  |  |  |  |  |  |  |  |  |
| Design |  | ■ | ■ | ■ | ■ |  |  |  |  |  |  |  |
| Development |  |  |  | ■ | ■ | ■ | ■ | ■ |  |  |  |  |
| Testing |  |  |  |  |  |  |  | ■ | ■ |  |  |  |
| Deployment |  |  |  |  |  |  |  |  | ■ | ■ | ■ | ■ |

**Figure 3.1 Gantt Chart**

## 3.2 Feasibility

After doing the project speak and shop project, study and analyzing all the existing or required functionalities of the system, the next task is to do the feasibility study for the project. All projects are feasible - given unlimited resources and infinite time. Feasibility study includes consideration of all the possible ways to provide a solution to the given problem. The proposed solution should satisfy all the user requirements and should be flexible enough so that future changes can be easily done based on the future upcoming requirements. Preliminary investigation examine project feasibility, the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All system is feasible if they are unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:

• Technical Feasibility

• Operation Feasibility

• Economical Feasibility

*Technical Feasibility*

The technical issue usually raised during the feasibility stage of the investigation includes the following:

- Does the necessary technology exist to do what is suggested?
- Do the proposed equipment's have the technical capacity to hold the data required to use the system?

- Will the proposed system provide adequate response to inquiries, regardless of the number or location of users?
- Can the system be upgraded if developed?
- Are there technical guarantees of accuracy, reliability, ease of access and data security?

Earlier no system existed to cater to the needs of 'Secure Infrastructure Implementation System'. The current system developed is technically feasible. It is a web based user interface for audit workflow at NIC-CSD. Thus it provides an easy access to the users. The database's purpose is to create, establish and maintain a workflow among various entities in order to facilitate all concerned users in their various capacities or roles. Permission to the users would be granted based on the roles specified. Therefore, it provides the technical guarantee of accuracy, reliability and security. The software and hard requirements for the development of this project are not many and are already available in-house at NIC or are available as free as open source. The work for the project is done with the current equipment and existing software technology. Necessary bandwidth exists for providing a fast feedback to the users irrespective of the number of users using the system.

*Operational Feasibility*

Proposed projects are beneficial only if they can be turned out into information system. That will meet the organization's operating requirements. Operational feasibility aspects of the project are to be taken as an important part of the project implementation. Some of the important issues raised are to test the operational feasibility of a project includes the following: -

- Is there sufficient support for the management from the users?
- Will the system be used and work properly if it is being developed and implemented?
- Will there be any resistance from the user that will undermine the possible application benefits?

This system is targeted to be in accordance with the above-mentioned issues. Beforehand, the management issues and user requirements have been taken into consideration. So there is no question of resistance from the users that can undermine the possible application benefits. The well-planned design would ensure the optimal utilization of the computer resources and would help in the improvement of performance status.

*Economic Feasibility*

A system can be developed technically and that will be used if installed must still be a good investment for the organization. In the economical feasibility, the development cost in creating the system is evaluated against the ultimate benefit derived from the new systems. Financial benefits must equal or exceed the costs. The system is economically feasible. It does not require any addition hardware or software. Since the interface for this system is developed using the existing resources and technologies available at NIC, There is nominal expenditure and economical feasibility for certain.

# Chapter 4: Software Design

## 4.1 System Design

System Design of Speak and Shop System In this phase, a logical system is built which fulfils the given requirements. Design phase of software development deals with transforming the client's requirements into a logically working system. Normally, design is performed in the following in the following two steps:

1. ***Primary Design Phase:***
   In this phase, the system is designed at block level. The blocks are created on the basis of analysis done in the problem identification phase. Different blocks are created for different functions emphasis is put on minimizing the information flow between blocks. Thus, all activities which require more interaction are kept in one block.

2. ***Secondary Design Phase:***
   In the secondary phase the detailed design of every block is performed. The general tasks involved in the design process are the following:

   i. Design various blocks for overall system processes.
   ii. Design smaller, compact and workable modules in each block.
   iii. Design various database structures.
   iv. Specify details of programs to achieve desired functionality.
   v. Design the form of inputs, and outputs of the system.
   vi. Perform documentation of the design.
   vii. System reviews.

*User Interface Design*

User Interface Design is concerned with the dialogue between a user and the computer. It is concerned with everything from starting the system or logging into the system to the eventually presentation of desired inputs and outputs. The overall flow of screens and messages is called a dialogue The following steps are various guidelines for User Interface Design:

1. The system user should always be aware of what to do next.
2. The screen should be formatted so that various types of information, instructions and messages always appear in the same general display area.
3. Message, instructions or information should be displayed long enough to allow the system user to read them.
4. Use display attributes sparingly.
5. Default values for fields and answers to be entered by the user should be specified.
6. A user should not be allowed to proceed without correcting an error.
7. The system user should never get an operating system message or fatal error.

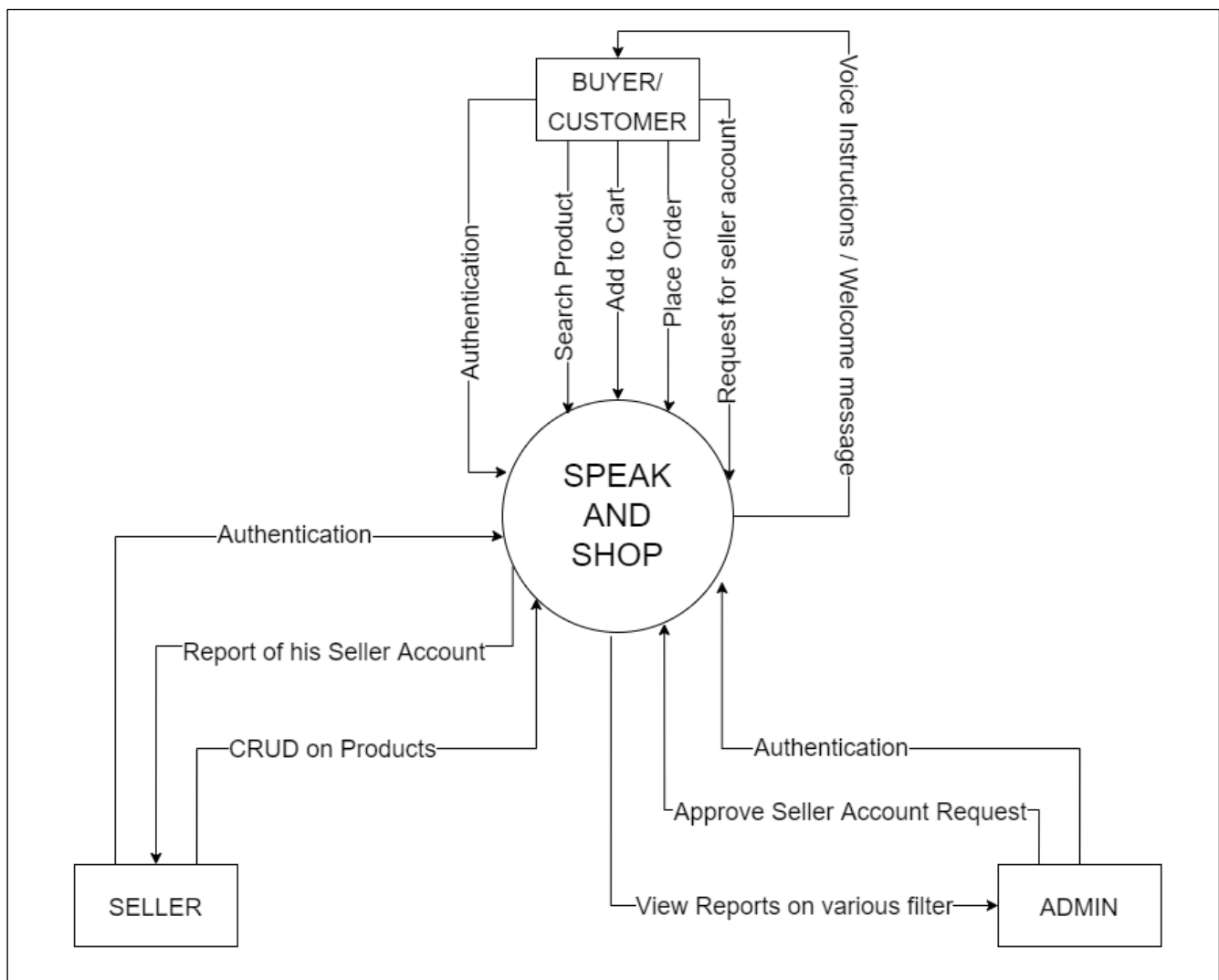## 4.2 Detailed Design

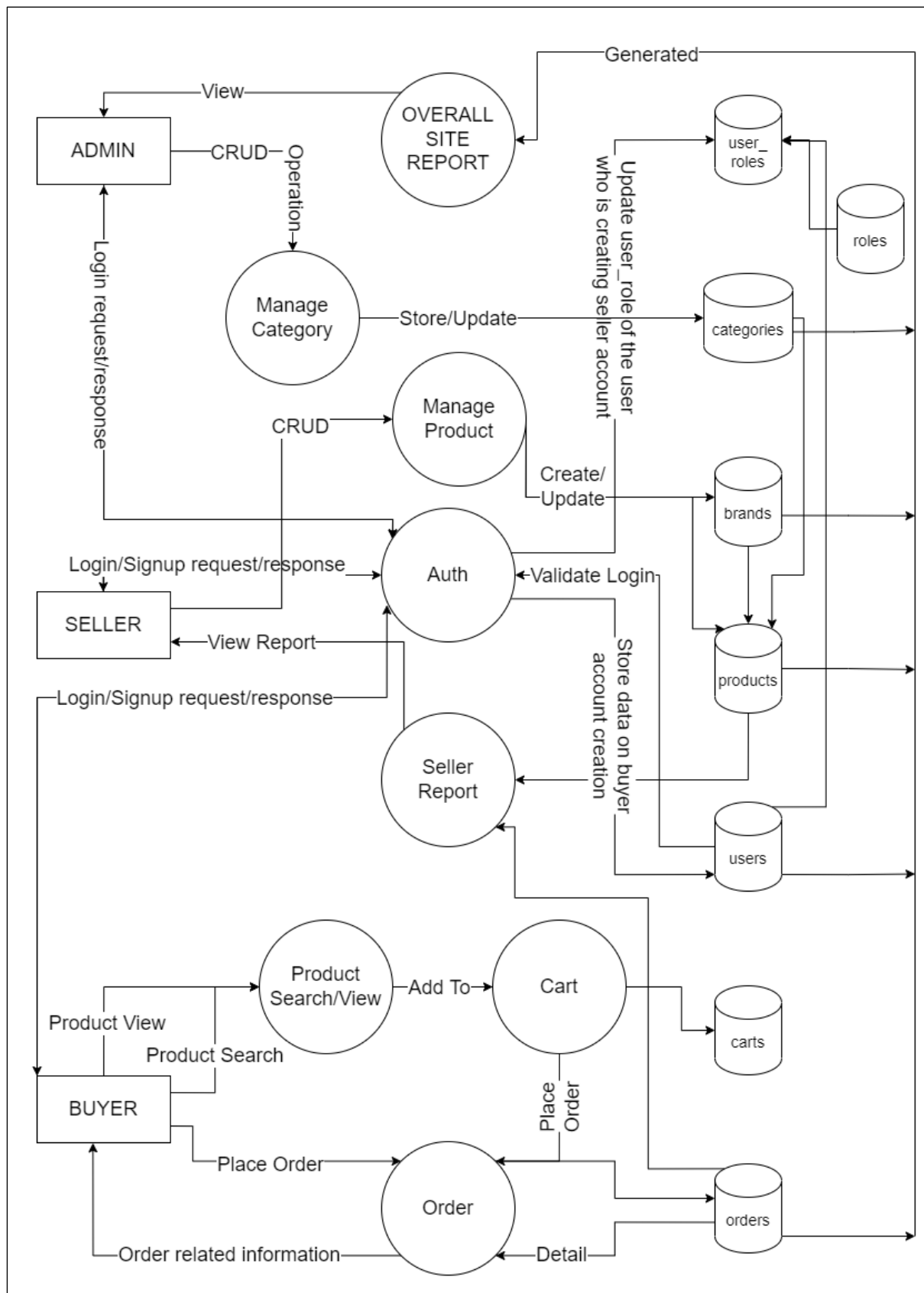### 4.2.1 0 level DFD



Figure 4.1: 0 Level DFD

## 4.2.2 1 level DFD

# Chapter 5: Database Design
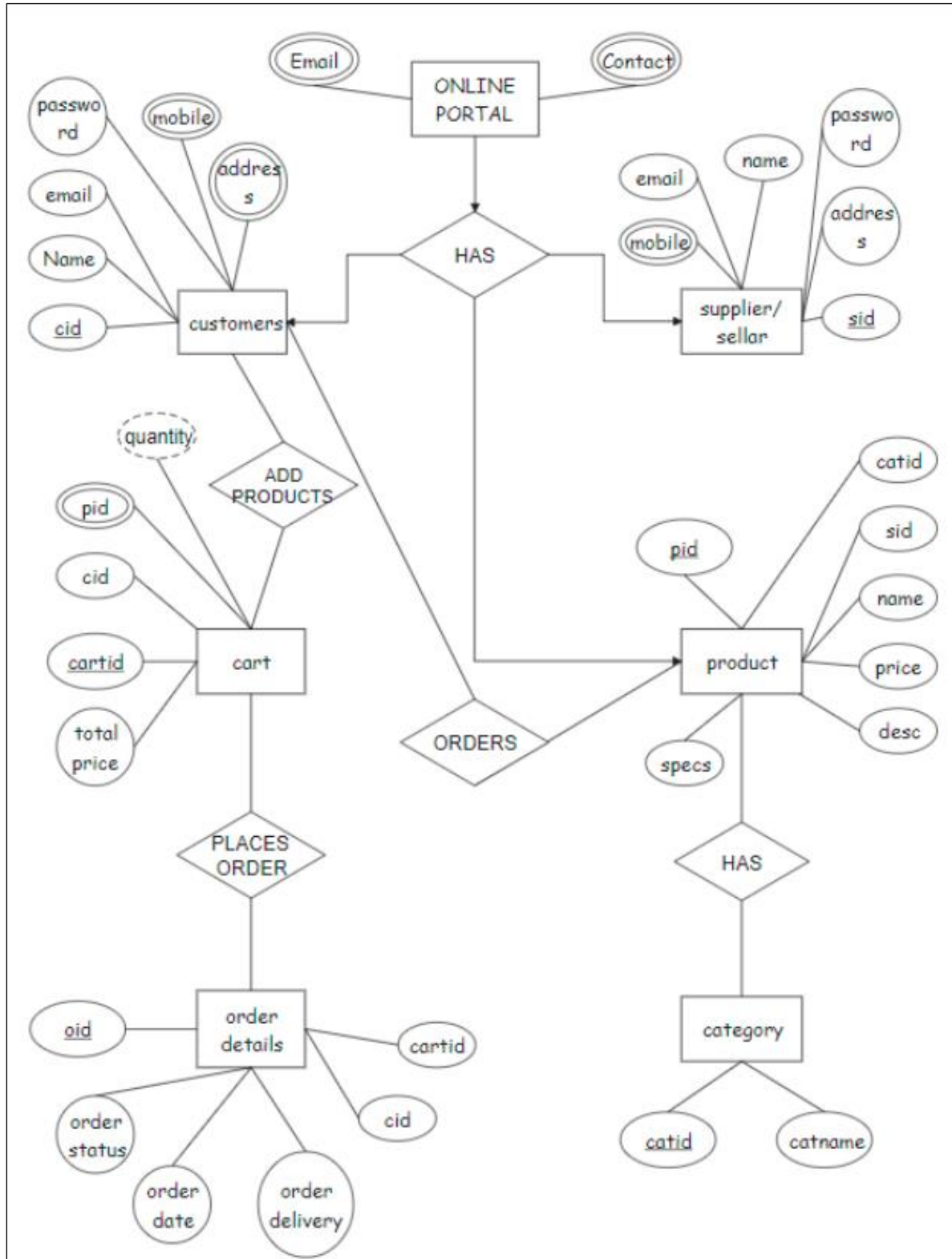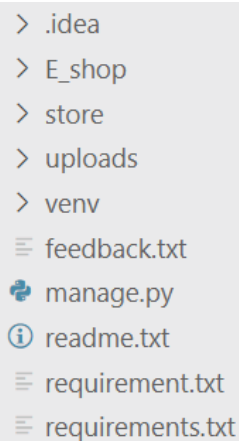
## 5.1 ER Diagram

## 5.2 Schema Diagram



**Figure 5.2: Schema Diagram**

# Chapter 6: Coding

After all the project planning and discussions the main part comes into act where our planning must be converted to desired output to our work through logically well-developed code. Coding doesn't means to just put your knowledge in coding and bring out the required task anyhow, but real coding means that it must follow some certain coding rules as: coding structure, maintaining proper folder/file structure, naming conviction that are used for variable, using comment as essential part in our code when working in a team environment so that our works may be easy to understand by other members as well as to any 3$^{rd}$ person who reviews our project, separating different coding files from each other for example all style files must be inside a separate folder with proper name as style or css also in same way any js or images that are being used in the front end side should be wrapped in a folder named assets.
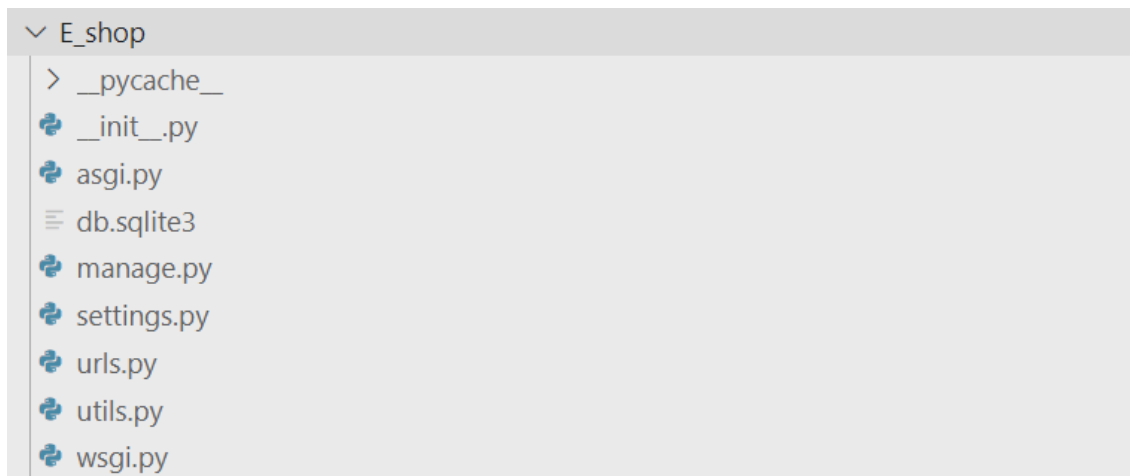
```
> .idea
> E_shop
> store
> uploads
> venv
≡ feedback.txt
🐍 manage.py
ⓘ readme.txt
≡ requirement.txt
≡ requirements.txt
```
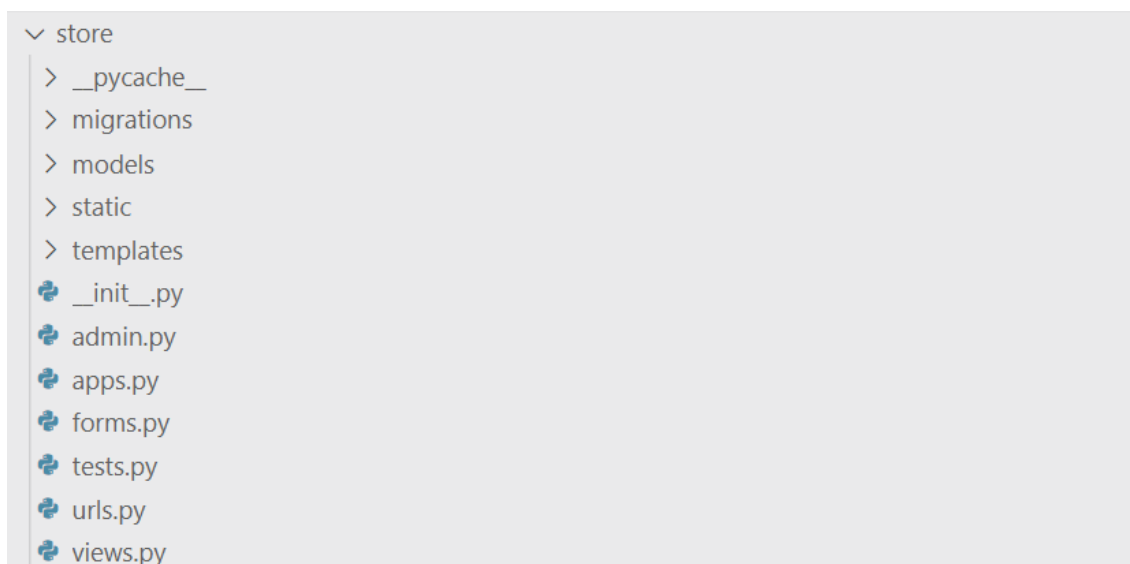
**Figure 6.1: Project Root Folder Structure**

Above is the folder structure of the root folder of the project. As we can see in this image we have multiple folders in this directory and some files too. Let's have a quick view on every directory's structure one by one.
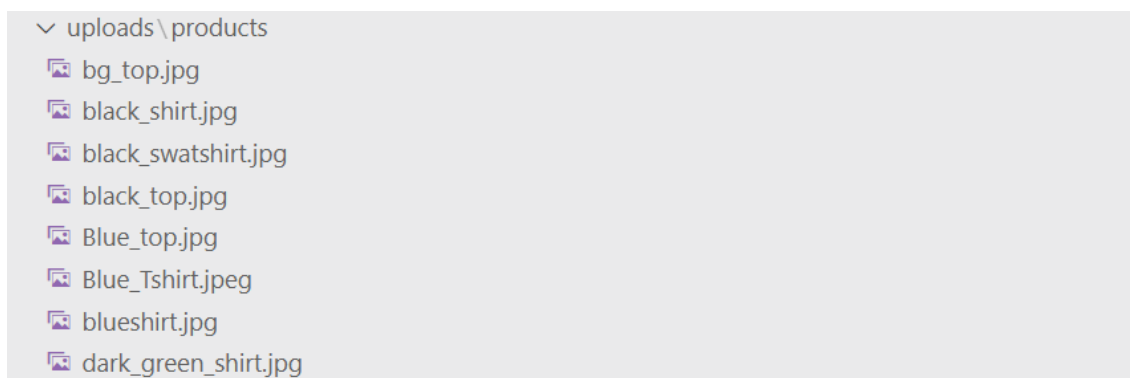
1. E_shop

```
∨ E_shop
  > __pycache__
  🐍 __init__.py
  🐍 asgi.py
  ≡ db.sqlite3
  🐍 manage.py
  🐍 settings.py
  🐍 urls.py
  🐍 utils.py
  🐍 wsgi.py
```

2. Store

```
∨ store
  > __pycache__
  > migrations
  > models
  > static
  > templates
  🐍 __init__.py
  🐍 admin.py
  🐍 apps.py
  🐍 forms.py
  🐍 tests.py
  🐍 urls.py
  🐍 views.py
```

3. Uploads

```
∨ uploads\products
  🖼 bg_top.jpg
  🖼 black_shirt.jpg
  🖼 black_swatshirt.jpg
  🖼 black_top.jpg
  🖼 Blue_top.jpg
  🖼 Blue_Tshirt.jpeg
  🖼 blueshirt.jpg
  🖼 dark_green_shirt.jpg
```

4. venv

```
∨ venv
  > Lib
  > Scripts
  ◈ .gitignore
  ⚙ pyvenv.cfg
```

That's all with the directory structure of the project . Now let's get started with the code.

**File: urls.py**

```python
from functools import reduce
from json import JSONEncoder
import os
from pickle import NONE
from types import NoneType
from django.shortcuts import render,redirect
from django.http import HttpResponse
from django.template import engines
from .models.product import Product
from .models.category import Category
from .models.order import *
from .models.seller import *
from django.http import JsonResponse
from django.db.models.functions import ExtractMonth
from django.db.models import Max,Min,Count,Avg
from django.template.loader import render_to_string
from django.views.decorators.csrf import csrf_exempt
from django.contrib.auth.models import User
from django.contrib import messages
from django.contrib.auth import authenticate, login, logout
from .forms import AddressBookForm, Productform, Sellerform, SignupForm,ProfileForm
from django.contrib.auth.decorators import login_required
import razorpay
from django.conf import settings
import datetime
from django.contrib.auth.hashers import make_password,check_password

now = datetime.datetime.now()

# Create your views here.
client = razorpay.Client(
    auth=(settings.RAZOR_KEY_ID, settings.RAZOR_KEY_SECRET))
```

```python
def index(request):
    products=NoneType
    categories=Category.get_all_category()
    categoryId=request.GET.get('category')
    latest=Product.get_latest_product()
    if categoryId:
        products=Product.get_all_product_by_categoryid(categoryId)
    else:
        products=Product.get_limited_product()
    # voice message
    # import pyttsx3
    # engine = pyttsx3.init()
    # speak = engine.say("hello");
    # engine.runAndWait()
    data={}
    data['products']=products
    data['latest']=latest
    data['categories']=categories
    # data['speak'] = speak
    return render(request,'index.html',data)


def shop(request):
    products=NoneType
    categories=Category.get_all_category()
    categoryId=request.GET.get('category')
    if categoryId:
        products=Product.get_all_product_by_categoryid(categoryId)
    else:
        products=Product.get_all_product()
    data={}
    data['products']=products
    data['categories']=categories
    return render(request,'shop.html',data)


def search(request):
    q=request.GET['q']
    data={}
    # data=Product.objects.filter(name__icontains='q')
    products=NoneType
    res="
    categories=Category.get_all_category()
    products=Product.search_product(q)
    if products.exists():
        data['products']=products
    else:
        res="Product not found !!"
    data['categories']=categories
    return render(request,'search.html',{'products':products,'categories':categories,'response':res})
```

```python
def product_detail(request,slugs):
    product=Product.objects.filter(slugs__iexact=slugs)
    categories=Category.get_all_category()
    data={}
    data['products']=product
    data['categories']=categories

    return render(request,'product_detail.html',data)


def filter_product(request):
    cat_id=request.GET['id']
    products=Product.get_all_product_by_categoryid_onfilter(cat_id)
    return JsonResponse({'prod_data':list(products)})


def add_to_cart(request):
    # del request.session['cartdata']
    cart_p={}
    cart_p[str(request.GET['id'])]={
        'image':request.GET['image'],
        'title':request.GET['title'],
        'qty':int(request.GET['qty']),
        'price':float(request.GET['price']),
    }
    if 'cartdata' in request.session:
        if str(request.GET['id']) in request.session['cartdata']:
            cart_data=request.session['cartdata']
            cart_data[str(request.GET['id'])]['qty']=int(cart_p[str(request.GET['id'])]['qty'])
            cart_data.update(cart_data)
            request.session['cartdata']=cart_data
        else:
            cart_data=request.session['cartdata']
            cart_data.update(cart_p)
            request.session['cartdata']=cart_data
    else:
        request.session['cartdata']=cart_p

    return JsonResponse({'data':request.session['cartdata'],'totalitems':len(request.session['cartdata'])})


def cart_list(request):
    total_amt=0
    c=0
    if 'cartdata' in request.session:
        for p_id,item in request.session['cartdata'].items():
            total_amt+=int(item['qty'])*float(item['price'])
            c=len(request.session['cartdata'])
        print(c)
```

```python
        return render(request,
'cart.html',{'cart_data':request.session['cartdata'],'totalitems':len(request.session['cartdata']),'total_amt':
total_amt,'c':range(1,c+1)})
    else:
        messages.error(request,"Empty Cart")
        return render(request, 'cart.html',{'cart_data':'','totalitems':0,'total_amt':total_amt})


def delete_cart_item(request):
    p_id=str(request.GET['id'])
    if 'cartdata' in request.session:
        if p_id in request.session['cartdata']:
            cart_data=request.session['cartdata']
            del request.session['cartdata'][p_id]
            request.session['cartdata']=cart_data
    total_amt=0
    for p_id,item in request.session['cartdata'].items():
        total_amt+=int(item['qty'])*float(item['price'])
    t=render_to_string('cart.html',{'cart_data':request.session['cartdata'],'totalitems':len(request.session['
cartdata']),'total_amt':total_amt})
    return JsonResponse({'data':t,'totalitems':len(request.session['cartdata'])})


def sample(request):
    text = request.POST.get('your_name',False);
    print(text)
    return render(request,'sample.html')


import pyttsx3
@csrf_exempt
def welcome_message(request):
    if request.method == "POST":
        engine = pyttsx3.init()
        engine.setProperty('rate', 125)
        voices = engine.getProperty('voices')
        # engine.setProperty('voice', voices[1].id)
        engine.say('Hello sir/Mam, Welcome to our speak and shop website.')
        # engine.say('Some instructions to use this are ')
        # engine.say('click on start speaking to recognize voice')
        # engine.say('Then use command for particular work, like use find command for search any
product')
        # engine.say('show for product details')
        # engine.say('go to cart, add to cart are some commands to use our website')
        engine.runAndWait()
        return HttpResponse()


@csrf_exempt
def testing(request):
    if request.method == "POST":
        name = request.POST.get('start')
```

21

```python
        # name.replace(" ","")
        prod=Product.get_all_product_list()
        # print(prod)
        for n in prod:
            print(n['name'])
            if n['name'].lower() == name:
                temp = "matched"
                return HttpResponse(n['id'])
            else:
                temp = "not matched"
        if temp == "not matched" :
            return HttpResponse("not matched"+name)


def signup(request):
    if request.method=='POST':
        form=SignupForm(request.POST)
        username=request.POST.get('username')
        mod=User.objects.get(username=username)
        if mod:
            messages.success(request,"User Already exists")
            return redirect('signup')
        if form.is_valid():
            username=form.cleaned_data.get('username')
            pwd=form.cleaned_data.get('password1')
            form.save()
            user=authenticate(username=username,password=pwd)
            login(request, user)
            if user is not None:
                #login(request, user)

                messages.success(request, "Registered Successfully!!")
                messages.success(request, "Logged in Successfully!!")
                return redirect('my_dashboard')
        else:
            messages.error(request,"Invalid Form")
    form=SignupForm()
    return render(request, 'registration/signup.html',{'form':form})


def auth_login(request):
    if request.user.is_authenticated:
        messages.success(request,"Already Logged In")
        return redirect('my_dashboard')
    else:
        if request.method=='POST':
            name=request.POST.get('username')
            pwd=request.POST.get('password')
            user=authenticate(username=name,password=pwd)
            if user is not None:
```

```python
            login(request, user)
            messages.success(request,"Logged in Successfully!!")

            if request.path == '?next=/checkout/':
                return redirect('checkout')
            else:
                return redirect('my_dashboard')
        else:
            messages.error(request,"Invalid Username or Password")
            return redirect('login')
    return render(request,'registration/login.html')


def signout(request):
    logout(request)
    messages.success(request, "Logged Out Successfully!!")
    return redirect('login')


@login_required
def checkout(request):
    total_amt=0
    totalAmt=0
    if 'cartdata' in request.session:
        for p_id,item in request.session['cartdata'].items():
            totalAmt+=int(item['qty'])*float(item['price'])
        total_amt=totalAmt*100
        payment_order = client.order.create(dict(amount= total_amt, currency= 'INR', payment_capture=
1 ))
        payment_order_id=payment_order['id']
        # Order
        d=User.objects.get(username=request.user)
        # Process Payment
        address=UserAddressBook.objects.filter(user=request.user,status=True).first()
        print(request.method)
        context={}
        context['cart_data']=request.session['cartdata']
        context['totalitems']=len(request.session['cartdata'])
        context['total_amt']=total_amt//100
        context['email']=d.email
        context['name']=d.first_name
        context['amount']=total_amt*100
        context['api_key']=settings.RAZOR_KEY_ID
        context['order_id']=payment_order_id
        context['callback_url']='/paymenthandler/'
        if  address:
            context['address']=address
            request.session['addr']=address.address
            request.session['mob']=address.mobile
```

```
    #return render(request,
'checkout.html',{'cart_data':request.session['cartdata'],'totalitems':len(request.session['cartdata']),'tot
al_amt':total_amt,'form':form,'amount':total_amt,'api_key':'rzp_test_E0xoOAqkc15oqs','order_id':pa
yment_order_id,'callback_url':'paymenthandler/'})
        return render(request, 'checkout.html',context)
    else:
        messages.success(request,"Add/Select  address before checkout !!")
        return redirect('my-addressbook')


@csrf_exempt
def paymenthandler(request):
    # only accept POST request.
    if request.method == "POST":
        total_amt=0
        totalAmt=0
        addr=request.session['addr']
        mob=request.session['mob']
        if 'cartdata' in request.session:
            for p_id,item in request.session['cartdata'].items():
                totalAmt+=int(item['qty'])*float(item['price'])
                get_sid=Product.objects.get(name=item['title'])
                si=get_sid.sid
                print(get_sid.sid)
            order=CartOrder.objects.create(
                user=request.user,
                order_id=request.POST.get('razorpay_order_id'),
                total_amt=totalAmt,
                paid_status=True,
                sid=si
            )
        # End
            del request.session['addr']
            del request.session['mob']
            #Orderitem
            for p_id,item in request.session['cartdata'].items():
                total_amt+=int(item['qty'])*float(item['price'])
                # OrderItems
                items=CartOrderItems.objects.create(
                    order=order,
                    invoice_no='INV-'+str(order.id),
                    item=item['title'],
                    image=item['image'],
                    address=addr,
                    mobile=mob,
                    qty=item['qty'],
                    price=item['price'],
                    total=float(item['qty'])*float(item['price'])
                    )
```

```python
        # End
    try:
        # get the required parameters from post request.
        payment_id = request.POST.get('razorpay_payment_id')
        razorpay_order_id = request.POST.get('razorpay_order_id')
        signature = request.POST.get('razorpay_signature')
        params_dict = {
            'razorpay_order_id': razorpay_order_id,
            'razorpay_payment_id': payment_id,
            'razorpay_signature': signature
        }
        # verify the payment signature.
        result = client.utility.verify_payment_signature(
            params_dict)
        if result is not None:
            amount = 85000  # Rs. 200
            try:
                # capture the payemt
                client.payment.capture(payment_id, amount)
                # render success page on successful caputre of payment
                del request.session['cartdata']
                messages.success(request,"Payment Done")
                return redirect('home')
            except:
                # if there is an error while capturing payment.
                del request.session['cart_data']
                messages.success(request,"Payment Done")
                return redirect('home')
        else:
            # if signature verification fails.
            return render(request, 'payment-fail.html')
    except:
        # if we don't find the required parameters in POST data
        del request.session['cartdata']
        messages.success(request,"Payment DOne")
        return redirect('home')
else:
    # if other than POST request is made.
    return HttpResponse('error2')


@csrf_exempt
def payment_done(request):
    returnData=request.POST
    return render(request, 'payment-success.html')


@csrf_exempt
def payment_error(request):
    if request.method=='POST':
```

```python
    data={
        'error_code':request.POST.get('error_code'),
        'error_desc':request.POST.get('error_desc'),
        'error_source':request.POST.get('error_source'),
        'error_step':request.POST.get('error_step'),
        'error_reason':request.POST.get('error_reason'),
        'error_orderid':request.POST.get('error_orderid'),
        'error_pid':request.POST.get('error_pid'),
    }
    return render(request, 'payment-fail.html',data)


import calendar
@login_required
def my_dashboard(request):
    orders=CartOrder.objects.filter(user=request.user).annotate(month=ExtractMonth('order_dt')).values('month').annotate(count=Count('id')).values('month','count')
    monthNumber=[]
    totalOrders=[]
    for d in orders:
        monthNumber.append(calendar.month_name[d['month']])
        totalOrders.append(d['count'])
    return render(request,
'user/dashboard.html',{'monthNumber':monthNumber,'totalOrders':totalOrders})


@login_required
def my_orders(request):
    orders=CartOrder.objects.filter(user=request.user).order_by('-id')
    return render(request, 'user/orders.html',{'orders':orders})


@login_required
def my_order_items(request,id):
    order=CartOrder.objects.get(pk=id)
    orderitems=CartOrderItems.objects.filter(order=order).order_by('-id')
    return render(request, 'user/order-items.html',{'orderitems':orderitems})


@login_required
def my_addressbook(request):
    addbook=UserAddressBook.objects.filter(user=request.user).order_by('-id')
    return render(request, 'user/addressbook.html',{'addbook':addbook})


def save_address(request):
    msg=None
    if request.method=='POST':
        form=AddressBookForm(request.POST)
        if form.is_valid():
            saveForm=form.save(commit=False)
            saveForm.user=request.user
            if 'status' in request.POST:
```

```python
                UserAddressBook.objects.update(status=False)
            saveForm.save()
            messages.success(request,"Address Added Successfully")
        form=AddressBookForm
        return render(request, 'user/add-address.html',{'form':form})


def activate_address(request):
    a_id=str(request.GET['id'])
    UserAddressBook.objects.update(status=False)
    UserAddressBook.objects.filter(id=a_id).update(status=True)
    return JsonResponse({'bool':True})


def update_address(request,id):
    address=UserAddressBook.objects.get(pk=id)
    msg=None
    if request.method=='POST':
        form=AddressBookForm(request.POST,instance=address)
        if form.is_valid():
            saveForm=form.save(commit=False)
            saveForm.user=request.user
            if 'status' in request.POST:
                UserAddressBook.objects.update(status=False)
            saveForm.save()

            messages.success(request,"Address Updated Successfully")
        form=AddressBookForm(instance=address)
        return render(request, 'user/update-address.html',{'form':form})


@login_required
def edit_profile(request):
    msg=None
    if request.method=='POST':
        form=ProfileForm(request.POST,instance=request.user)
        if form.is_valid():
            form.save()
            messages.success(request,"Profile Updated Successfully")
        form=ProfileForm(instance=request.user)
        return render(request, 'user/edit-profile.html',{'form':form})


@login_required
def pass_changed(request):
    messages.success(request,"Password Updated Successfully")
    return redirect('my_dashboard')


def pass_re_changed(request):
    messages.success(request,"Password Reset Successfully")
    return redirect('login')
```

```python
def check_pass(a,b):
    if(a==b):
        return True
    else:
        return False


def seller_signup(request):
    if request.session.get('seller'):
        return render(request, 'seller/dashboard.html')
    if request.method=='POST':
        form=Sellerform(request.POST)
        if form.is_valid():
            username=form.cleaned_data.get('username')
            pwd=form.cleaned_data.get('password1')
            raw_pass = make_password(form.cleaned_data.get('password1'))
            if Seller.isExists(username):
                messages.success(request,"User Already exists")
                return redirect('seller_signup')
            else:
                form.save()
                Seller.objects.filter(username=username).update(password=pwd)
                seller = Seller.get_seller_by_username(username)
                if seller:
                    flag = check_pass(pwd, seller.password)
                    if flag:
                        request.session['seller'] = seller.id
                        messages.success(request, "Registered Successfully!!")
                        messages.success(request, "Logged in Successfully!!")
                        return redirect('seller_dashboard')
        else:
            messages.error(request,"Invalid Form")
    form=Sellerform()
    return render(request, 'seller/register.html',{'form':form})


def seller_login(request):
    if request.session.get('seller'):
        return render(request, 'seller/dashboard.html')
    if request.method=='POST':
        username=request.POST.get('username')
        pwd=request.POST.get('password')
        rp=make_password(pwd)
        print(rp)
        print(check_password(pwd,rp))
        seller = Seller.get_seller_by_username(username)
        print(seller)
        print(pwd)
        print(make_password(pwd))
        print(seller.password)
```

```python
            psd=seller.password
            print(check_password('byEEeAbz4U@WwS8','pbkdf2_sha256$260000$EpXKR9PmsR7MtN
KznJunaG$GCq7Yy'))
        if seller:
            flag = check_pass(pwd,psd)
            print(flag)
            if flag:
                request.session['seller'] = seller.id
                messages.success(request, "Logged in Successfully!!")
                return redirect('seller_dashboard')
            else:
                messages.error(request,"Invaild Username or Password")
    form=Sellerform()
    return render(request, 'seller/login.html',{'form':form})


def seller_logout(request):
    request.session.clear()
    return redirect('seller_login')


def seller_dashboard(request):
    if request.session.get('seller'):
        seller_detail=Seller.objects.filter(id=request.session['seller']).all()
        pd=Product.objects.filter(sid=request.session['seller']).count()
        ct=CartOrder.objects.filter(sid=request.session['seller']).count()
        print(pd)
        return render(request, 'seller/dashboard.html',{'data':seller_detail,'pd':pd,'ct':ct})

    else:
        return redirect('seller_login')


def seller_profile(request):
    if request.session.get('seller'):
        data=Seller.objects.filter(id=request.session['seller']).all()
        form=Sellerform()
        return render(request,'seller/profile.html',{'data':data})
    else:
        return redirect('seller_login')


def seller_profile_edit(request):
    if request.session.get('seller'):
        data=Seller.objects.filter(id=request.session['seller']).all()
        msg=None
        if request.method=='POST':
            get_sid=Seller.objects.get(id=request.session['seller'])
            username=request.POST.get('username')
            pwd=request.POST.get('password1')
            if Seller.isExists(username):
                msg="User Already exists"
```

```
        else:
            get_sid.first_name=request.POST.get('first_name')
            get_sid.last_name=request.POST.get('last_name')
            get_sid.email=request.POST.get('email')
            get_sid.username=request.POST.get('username')
            get_sid.password=request.POST.get('password1')
            get_sid.save()
            Seller.objects.filter(username=username).update(password=pwd)
            msg="Your Profile Updated Successfully"
        return render(request,'seller/edit_profile.html',{'data':data,'msg':msg})
    else:
        return redirect('seller_login')


def seller_product(request):
    msg="
    if request.method == 'POST':
        form = Productform(request.POST, request.FILES)
        seller_id=request.session['seller']
        if form.is_valid():
            form.save()
            Product.objects.filter(name=request.POST['name']).update(sid=seller_id)
            msg="Product Added !!"
            return render('seller_product')
        else:
            msg="invalid !!"
    form=Productform()
    cat=Category.get_all_category()
    return render(request, 'seller/product.html',{'form':form,'cat':cat,'msg':msg})


def seller_edit_product(request,pid):
    cat=Category.get_all_category()
    get_sid=Product.objects.get(id=pid)
    get_s=get_sid.sid
    msg="
    print(get_s)
    if(request.session['seller']==get_s):
        prod = Product.objects.get(id=pid)
        msg=""
        if request.method == "POST":
            if len(request.FILES) != 0:
                if len(prod.image) > 0:
                    os.remove(get_sid.image.path)
                prod.image = request.FILES['image']
            prod.name = request.POST.get('name')
            prod.description = request.POST.get('description')
            prod.category = Category.objects.get(name=request.POST.get('category'))
            prod.price = request.POST.get('price')
            prod.save()
```

```python
                messages.success(request, "Product Updated Successfully")
                return redirect('seller_product_list')
        else:
            msg="You are not authoroised !!"
        return render(request,'seller/product_edit.html',{'msg':msg,'data':get_sid,'cat':cat})


@csrf_exempt
def delete_product(request):
    if request.method=="POST":
        pid=request.POST.get('pid')
        prod = Product.objects.get(id=pid)
        if len(prod.image) > 0:
            os.remove(prod.image.path)
        prod.delete()
        return JsonResponse({'status':1})


def list_product(request):
    products=NoneType
    products=Product.get_seller_product(request.session['seller'])
    return render(request,'seller/product_list.html',{'products':products})


def seller_order(request):
    orders=NoneType
    msg="You have checked"

    orders=CartOrder.get_seller_order(request.session['seller'])
    return render(request,'seller/order.html',{'orders':orders,'msg':msg})


def seller_order_status(request,oid):
    data=CartOrder.objects.get(order_id=oid)
    if request.method=='POST':
        data.order_status=request.POST.get('status')

        data.save()
        return redirect('seller_order')
    return render(request,'seller/change_status.html',{'id':oid,'status':data.order_status})


def save_product(request):
    if request.method=="POST":
        form=Productform(request.POST)
        if form.is_valid():
            pid=request.POST.get('pid')
            name=request.POST['name']
            price=request.POST['price']
            category=request.POST['category']
            description=request.POST['description']
            image=request.POST['image']
            if(pid==''):
```

```
            pd=Product(name = name,price = price,category = category,description=description,
image=image)
        else:
            pd=Product(id=pid,name = name,price = price,category = category,description=description,
image=image)
        pd.save()
        # std=User.objects.values()
        # student_data=list(std)

        return JsonResponse({'status':1})
    else:
        return JsonResponse({'status':0})
```

**File: urls.py**

```
from django.contrib import admin
from django.urls import path,include
from  .views import *
from django.contrib.auth import views as auth_views

urlpatterns = [
    path(",index,name='home'),
    path('search',search),
    path('shop/',shop),
    path('product/<str:slugs>',product_detail),
    path('filter_product/',filter_product,name='filter_product'),

    path('seller/sample/',sample),

    path('seller/signup/',seller_signup,name='seller_signup'),
    path('seller/login/',seller_login,name='seller_login'),
    path('seller/logout/',seller_logout,name='seller_logout'),
    path('seller/dashboard/',seller_dashboard, name='seller_dashboard'),
    path('seller/profile/',seller_profile, name='seller_profile'),
    path('seller/profile/edit/',seller_profile_edit, name='seller_profile_edit'),
    path('seller/product/',seller_product, name='seller_product'),
    path('seller/orders/',seller_order, name='seller_order'),
    path('seller/product/save',save_product, name='save'),
    path('seller/product/edit/<int:pid>',seller_edit_product, name='seller_edit_product'),
    path('seller/product/delete/',delete_product, name='delete_product'),
    path('seller/product/list/',list_product, name='seller_product_list'),
    path('seller/order/<str:oid>',seller_order_status, name='seller_order_status'),

    path('accounts/signup/',signup,name='signup'),
    path('accounts/login/',auth_login,name='login'),
    path('logout', signout, name='signout'),

path('password_reset/',auth_views.PasswordResetView.as_view(template_name='password_reset.html'),name='reset_passwo
rd'),
    path('reset_password_sent/', auth_views.PasswordResetDoneView.as_view(template_name="password_reset_sent.html"),
name="password_reset_done"),

path('reset/<uidb64>/<token>/',auth_views.PasswordResetConfirmView.as_view(template_name="password_reset_form.ht
ml",success_url='/password_reset_done'), name="password_reset_confirm"),
    path('reset_password_complete/',
auth_views.PasswordResetCompleteView.as_view(template_name="password_reset_done.html"),
name="password_reset_complete"),
```

```
    path('cart/',cart_list,name='cart'),
    path('add-to-cart/',add_to_cart,name='add_to_cart'),
    path('delete-from-cart/',delete_cart_item,name='delete_cart_item'),
    path('checkout/',checkout,name='checkout'),

    path('wlcm_msg', welcome_message),
    path('test',testing),
    path('shop/test',testing),
    path('cart/test',testing),

    #path('payment-done/', payment_done,name='payment_done'),
    path('paymenthandler/',paymenthandler,name='paymenthandler'),
    path('payment-error/',payment_error,name='payment_error'),

    path('my-dashboard',my_dashboard, name='my_dashboard'),
    path('my-orders',my_orders, name='my_orders'),
    path('my-orders-items/<int:id>',my_order_items, name='my_order_items'),
    path('edit-profile',edit_profile, name='edit-profile'),

path('password/',auth_views.PasswordChangeView.as_view(template_name='user/password_change_form.html',success_url
='password_change_done'),name='password_change'),
    path('password_change_done',pass_changed, name='pass_changed'),
    path('password_reset_done',pass_re_changed, name='pass_re_changed'),

    path('addressbook',my_addressbook, name='my-addressbook'),
    path('add-address',save_address, name='add-address'),
    path('activate-address',activate_address, name='activate-address'),
    path('update-address/<int:id>',update_address, name='update-address'),
]
```

# Chapter 7: Snapshots & Code of Modules

# *USER PART*
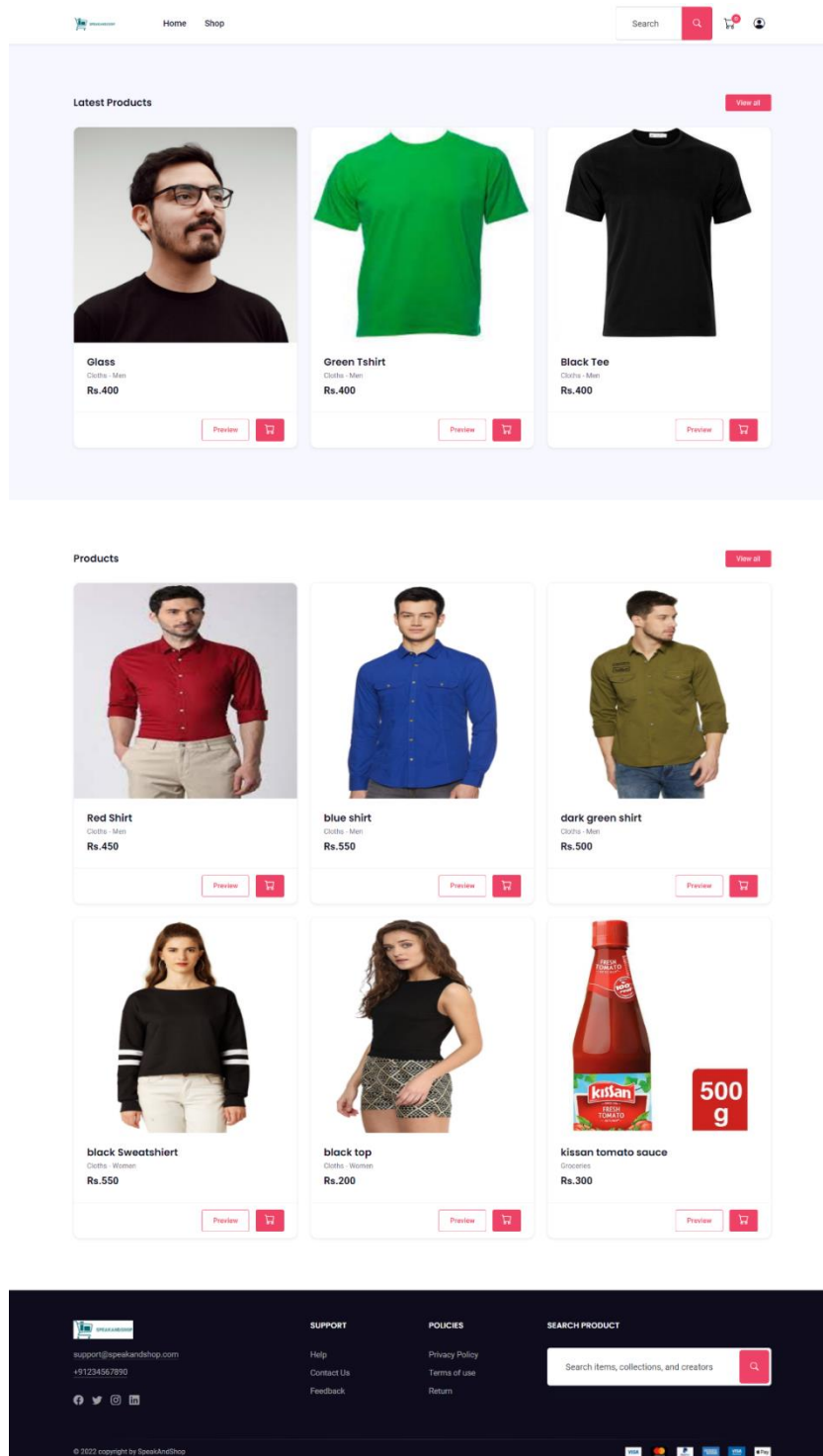
## Base.html

```
{% load static %}
<!DOCTYPE html>
<html lang="zxx">
<head>
    <!-- Title -->
    <title>{% block title %}
      {% endblock %}</title>
    <!-- Required Meta Tags Always Come First -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <meta http-equiv="x-ua-compatible" content="ie=edge">
    <style>
      #voice_start_btn {
        background-color : #31B0D5;
        color: white;
        padding: 10px 20px;
        border-radius: 4px;
        border-color: #46b8da;
      }
      #voice_stop_btn {
        background-color : #31B0D5;
        color: white;
        padding: 10px 20px;
        border-radius: 4px;
        border-color: #46b8da;
      }
      #mybutton {
        position: fixed;
        bottom: 100px;
        right: 50px;
        }
    </style>

    <!-- Favicon -->
    <link rel="shortcut icon" href="#">
    <!-- CSS Template -->
    <link href="{%static 'css/style.css'%}" rel="stylesheet">
    <!-- CSS Custom -->
    <link href="{%static 'css/custom.css'%}" rel="stylesheet">
    <!-- alertify CSS -->
    <link rel="stylesheet" href="//cdn.jsdelivr.net/npm/alertifyjs@1.13.1/build/css/alertify.min.css"/>
    <!-- alertify Default theme -->
    <link rel="stylesheet"
href="//cdn.jsdelivr.net/npm/alertifyjs@1.13.1/build/css/themes/default.min.css"/>
    <!-- jquery -->
    <script src="{%static '/js/jquery-3.5.1.min.js'%}"></script>
```

```
    <!--cart ajax js-->
</head>
<body>
  <!-- Preload -->
  <div id="loading" class="preloader">
    <div class="spinner-border text-primary" role="status"></div>
  </div>
  <!-- End Preload -->
  <script src="//cdn.jsdelivr.net/npm/alertifyjs@1.13.1/build/alertify.min.js"></script>
  <script>
    alertify.set('notifier','position', 'top-right');
    {% for msg in messages %}
      alertify.success('{{msg}}');
    {% endfor %}
  </script>
  <!-- Mini Cart -->
  {% comment %} <div class="modal px-modal-right fade" id="modalMiniCart" tabindex="-1"
role="dialog" aria-hidden="true">
    <!-- Shopping Cart -->
    <div class="modal-dialog px-modal-vertical">
      <div class="modal-content">
        <!-- Header-->
        <div class="modal-header border-bottom">
          <h6 class="m-0 fw-bold">
            Your Cart (2)
          </h6>
          <!-- Close -->
          <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close">
          </button>
        </div>
        <div class="modal-body">
          <!-- List group -->
          <ul class="list-unstyled m-0 p-0">
            <li class="py-3 border-bottom">
              <div class="row align-items-center">
                <div class="col-4">
                  <!-- Image -->
                  <a href="#">
                    <img class="img-fluid border" src="{%static '/img/product-2.png'%}"
alt="...">
                  </a>
                </div>
                <div class="col-8">
                  <!-- Title -->
                  <p class="mb-2">
                    <a class="text-dark fw-500" href="#">Cotton floral print Dress</a>
                    <span class="m-0 text-muted w-100 d-block">$40.00</span>
                    <a class="small link-danger ms-auto" href="#!">
```

```
                    <i class="bi bi-x"></i> Remove
                  </a>
                </p>
                <!--Footer -->
                <div class="d-flex align-items-center">
                </div>
              </div>
            </div>
          </li>
          <li class="py-3 border-bottom">
            <div class="row align-items-center">
              <div class="col-4">
                <!-- Image -->
                <a href="#">
                  <img class="img-fluid border" src="{%static '/img/product-2.png'%}"
alt="...">
                </a>
              </div>
              <div class="col-8">
                <!-- Title -->
                <p class="mb-2">
                  <a class="text-dark fw-500" href="#">Cotton floral print Dress</a>
                  <span class="m-0 text-muted w-100 d-block">$40.00</span>
                  <a class="small link-danger ms-auto" href="#!">
                    <i class="bi bi-x"></i> Remove
                  </a>
                </p>
                <!--Footer -->
                <div class="d-flex align-items-center">
                </div>
              </div>
            </div>
          </li>
        </ul>
      </div>
      <!-- Footer -->
      <div class="mt-auto p-3 pt-0">
        <div class="row g-0 py-2">
          <div class="col-8">
            <span class="text-dark">Subtotal</span>
          </div>
          <div class="col-4 text-end">
            <span class="ml-auto">$89.00</span>
          </div>
        </div>
        <div class="row g-0 py-2">
          <div class="col-8">
            <span class="text-dark">Taxes:</span>
```

```html
            </div>
            <div class="col-4 text-end">
              <span class="ml-auto">$89.00</span>
            </div>
          </div>
          <div class="row g-0 pt-2 mt-2 border-top fw-bold text-dark">
            <div class="col-8">
              <span class="text-dark">Subtotal</span>
            </div>
            <div class="col-4 text-end">
              <span class="ml-auto">$89.00</span>
            </div>
          </div>
          <div class="pt-4">
            <a class="btn btn-block btn-dark w-100 mb-3" href="checkout.html">Continue to
Checkout</a>
            <a class="btn btn-block btn-outline-dark w-100" href="cart.html">View Cart</a>
          </div>
        </div>
        <!-- Buttons -->
      </div>
    </div>
  </div> {% endcomment %}
  <!-- End Mini Cart -->
  <!-- Header -->
  <header class="header-main header-light shadow-sm position-relative z-index-4">
    <nav class="navbar navbar-expand-lg navbar-light">
      <div class="container">
        <!-- Logo -->
        <a class="navbar-brand" href="/">
          <img src="{%static '/img/logooo.png'%}" title="" alt="" >
        </a>
        <!-- Logo -->
        <!-- Menu -->
        <div class="collapse navbar-collapse" id="navbarSupportedContent">
          <ul class="navbar-nav me-auto ps-lg-8">
            <li class="nav-item">
              <a href="/" id="home-btn" class="nav-link">Home</a>
            </li>
            <li class=" nav-item">
              <a href="{{BASE_DIR}}/shop" id="shop-btn" class="nav-link">Shop</a>
            </li>
          </ul>
        </div>
        <!-- End Menu -->
        <div class="nav flex-nowrap align-items-center header-right">
          <!-- Nav Search-->
          <div class="nav-item">
```

```html
                <form class="input-group" action="/search">
                    <input type="text" name="q" id="q" class="form-control form-control-lg"
placeholder="Search">
                    <button class="btn btn-primary" id="search_btn" type="submit"><i class="bi bi-
search"></i></button>
                </form>
            </div>
            <!-- Cart -->
            <div class="nav-item d-none d-xl-block">
                <a class="nav-link" href="/cart">
                    <span class="cart-list" data-cart-items="{{request.session.cartdata|length}}">
                        <i class="bi bi-cart" id="cart-btn"></i>
                    </span>
                </a>
            </div>
            <!-- Acount -->
            <div class="nav-item dropdown">
                <a class="nav-link" href="#" role="button" id="dropdown_myaccount" data-bs-
toggle="dropdown" aria-haspopup="true" aria-expanded="false">
                    <i class="bi bi-person-circle"> </i>
                </a>
                <div class="dropdown-menu dropdown-menu-end mt-2 shadow" aria-
labelledby="dropdown_myaccount">
                    {% if user.is_authenticated %}
                        <a class="dropdown-item" href="{%url 'my_dashboard'%}">My Account</a>
                        <a class="dropdown-item" href="{%url 'checkout'%}">Checkout</a>
                        <a class="dropdown-item" href="/logout">Logout</a>
                    {% else %}
                        <a class="dropdown-item" id="login-btn" href="{%url 'login'%}">Login</a>
                        <a class="dropdown-item" id="signup-btn" href="{%url
'signup'%}">Register</a>
                    {% endif %}
                </div>
            </div>
            <!-- Mobile Toggle -->
            <button class="navbar-toggler ms-2" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false"
aria-label="Toggle navigation">
                <span class="navbar-toggler-icon"></span>
            </button>
            <!-- End Mobile Toggle -->
        </div>
    </div>
    </nav>
</header>
<!-- Header End -->
<!-- Main -->
<main>
```

```
{% block content %}
{% endblock %}
<div id="mybutton">
  <button id="voice_start_btn">
    start speaking
  </button>
  {% comment %} <button id="voice_start_btn">
    stop speaking
  </button> {% endcomment %}
</div>
</main>
<!-- End Main -->
<!-- Footer -->
<footer class="bg-dark footer">
  <div class="footer-top py-6">
    <div class="container">
      <div class="row">
        <div class="col-md-6 col-lg-4 my-3">
          <div class="pb-3">
            <img src="{%static '/img/logooo.png'%}" title="" alt="">
          </div>
          <address class="white-link mb-4">
            <p class="mb-2"><a class="border-bottom border-secondary"
href="mailto:support@domain.com">support@speakandshop.com</a></p>
            <p class="mb-2"><a class="border-bottom border-secondary" href="tel:820-885-
3321">+91234567890</a></p>
          </address>
          <div class="nav white-link fs-5">
            <a class="me-3" href="#">
              <i class="bi bi-facebook"></i>
            </a>
            <a class="me-3" href="#">
              <i class="bi bi-twitter"></i>
            </a>
            <a class="me-3" href="#">
              <i class="bi bi-instagram"></i>
            </a>
            <a class="me-3" href="#">
              <i class="bi bi-linkedin"></i>
            </a>
          </div>
        </div>
        <div class="col-sm-6 col-lg-2 my-3">
          <h6 class="text-white text-uppercase mb-5">
            Support
          </h6>
          <ul class="list-unstyled white-link footer-link-1 m-0">
            <li class="pb-2">
```

```html
            <a href="#">Help</a>
          </li>
          <li class="pb-2">
            <a href="#">Contact Us</a>
          </li>
          <li class="pb-2">
            <a href="#">Feedback</a>
          </li>
        </ul>
      </div>
      <div class="col-sm-6 col-lg-2 my-3">
        <h6 class="text-white text-uppercase mb-5">
          Policies
        </h6>
        <ul class="list-unstyled white-link footer-link-1 m-0">
          <li class="pb-2">
            <a href="#">Privacy Policy</a>
          </li>
          <li class="pb-2">
            <a href="#">Terms of use</a>
          </li>
          <li class="pb-2">
            <a href="#">Return</a>
          </li>
        </ul>
      </div>
      <div class="col-md-6 col-lg-4 my-3">
        <h6 class="text-white text-uppercase mb-5">
          Search Product
        </h6>
        <form class="d-flex flex-row mb-2 p-1 bg-white shadow-lg input-group rounded-3"
action="/search">
          <input type="text" class="form-control form-control-lg rounded-0 border-0
shadow-none" name="q" id="q" placeholder="Search items, collections, and creators">
          <button class="btn btn-primary rounded-3 flex-shrink-0 shadow-none border-0"
type="submit"><i class="bi bi-search"></i></button>
        </form>
      </div>
    </div>
  </div>
</div>
<div class="footer-bottom py-2 border-style light top small">
  <div class="container">
    <div class="row">
      <div class="col-md-6 text-center text-md-start py-1">
        <p class="m-0 text-white-65">© <script>document.write(new
Date().getFullYear())</script> copyright by <a href="/" class="text-reset">SpeakAndShop</a></p>
      </div>
```

```
            <div class="col-md-6 text-center text-md-end py-1">
               <img src="{%static '/img/payments.png'%}" title="" alt="">
            </div>
          </div>
        </div>
      </div>
    </footer>
    <!-- End Footer -->


    <!--bootstrap-->
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
ka7Sk0Gln4gmtz2MlQnikT1wXgYsOg+OMhuP+IlRH9sENBO0LRn5q+8nbTov4+1p"
crossorigin="anonymous"></script>
    <!-- owl-carousel -->
    <script src="{%static '/vendor/owl-carousel/js/owl.carousel.min.js'%}"></script>
    <!-- magnific -->
    <script src="{%static '/vendor/magnific/jquery.magnific-popup.min.js'%}"></script>
    <!-- working form -->
    <script src="{%static '/vendor/mail/js/form.min.js'%}"></script>
    <script src="{%static '/vendor/mail/js/script.js'%}"></script>
    <!-- Theme Js -->
    <script src="{%static '/js/custom.js'%}"></script>
    <script src="{%static '/js/voice.js'%}"></script>
</body>
  <!-- end body -->
</html>
```
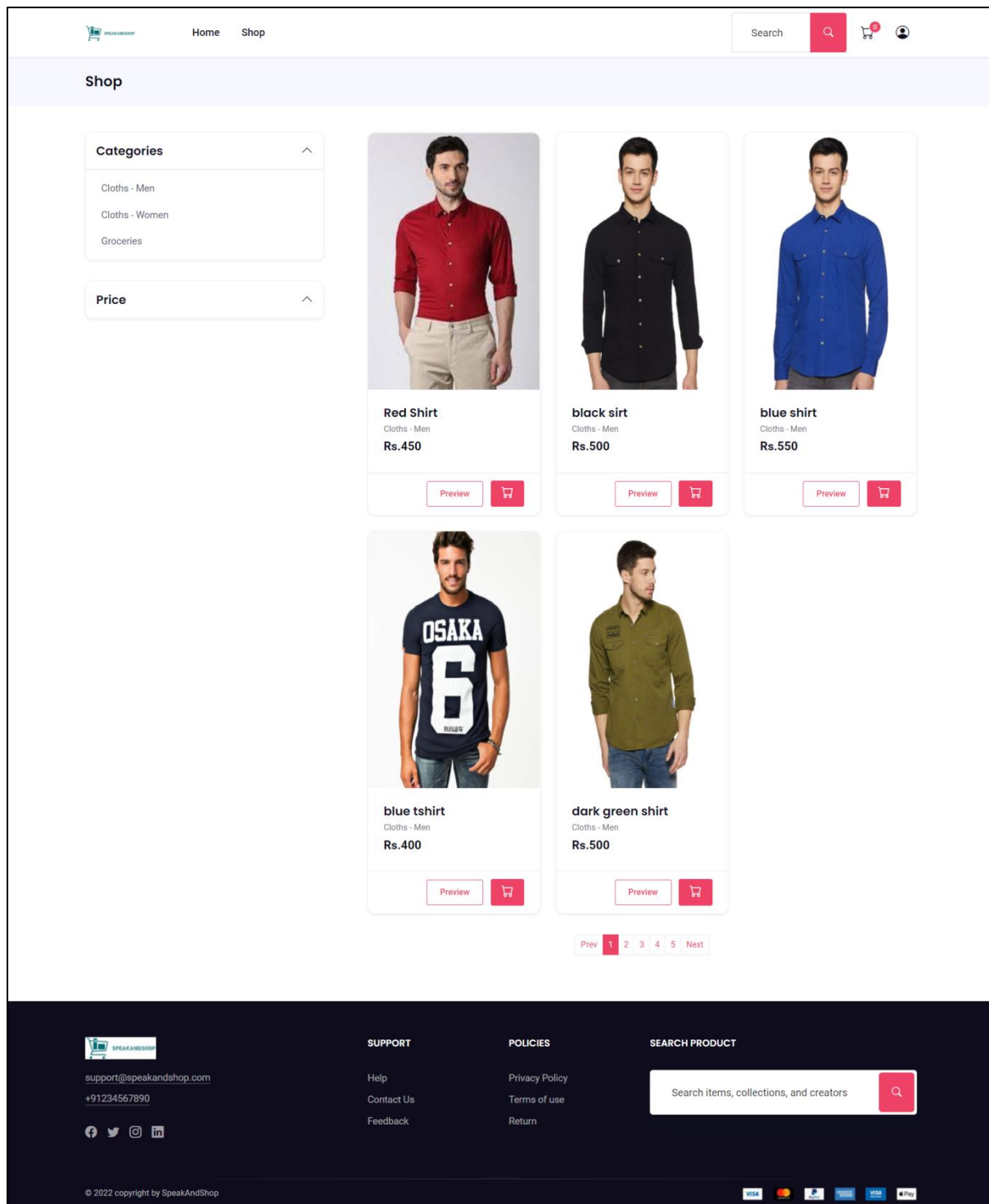
## Index.html

```
{% extends "base.html" %}
{%load static%}
{% block title %}SpeakAndShop{% endblock %}
{% block content %}
<script type="text/javascript" src="{%static 'ajax/cart2.js'%}"></script>
<!-- Section -->
{% comment %} <section class="section bg-cover bg-light">
  <div class="container">
    <div class="row justify-content-center">
      <div class="col-lg-7 col-xl-6 text-center">
        <form class="d-flex flex-row mb-2 p-1 bg-white shadow-lg input-group rounded-3"
action="/search">
           <input type="text" class="form-control form-control-lg rounded-0 border-0 shadow-none"
name="q" id="q" placeholder="Search items, collections, and creators">
           <button class="btn btn-primary rounded-3 flex-shrink-0 shadow-none border-0"
type="submit"><i class="bi bi-search"></i></button>
        </form>
```

```
            </div>
        </div>
    </div>
</section> {% endcomment %}
<!-- End Section -->
<!--Latest Product Section -->
<section class="section bg-gray-100">
  <div class="container">
    <div class="row mb-5 align-items-center">
      <div class="col">
        <h5 class="mb-0">Latest Products</h5>
      </div>
      <div class="col text-end">
        <a class="btm btn-sm btn-primary" href="/shop">View all</a>
      </div>
    </div>
    <!-- Products -->
    <div class="row g-3 g-xl-4">
     {%for product in latest%}
     <div class="col-md-6 col-lg-4" >
       <div class="card shadow-hover " >
         <img class="card-img-top " src="images/{{product.image}}" style="height:25rem" title=""
alt="">
         <div class="card-body ">
           <h5 class="m-0"><a class="text-reset text-truncate d-block w-100"
href="#">{{product.name}}</a></h5>
           <span class="small">{{product.category}}</span>
           <div class="font-w-600 text-dark fs-5">
            Rs.{{product.price}}
           </div>
         </div>
         <div class="card-footer border-top d-flex ">
           {% comment %} <div>
             <span>220 Sale</span>
             <div class="text-warning small">
               <i class="bi bi-star-fill small"></i>
               <i class="bi bi-star-fill small"></i>
               <i class="bi bi-star-fill small"></i>
               <i class="bi bi-star-fill small"></i>
               <i class="bi bi-star-fill small"></i>
               <span class="text-body small">(4k)</span>
             </div>
           </div> {% endcomment %}
           <div class="ms-auto position-relative z-index-1">
             <a class="btn btn-outline-primary btn-sm me-2" id="product{{product.id}}"
href="/product/{{product.slugs}}">Preview</a>
```

```
            <a class="btn btn-primary btn-sm px-3 cart{{product.id}}" id="addToCart" data-
id="{{product.id}}" data-name="{{product.name}}" data-price="{{product.price}}" data-
image="{{product.image}}"><i class="bi bi-cart fs-5 lh-1"></i></a>
          </div>
        </div>
      </div>
   </div>
   {% endfor %}
  </div>
  <!-- End Products -->
 </div>
</section>
<!--Latest Product End Section -->
<!-- Section -->
<section class="section">
 <div class="container">
   <div class="row mb-5 align-items-center">
     <div class="col">
       <h5 class="mb-0">Products </h5>
     </div>
     <div class="col text-end">
       <a class="btm btn-sm btn-primary" href="/shop">View all</a>
     </div>
   </div>
   <!-- Products -->
   <div class="row g-3 g-xl-4">
     {%for product in products%}
     <div class="col-md-6 col-lg-4" >
       <div class="card shadow-hover " >
         <img class="card-img-top " src="images/{{product.image}}" style="height:25rem"
title="" alt="">
         <div class="card-body ">
           <h5 class="m-0"><a class="text-reset text-truncate d-block w-100"
href="#">{{product.name}}</a></h5>
           <span class="small">{{product.category}}</span>
           <div class="font-w-600 text-dark fs-5">
            Rs.{{product.price}}
           </div>
         </div>
         <div class="card-footer border-top d-flex ">
           {% comment %} <div>
             <span>220 Sale</span>
             <div class="text-warning small">
               <i class="bi bi-star-fill small"></i>
               <i class="bi bi-star-fill small"></i>
               <i class="bi bi-star-fill small"></i>
               <i class="bi bi-star-fill small"></i>
               <i class="bi bi-star-fill small"></i>
```

```
              <span class="text-body small">(4k)</span>
            </div>
          </div> {% endcomment %}
          <div class="ms-auto position-relative z-index-1">
            <a class="btn btn-outline-primary btn-sm me-2" id="product{{product.id}}"
href="product/{{product.slugs}}">Preview</a>
                <a class="btn btn-primary btn-sm px-3 cart{{product.id}}" id="addToCart" data-
id="{{product.id}}" data-name="{{product.name}}" data-price="{{product.price}}" data-
image="{{product.image}}"><i class="bi bi-cart fs-5 lh-1"></i></a>
            </div>
          </div>
        </div>
      </div>
    {% endfor %}
  </div>
  <!-- End Products -->
 </div>
</section>
<!-- End Section -->
{% comment %} <div class="container-fluid mt-3">
<div class="row">
  <!-- filters -->
  <div class="col-lg-2 mx-auto">
    <a href="/" class="list-group-item d-flex justify-content-between align-items-center">
      All Products
    </a>
    <ul class="list-group">
      {%for category in categories%}
    <a href="/?category={{category.id}}" class="list-group-item d-flex justify-content-between
align-items-center">
      {{category.name}}
    </a>
      {% endfor %}
    </ul>
  </div>
  <!-- products -->
  <div id="product" class="col-lg-10 mx-auto">
      <div class="row mx-auto">
        {%for product in products%}
        <div class="card mx-auto mb-4" style="width: 18rem;">
          <img src="images/{{product.image}}" class="card-img-top  h-75" alt="..." >
          <div class="card-body">
          <h5 class="card-title">{{product.name}}</h5>
          <p class="card-text">Rs.{{product.price}}</p>
          <a href="#" class="btn btn-primary btn-sm">Add to cart</a>
          <a href="product/{{product.slugs}}" class="btn btn-outline-success btn-sm">View
Product</a>
            </div>
```

```
          </div>
            {% endfor %}
          </div>
     </div>
</div>
</div> {% endcomment %}
{% comment %} <script>
   window.onload = function(){
     alert("hello");
    // var msg = new SpeechSynthesisUtterance('I see dead people!');
    //msg.voice = speechSynthesis.getVoices().filter(function(voice) { return voice.name ==
'Whisper'; })[2];
    //speechSynthesis.speak(msg);
    const speech = new SpeechSynthesisUtterance();
    // // var speech = window.speechSynthesis;
    //const allvoices = speech.getVoices();
    //speech.voice = allvoices[2];
    speech.text = "hello welcome to speak and shop";
    speech.volume = 1;
    window.speechSynthesis.speak(speech);
    }
</script> {% endcomment %}
{% endblock %}
```

**Shop.html**

{% extends "base.html" %}
{%load static%}

```
{% block title %}Store | SpeakAndShop{% endblock %}
{% block content %}
<script type="text/javascript" src="{%static 'ajax/cart2.js'%}"></script>
<script type="text/javascript" src="{%static 'ajax/pagination.js'%}"></script>
<!--Breadcrumb Start-->
<div class="py-3 bg-gray-100">
   <div class="container">
     <div class="row align-items-center">
       <div class="col-lg-6 my-2">
          <h1 class="m-0 h4 text-center text-lg-start">Shop</h1>
       </div>
     </div>
   </div>
</div>
<!--Breadcrumb end-->
<!-- Section -->
<section class="py-6">
   <div class="container">
     <div class="row g-4 flex-row-reverse">
       <div class="col-lg-8" id="jar">
         <!-- Products -->
         <div class="row g-3 g-xl-4 " id="filteredProducts">
           {%for product in products%}
           <div class="col-md-6 col-lg-4 content" >
             <div class="card shadow-hover " >

                <img class="card-img-top prod-img" src="/images/{{product.image}}" data-
value="{{product.image}}" style="height:25rem" title="" alt="">
                <div class="card-body ">
                  <h5 class="m-0"><a class="text-reset text-truncate d-block w-100"
href="#">{{product.name}}</a></h5>
                  <span class="small">{{product.category}}</span>
                  <input type="hidden" class="product-id" value="{{product.id}}"/>
                  <input type="hidden" class="product-name" value="{{product.name}}"/>
                  <div class="font-w-600 text-dark fs-5 price" data-value={{product.price}}>
                  Rs.{{product.price}}
                  </div>
                </div>
                <div class="card-footer border-top d-flex ">
                  <div class="ms-auto position-relative z-index-1">
                     <a class="btn btn-outline-primary btn-sm me-2" id="product{{product.id}}"
href="/product/{{product.slugs}}">Preview</a>
                        <button class="btn btn-primary btn-sm px-3 cart{{product.id}}"  id="addToCart"
data-id="{{product.id}}" data-name="{{product.name}}" data-price="{{product.price}}" data-
image="{{product.image}}"><i class="bi bi-cart fs-5 lh-1"></i></button>
```

```
          </div>
        </div>
      </div>
    </div>
    {% endfor %}
  </div>
  <!-- End Products -->
  <!-- Pagination Start -->
  {% comment %} <button   id="loadMore">Load More</button> {% endcomment %}
  <nav id="page-id">
    <ul class="pagination justify-content-center pagination-sm m-5">
    </ul>
   </nav>
  <!-- End PAgination -->
</div>
<!-- Sidebar -->
<script type="text/javascript" src="{%static 'ajax/filter.js'%}"></script>
<div class="col-lg-4 pe-xl-8">
  <!-- Categories -->
  <div class="card mb-5">
    <div class="card-header p-3">
      <a class="h5 d-flex m-0" data-bs-toggle="collapse" href="#shop_categories"
role="button" aria-expanded="true" aria-controls="shop_categories">Categories <i class="pe-none
bi bi-chevron-up ms-auto"></i></a>
    </div>
    <div class="collapse show" id="shop_categories">
      <div class="card-body border-top p-2">
        <ul class="nav flex-column">
          {% for cat in categories %}
          <li class="nav-item">
          <div class="nav-link filter" data-value="{{cat.id}}" id="filter" >
          {{cat.name}}</div>
          </li>
          {% endfor %}
        </ul>
      </div>
    </div>
  </div>
  <!-- End Categories -->
  <!-- End Tags -->
  <!-- Tags -->
  <div class="card mb-5">
    <div class="card-header p-3">
```

```html
                <a class="h5 d-flex m-0" data-bs-toggle="collapse" href="#shop_price" role="button"
aria-expanded="true" aria-controls="shop_price">Price <i class="pe-none bi bi-chevron-up ms-
auto"></i></a>
            </div>
            <div class="collapse " id="shop_price">
              <div class="card-body border-top p-3">
                <div class="d-flex align-items-center price-card">
                  <ul class="nav flex-column">
                    <li class="nav-item">
                    <div class="nav-link fprice" data-value="ascending" id="filter-price" >Low -
High</div>
                    <div class="nav-link fprice" data-value="descending" id="filter-price" >High -
Low</div>
                    {% comment %} <div class="nav-link filter-price" data-value="{{cat.id}}"
id="filter-price" >Newly Added</div> {% endcomment %}
                    </li>
                  </ul>
                </div>
              </div>
            </div>
          </div>
          <!-- End Tags -->
        </div>
        <!-- End Sidebar -->
      </div>
    </div>
</section>
<!-- End Section -->
{% endblock %}
```

## Cart2.js

```javascript
$(document).ready(function(){
  $(document).on('click',"#addToCart",function(){
    var _vm=(this);
    var _qty=1;
    var _productId=$(this).data('id');
    var _productName=$(this).data('name');
    var _productImage=$(this).data('image');
    var _productPrice=$(this).data('price');
    console.log(_qty,_productId,_productName,_productPrice,_productImage);
    $.ajax({
      url:'/add-to-cart',
      data:{
        'id':_productId,
```

```
                'image':_productImage,
                'qty':_qty,
                'title':_productName,
                'price':_productPrice
            },
            dataType:'json',
            beforeSend:function(){
                //_vm.attr('disabled',true);
            },
            success:function(res){
                $(".cart-list").text(res.totalitems);
                //_vm.attr('disabled',false);
                alertify.success("Product added to cart")
            }
        });
    });
});
```

## Search.html

{% extends "base.html" %}

{%load static%}

{% block title %}Search... | SpeakAndShop{% endblock %}

{% block content %}

<script type="text/javascript" src="{%static 'ajax/cart2.js'%}"></script>

```
<script type="text/javascript" src="{%static 'ajax/pagination.js'%}"></script>
<!--Breadcrumb Start-->
<div class="py-3 bg-gray-100">
   <div class="container">
     <div class="row align-items-center">
       <div class="col-lg-6 my-2">
         <h1 class="m-0 h4 text-center text-lg-start">Search</h1>
       </div>
     </div>
   </div>
</div>
<!--Breadcrumb end-->
<!-- Section -->
<section class="py-6">
   <div class="container">
     <div class="row g-4 flex-row-reverse">
       <div class="col-lg-8" id="jar">
         <!-- Products -->
         <div class="row g-3 g-xl-4" id="filteredProducts">
          <h3>Search Result for {{request.GET.q}}:</h3>
            {%for product in products%}
            <div class="col-md-6 col-lg-4 content" >
              <div class="card shadow-hover " >
                <img class="card-img-top " src="/images/{{product.image}}" style="height:25rem"
title="" alt="">
                <div class="card-body ">
                  <h5 class="m-0"><a class="text-reset text-truncate d-block w-100"
href="#">{{product.name}}</a></h5>
                  <span class="small">{{product.category}}</span>
                  <div class="font-w-600 text-dark fs-5">
                  Rs.{{product.price}}
                  </div>
                </div>
                <div class="card-footer border-top d-flex ">
                  <div class="ms-auto position-relative z-index-1">
                    <a class="btn btn-outline-primary btn-sm me-2" id="product{{product.id}}"
href="/product/{{product.slugs}}">Preview</a>
                    <a class="btn btn-primary btn-sm px-3 cart{{product.id}}" id="addToCart" data-
id="{{product.id}}" data-name="{{product.name}}" data-price="{{product.price}}" data-
image="{{product.image}}"><i class="bi bi-cart fs-5 lh-1"></i></a>
                  </div>
                </div>
              </div>
            </div>
            {% endfor %}
```

```
        <h4 class="m-5 text text-center text-warning">{{response}}</h4>
      </div>
      <!-- End Products -->
      <nav id="page-id">
        <ul class="pagination justify-content-center pagination-sm m-5">
        </ul>
      </nav>
    </div>
    <!-- Sidebar -->
    <script type="text/javascript" src="{%static 'ajax/filter.js'%}"></script>
    <div class="col-lg-4 pe-xl-8">
      <!-- Categories -->
      <div class="card mb-5">
        <div class="card-header p-3">
          <a class="h5 d-flex m-0" data-bs-toggle="collapse" href="#shop_categories"
role="button" aria-expanded="true" aria-controls="shop_categories">Categories <i class="pe-none
bi bi-chevron-up ms-auto"></i></a>
        </div>
        <div class="collapse show" id="shop_categories">
          <div class="card-body border-top p-2">
            <ul class="nav flex-column">
              {% for cat in categories %}
              <li class="nav-item">
              <div class="nav-link filter" data-value="{{cat.id}}" id="filter" >
              {{cat.name}}</div>
              </li>
              {% endfor %}
            </ul>
          </div>
        </div>
      </div>
      <!-- End Categories -->
      <!-- End Tags -->
      <!-- Tags -->
      <div class="card mb-5">
        <div class="card-header p-3">
          <a class="h5 d-flex m-0" data-bs-toggle="collapse" href="#shop_price" role="button"
aria-expanded="true" aria-controls="shop_price">Price <i class="pe-none bi bi-chevron-up ms-
auto"></i></a>
        </div>
        <div class="collapse " id="shop_price">
          <div class="card-body border-top p-3">
            <div class="d-flex align-items-center price-card">
              <ul class="nav flex-column">
```

```
                        <li class="nav-item">
                        <div class="nav-link filter-price" data-value="{{cat.id}}" id="filter-price" >Low -
High</div>
                        <div class="nav-link filter-price" data-value="{{cat.id}}" id="filter-price" >High -
Low</div>
                        {% comment %} <div class="nav-link filter-price" data-value="{{cat.id}}"
id="filter-price" >Newly Added</div> {% endcomment %}
                        </li>
                    </ul>
                </div>
            </div>
        </div>
    </div>
    <!-- End Tags -->
</div>
    <!-- End Sidebar -->
</div>
</div>
</section>
<!-- End Section -->
{% endblock %}
```
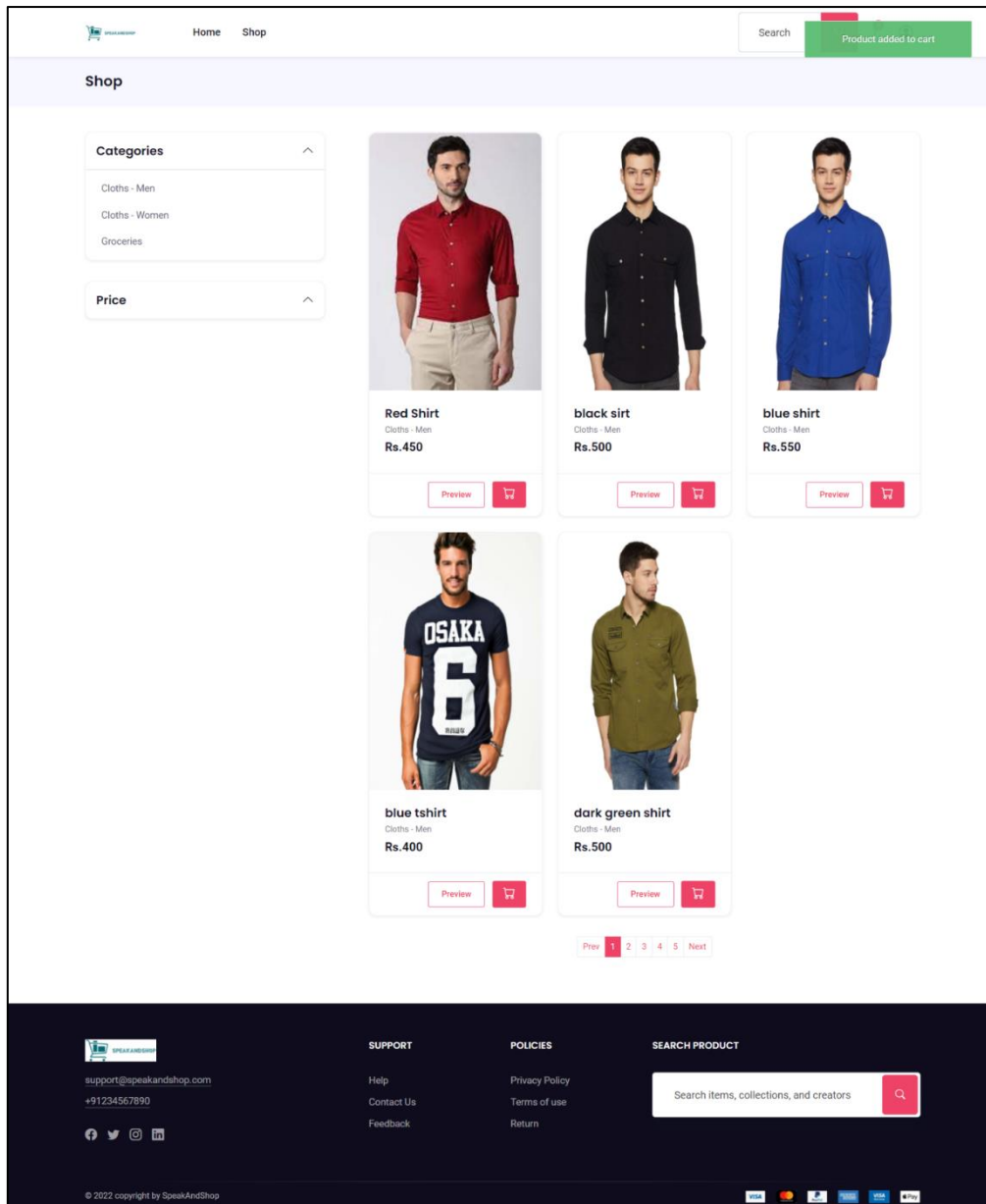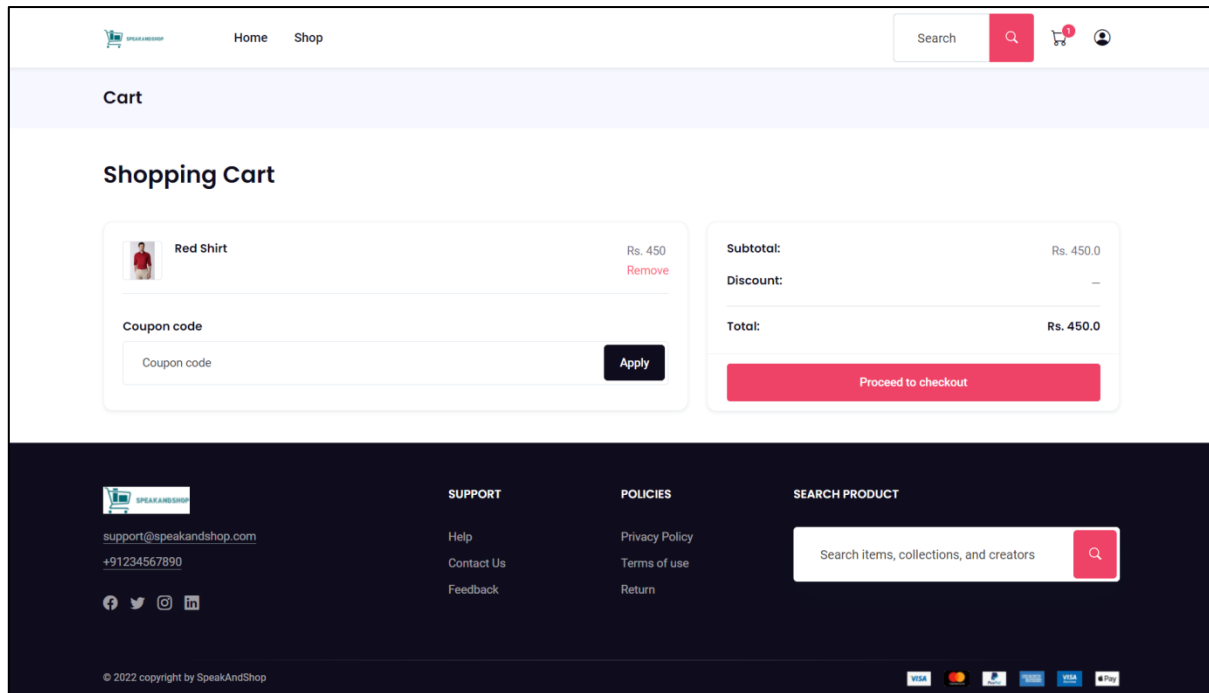
## Cart.js

```javascript
$(document).ready(function(){
    $(document).on('click',"#addToCart",function(){
        var _vm=(this);
        var _qty=1;
        var _productId=$(this).data('id');
        var _productName=$(this).data('name');
        var _productImage=$(this).data('image');
        var _productPrice=$(this).data('price');
        console.log(_qty,_productId,_productName,_productPrice,_productImage);
        $.ajax({
            url:'/add-to-cart',
            data:{
                'id':_productId,
                'image':_productImage,
                'qty':_qty,
                'title':_productName,
                'price':_productPrice
            },
            dataType:'json',
            beforeSend:function(){
```

```
      //_vm.attr('disabled',true);
    },
    success:function(res){
      $(".cart-list").text(res.totalitems);
      //_vm.attr('disabled',false);
      alertify.success("Product added to cart")
    }
  });
  });
});
```

**Cart.html**

```
{% extends "base.html" %}
{%load static%}
{% block title %}Cart | SpeakAndShop{% endblock %}
{% block content %}
<script type="text/javascript" src="{%static 'ajax/cart.js'%}"></script>
<!--Breadcrumb Start-->
<div class="py-3 bg-gray-100">
   <div class="container">
     <div class="row align-items-center">
       <div class="col-lg-6 my-2">
          <h1 class="m-0 h4 text-center text-lg-start">Cart</h1>
       </div>
     </div>
   </div>
</div>
<!--Breadcrumb end-->
<section class="py-6">
   <div class="container">
     <div class="pb-3">
       <h2 class="m-0">Shopping Cart</h2>
     </div>
     <div class="row">
       <div class="col-lg-7 mt-lg-4 mb-4 mb-lg-0">
          <div class="card cart-item-box">
            <div class="card-body">
              {% for product_id,item in cart_data.items %}
              <!-- Item -->
              <div class="d-flex border-bottom pb-3 mb-3">
                <div class="avatar border rounded">
                  <a href="#" class="d-block h-100">
                     <img src="/images/{{item.image}}" title="" alt="">
                  </a>
                </div>
                <div class="col ps-3">
                  <h6 class="m-0 font-w-600"><a class="text-reset"
href="/product/{{item.title|slugify}}">{{item.title}}</a></h6>
                  <div class="variation"></div>
                </div>
                <div class="ms-auto">
                  <span class="price">Rs. {{item.price}}</span><br />
                  <span class="remove">
                     <a class="link-danger delete-item "  data-item="{{product_id}}"
id="remove_btn_{{product_id}}">Remove</a>
                  </span>
                </div>
```

```
            </div>
            {% endfor %}
            <div class="pt-3">
               <h6>Coupon code</h6>
               <form class="d-flex flex-row mb-2 p-1 bg-white input-group rounded-3 border">
                  <input type="text" class="form-control rounded-0 border-0 shadow-none"
placeholder="Coupon code">
                     <button class="btn btn-dark rounded-3 flex-shrink-0 shadow-none border-0"
type="submit"> Apply</button>
               </form>
            </div>
          </div>
        </div>
      </div>
      <div class="col-lg-5 mr-auto mb-lg-0 mb-3 mt-lg-4">
        <div class="card ml-0 ml-lg-4">
          <div class="card-body">
            <div class="d-flex justify-content-between mb-3"><span class="h6 mb-
0">Subtotal:</span><span class="text-nav">Rs. {{total_amt}}</span></div>
               <div class="d-flex justify-content-between mb-3"><span class="h6 mb-
0">Discount:</span><span class="text-nav">—</span></div>
               <hr>
               <div class="d-flex justify-content-between"><span class="h6 mb-
0">Total:</span><span class="h6 mb-0">Rs. {{total_amt}}</span></div>
            </div>
            {% if totalitems %}
            <div class="card-footer">
               <a class="btn btn-primary w-100" href="{% url 'checkout' %}" id="checkout_btn">
                  Proceed to checkout
               </a>
            </div>
            {% endif %}
          </div>
        </div>
      </div>
    </div>
  </div>
</section>
{% endblock %}
```

## Cart.js

```javascript
$(document).ready(function(){
  $(document).on('click',"#addToCart",function(){
    var _vm=(this);
    var _qty=1;
    var _productId=$('.product-id').val();
    var _productName=$('.product-name').val();
    var _productImage=$('.prod-img').data('value');
    var _productPrice=$('.price').data('value');
    console.log(_qty,_productId,_productName,_productPrice,_productImage);
    $.ajax({
      url:'/add-to-cart',
      data:{
        'id':_productId,
        'image':_productImage,
        'qty':_qty,
        'title':_productName,
        'price':_productPrice
      },
      dataType:'json',
      beforeSend:function(){
        //_vm.attr('disabled',true);
      },
      success:function(res){
        $(".cart-list").text(res.totalitems);
```

```
            //_vm.attr('disabled',false);
            alertify.success("Product added to cart")
        }
    });
});
$(document).on('click','.delete-item',function(){
    var _pId=$(this).attr('data-item');
    var _vm=$(this);
    // Ajax
    $.ajax({
        url:'/delete-from-cart',
        data:{
            'id':_pId,
        },
        dataType:'json',
        beforeSend:function(){
            //_vm.attr('disabled',true);
        },
        success:function(res){
            $(".cart-list").text(res.totalitems);
            //_vm.attr('disabled',false);
            location.reload();
            alertify.success("Product removed from cart")
        }
    });
    // End
});
```

**Login.html**

```
{% extends "base.html" %}
{%load static%}
{% block title %}Login | SpeakAndShop{% endblock %}
{% block content %}
<!-- login page -->
<div class="section bg-gray-100">
    <div class="container">
        <div class="justify-content-center row">
            <div class="col-md-7 col-lg-5 col-xxl-4">
                <div class="card">
                    <div class="card-header bg-transparent py-3 border-bottom">
```
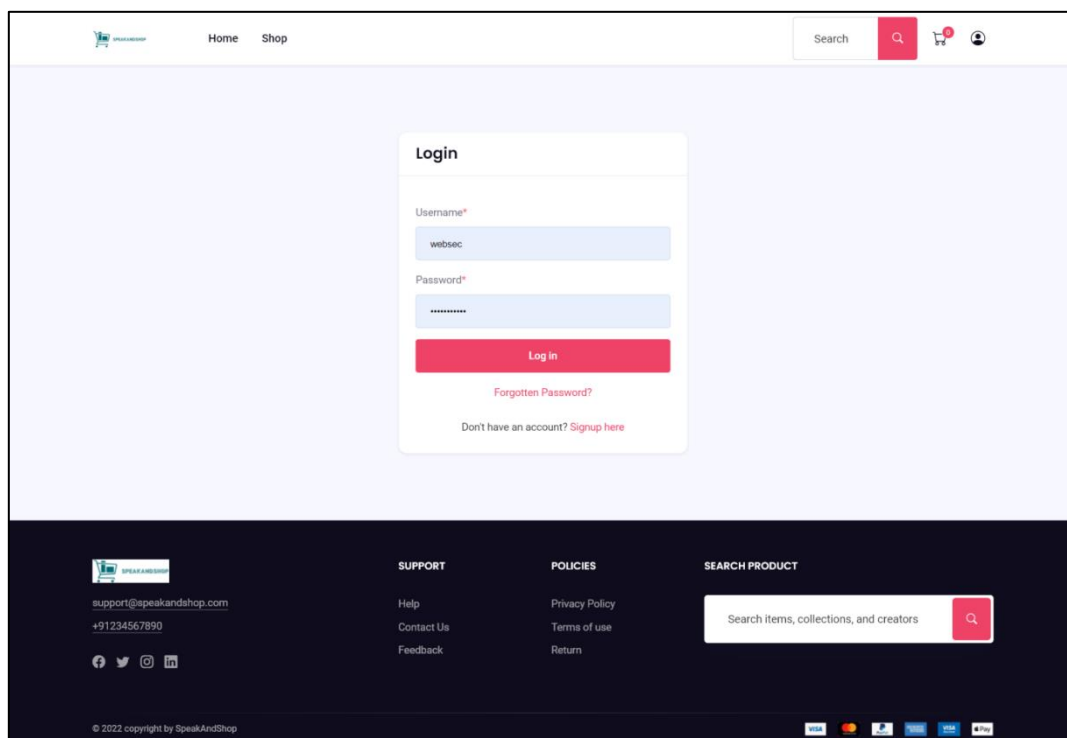


```
                        <h3 class="h4 mb-0">Login </h3>
                    </div>
                    <div class="card-body">
                        <form class="" method="POST">
                            {% csrf_token %}
                            <div class="form-group">
                            </div>
                            <div class="col-md-12">
                                <div class="form-group mb-3">
                                    <label for="name_1" class="form-label">Username<span class="text-
danger">*</span></label>
                                    <input type="text" id="username_id" name="username" class="form-control" >
```
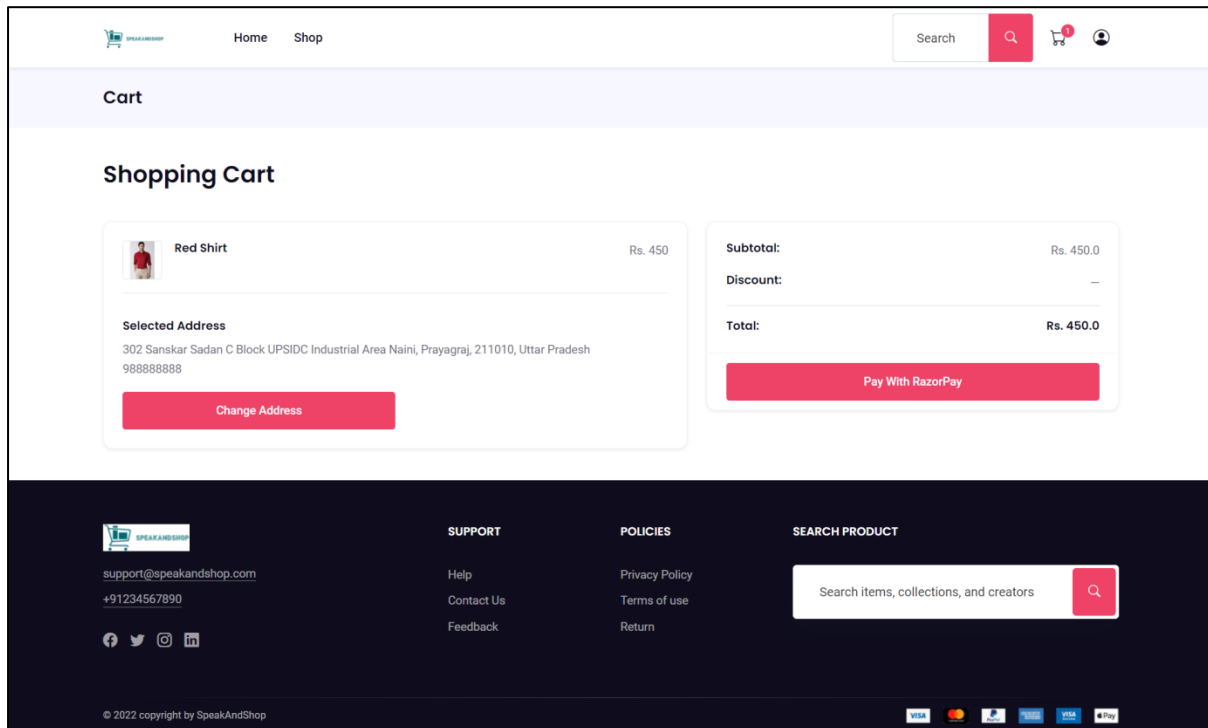
```
                </div>
              </div>
              <div class="col-md-12">
                <div class="form-group mb-3">
                  <label for="name_1" class="form-label">Password<span class="text-
danger">*</span></label>
                    <input type="password" id="password_id" name="password" class="form-
control" >
                </div>
              </div>
              {% comment %} <div class="form-group mb-3">
                <label for="email_1" class="form-label">Username<span class="text-
danger">*</span></label>
                  {{form.username}}
              </div>
              <div class="form-group mb-3">
                <div class="row align-items-center">
                  <label class="form-label col" for="exampleInputPassword01">Password<span
class="text-danger">*</span></label>
                    <span class="col ms-auto small text-end"><a href="#!">Forgot
password?</a></span>
                </div>
                {{form.password}}
              </div>   {% endcomment %}
              <div class="form-group text-center">
                <button type="submit" value="Login" class="btn btn-primary w-100">
                  Log in
                </button>
              </div>
            </form>
            <div class="pt text-center">
              <span class="text-muted"><a href="{% url 'reset_password'%}">Forgotten
Password? </a></span>
            </div>
            <div class="pt-4 text-center">
              <span class="text-muted">Don't have an account? <a href="{% url
'signup'%}">Signup here</a></span>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
<!-- end login -->
```

{% endblock %}

## Checkout.html



```
{% extends "base.html" %}
{%load static%}
{% block title %}Checkout | SpeakAndShop{% endblock %}
{% block content %}
<script type="text/javascript" src="{%static 'ajax/cart.js'%}"></script>
<script type="text/javascript" src="{%static 'ajax/checkout.js'%}"></script>
{% comment %} <script src="https://checkout.razorpay.com/v1/checkout.js"></script> {%
endcomment %}
<!--Breadcrumb Start-->
<div class="py-3 bg-gray-100">
    <div class="container">
        <div class="row align-items-center">
            <div class="col-lg-6 my-2">
                <h1 class="m-0 h4 text-center text-lg-start">Cart</h1>
            </div>
        </div>
    </div>
</div>
<!--Breadcrumb end-->
<section class="py-6">
    <div class="container">
        <div class="pb-3">
```

```
            <h2 class="m-0">Shopping Cart</h2>
        </div>
        <div class="row">
            <div class="col-lg-7 mt-lg-4 mb-4 mb-lg-0">
                <div class="card cart-item-box">
                    <div class="card-body">
                        {% for product_id,item in cart_data.items %}
                        <!-- Item -->
                        <div class="d-flex border-bottom pb-3 mb-3">
                            <div class="avatar border rounded">
                                <a href="#" class="d-block h-100">
                                    <img src="/images/{{item.image}}" title="" alt="">
                                </a>
                            </div>
                            <div class="col ps-3">
                                <h6 class="m-0 font-w-600"><a class="text-reset"
href="/product/{{item.title|slugify}}">{{item.title}}</a></h6>
                                <div class="variation"></div>
                            </div>
                            <div class="ms-auto">
                                <span class="price">Rs. {{item.price}}</span><br />
                            </div>
                        </div>
                        {% endfor %}
                        <div class="pt-3">
                            <h6>Selected Address</h6>
                            {{address.address}}
                            <br>
                            {{address.mobile}}
                            <br>
                            <a class="btn btn-primary w-50 mt-3" href="{%url 'my-addressbook'%}">Change
Address</a>
                        </div>
                    </div>
                </div>
            </div>
            <div class="col-lg-5 mr-auto mb-lg-0 mb-3 mt-lg-4">
                <div class="card ml-0 ml-lg-4">
                    <div class="card-body">
                        <div class="d-flex justify-content-between mb-3"><span class="h6 mb-
0">Subtotal:</span><span class="text-nav">Rs. {{total_amt}}</span></div>
                        <div class="d-flex justify-content-between mb-3"><span class="h6 mb-
0">Discount:</span><span class="text-nav">—</span></div>
                        <hr>
```

```
            <div class="d-flex justify-content-between"><span class="h6 mb-
0">Total:</span><span class="h6 mb-0">Rs. {{total_amt}}</span></div>
            <input id="ordid" value="{{order_id}}" hidden/>
            <input id="apiid" value="{{api_key}}" hidden/>
        </div>
        {% comment %} {% if totalitems %} {% endcomment %}
        <div class="card-footer">
            <button class="btn btn-primary w-100 paywithrazorpay" id="rzp-button1">
                Pay With RazorPay
            </button>
            <!-- Razorpay's Javascript code. -->
            <script src="https://checkout.razorpay.com/v1/checkout.js"></script>
            <script>
                var options = {
                // Enter the Key ID generated from the Dashboard
                key: "{{ api_key }}",
                // Amount is in currency subunits.
                // Default currency is INR. Hence,
                // 50000 refers to 50000 paise
                amount: "{{amount}}",
                currency: "INR",
                // Your/store name.
                name: "SpeakAndShop",
                prefill: {
                    "name": "{{name}}",
                    "email": "{{email}}",
                    "contact": "{{address.mobile}}"
                },
                // Pass the `id` obtained in the response of Step 1
                order_id: "{{ order_id }}",
                callback_url: "{{ callback_url }}",
                };
                // initialise razorpay with the options.
                var rzp1 = new Razorpay(options);
                rzp1.on('payment.failed', function (response){
                    error_code0=response.error.code;
                    error_desc0=response.error.description;
                    error_source0=response.error.source;
                    error_step0=response.error.step;
                    error_reason0=response.error.reason;
                    error_orderid0=response.error.metadata.order_id;
                    error_pid0=response.error.metadata.payment_id;
                    $.ajax({
                        url:'/payment-error',
                        data:{
```
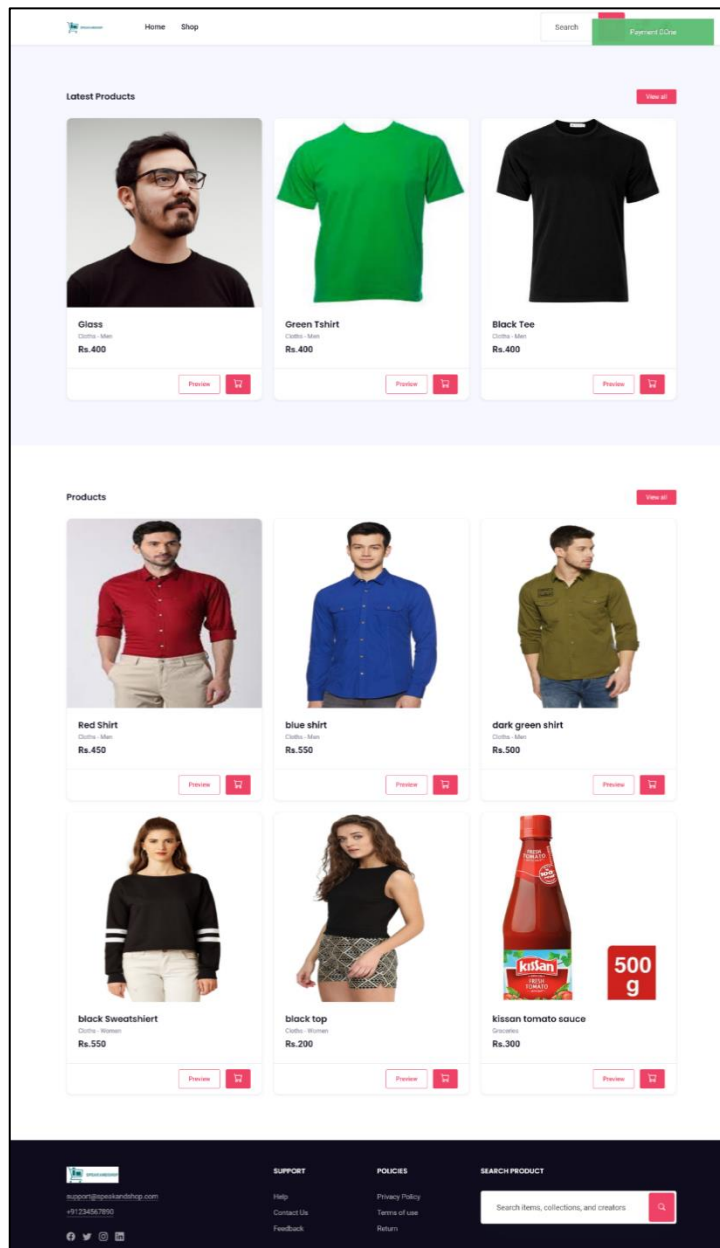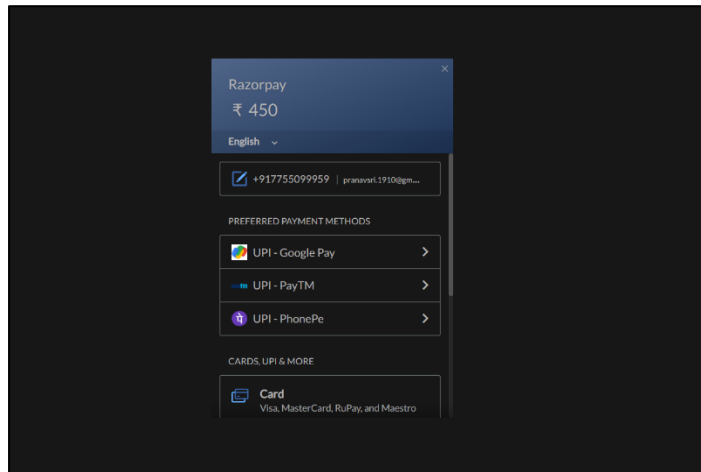
```
                            'error_code':error_code0,
                            'error_desc':error_desc0,
                            'error_source':error_source0,
                            'error_step':error_step0,
                            'error_reason':error_reason0,
                            'error_orderid':error_orderid0,
                            'error_pid':error_pid0
                        },
                        dataType:'json',
                        success:function(res){
                            $(".cart-list").text(res.totalitems);
                            //_vm.attr('disabled',false);
                            alertify.success("Product added to cart")
                        }
                    });
                });
                // add event listener to the payment button.
                document.getElementById("rzp-button1").onclick = function (e) {
                rzp1.open();
                e.preventDefault();
                };
            </script>
        {% comment %} {{form.render}} {% endcomment %}
    </div>
    {% comment %} {% endif %} {% endcomment %}
  </div>
 </div>
 </div>
</section>
{% endblock %}
```

## User-sidebar.html

```
<div class="list-group">
    <a href="{% url 'my_dashboard' %}" class="list-group-item {% if request.path == '/my-dashboard'
%}active {% endif %}">Dashboard</a>
    <a href="{% url 'edit-profile' %}" class="list-group-item {% if request.path == '/edit-profile' %}active
{% endif %}">Profile</a>
```
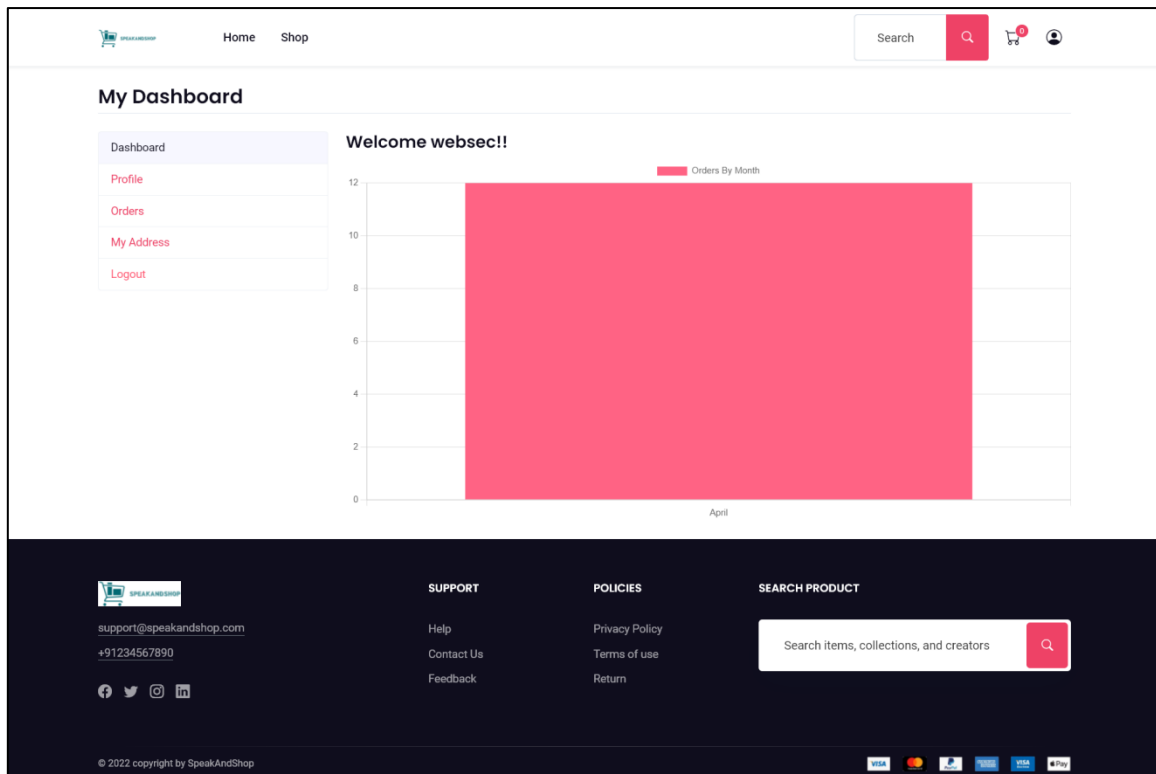
```
   <a href="{% url 'my_orders' %}" class="list-group-item {% if request.path == '/my-orders' %}active
{% endif %}">Orders</a>
   <a href="{% url 'my-addressbook' %}" class="list-group-item {% if request.path == '/my-
addressbook' %}active {% endif %}">My Address</a>
   <a href="{% url 'signout' %}" class="text-danger list-group-item">Logout</a>
</div>
```
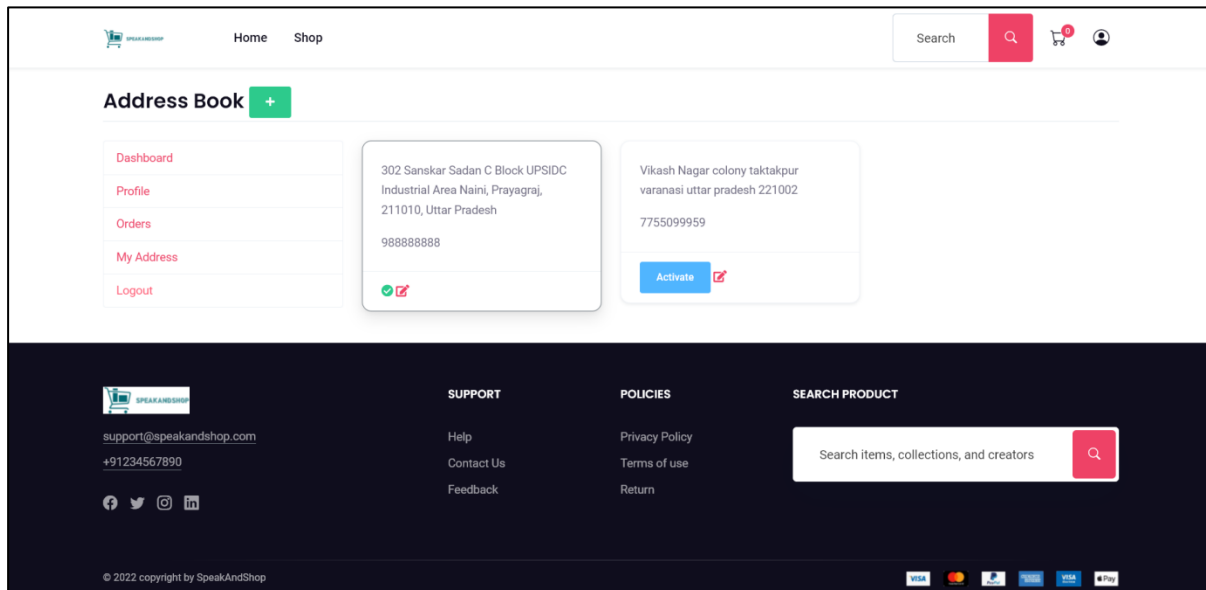
## Dashboard.html

```
{% extends "base.html" %}
{%load static%}
{% block title %}User Dashboard | SpeakAndShop{% endblock %}
{% block content %}
<main class="container my-4">
  <!-- Featured Products -->
  <h3 class="my-4 border-bottom pb-1">My Dashboard</h3>
  <div class="row">
    <div class="col-md-3">
      {% include 'user/user-sidebar.html' %}
    </div>
    <div class="col-md-9">
      <div>
        <h4>Welcome {{request.user}}!!</h4>
      <canvas id="myChart"></canvas>
      </div>
    </div>
  </div>
```

```
        </div>
        <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
<script type="text/javascript">
    const labels = {{monthNumber|safe}};
    const data = {
     labels: labels,
     datasets: [{
       label: 'Orders By Month',
       backgroundColor: 'rgb(255, 99, 132)',
       borderColor: 'rgb(255, 99, 132)',
       data: {{totalOrders}},
     }]
    };
    const config = {
     type: 'bar',
     data,
     options: {}
    };
    var myChart = new Chart(
        document.getElementById('myChart'),
        config
    );
</script>
</main>
{% endblock %}
```

## Orders.html

```
{% extends "base.html" %}
{%load static%}
{% block title %}Orders | SpeakAndShop{% endblock %}
{% block content %}
<main class="container my-4">
    <!-- Featured Products -->
    <h3 class="my-4 border-bottom pb-1">My Orders</h3>
    <div class="row">
        <div class="col-md-3">
            {% include 'user/user-sidebar.html' %}
        </div>
        <div class="col-md-9">
            <div class="table-responsive">
                <table class="table-bordered table">
                    <thead>
                        <tr>
                            <th>OrderNo</th>
```

```
                    <th>Total Amt</th>
                    <th>Order Status</th>
                    <th>Order Time</th>
                  </tr>
                </thead>
                <tbody>
                  {% for order in orders %}
                  <tr>
                    <td><a href="{% url 'my_order_items' order.id %}" class="btn btn-sm btn-
dark">{{order.id}}</a></td>
                      <td>{{order.total_amt}}</td>
                      <td>{{order.order_status}}</td>
                      <td>{{order.order_dt}}</td>
                  </tr>
                  {% endfor %}
                </tbody>
            </table>
          </div>
        </div>
      </div>
</main>
{% endblock %}
```
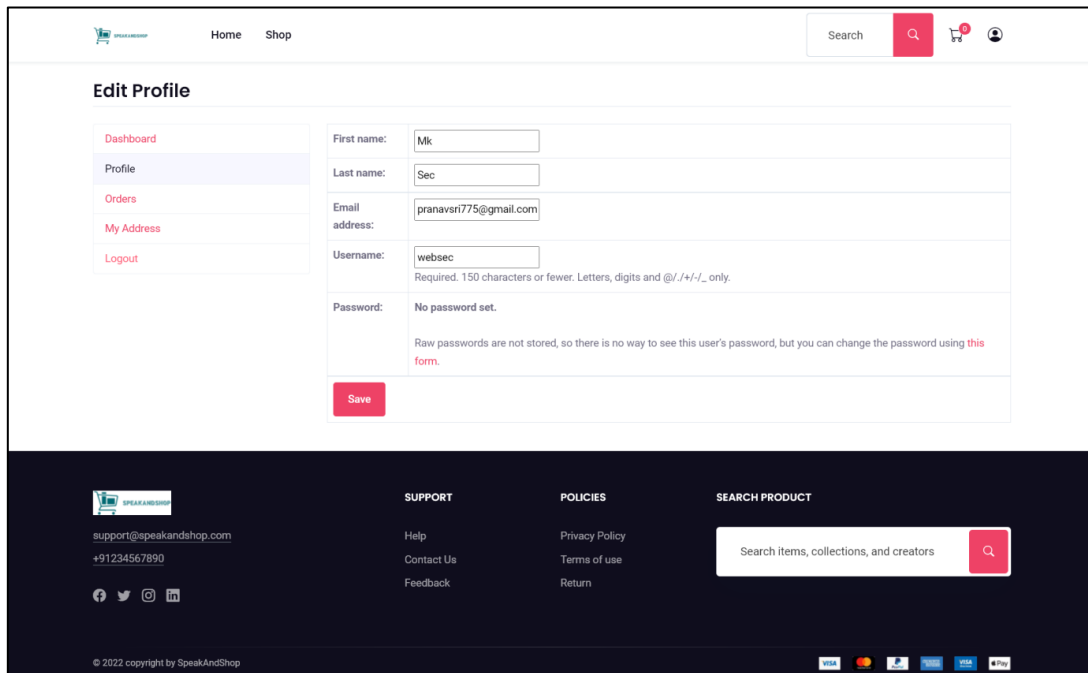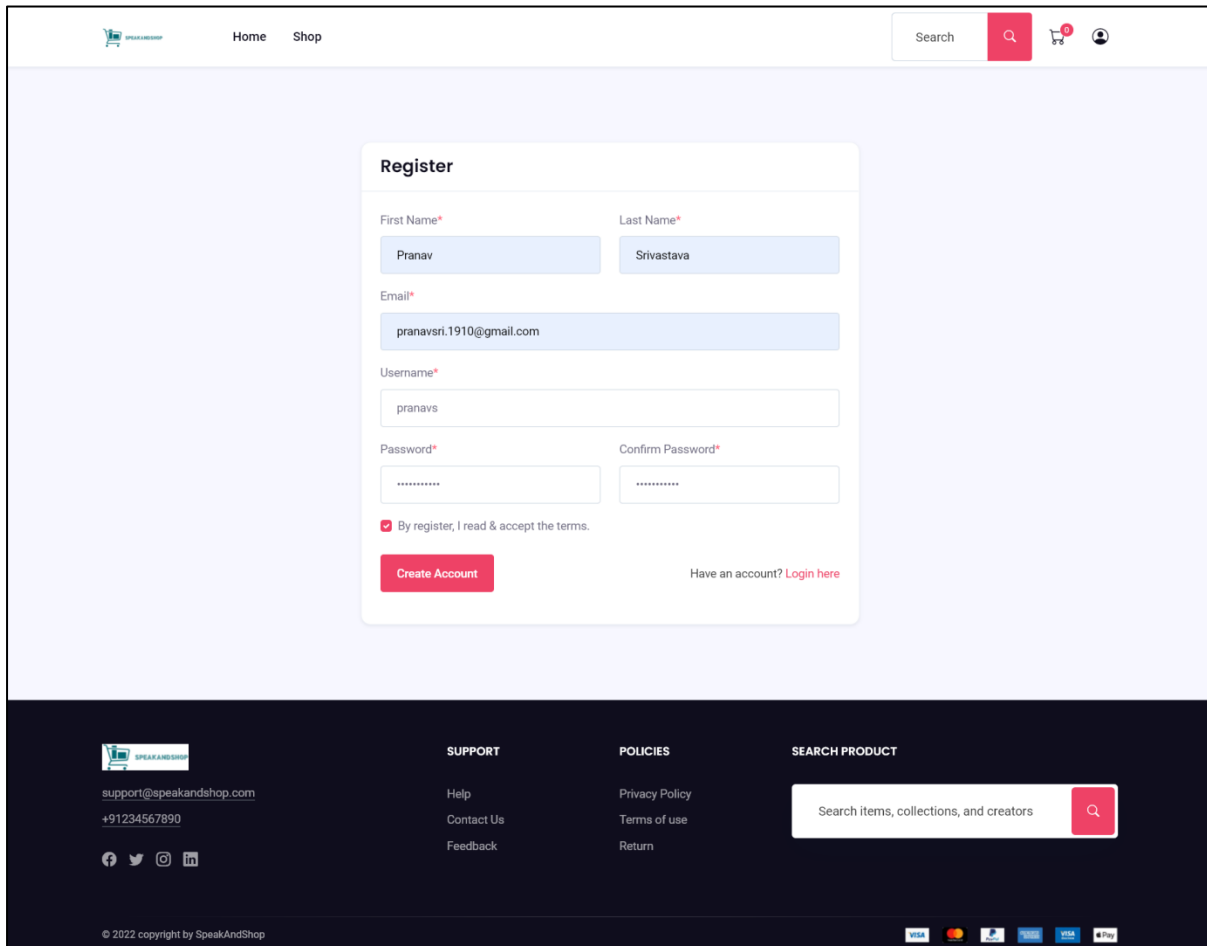
## Address-book.html

```
{% extends 'base.html' %}
{% load static %}
{% block title %}User Address Book Dashboard | SpeakAndShop{% endblock %}
{% block content %}
<script type="text/javascript" src="{%static 'ajax/address.js'%}"></script>
<main class="container my-4">
  <!-- Featured Products -->
  <h3 class="my-4 border-bottom pb-1">Address Book <a href="{% url 'add-address' %}"
class="float-right justify-center btn btn-sm btn-success"><i class="fa fa-plus"></i></a></h3>
  <div class="row">
    <div class="col-md-3">
      {% include 'user/user-sidebar.html' %}
    </div>
    <div class="col-md-9">
      <div class="row">
        {% if addbook %}
        {% else %}
        <div class="col-md-6">
        <a href="{% url 'add-address' %}" class="float-right justify-center btn btn-sm btn-
success">Click Here To Add Address <i class="fa fa-plus"></i></a>
```

```
            </div>
            {% endif %}
            {% for add in addbook %}
            <div class="col-md-4">

                    <div class="card mb-3 {% if add.status %}border-secondary shadow{% endif %} address{{add.id}} address">
                            <div class="card-body">
                                <p class="card-text">{{add.address}}</p>
                                <p class="card-text">{{add.mobile}}</p>
                            </div>
                            {% if add.status %}
                            <div class="card-footer">
                                <i class="fa fa-check-circle text-success check{{add.id}} check"></i>
                                <button style="display:none;" data-address="{{add.id}}" class="btn btn-sm btn-info activate-address btn{{add.id}} actbtn">Activate</button>
                                <a href="{% url 'update-address' add.id %}" class="float-right"><i class="fa fa-edit"></i></a>
                            </div>
                            {% else %}
                            <div class="card-footer footer{{add.id}}">
                                <i style="display:none;" class="fa fa-check-circle text-success check{{add.id}} check"></i>
                                <button data-address="{{add.id}}" class="btn btn-sm btn-info activate-address btn{{add.id}} actbtn">Activate</button>
                                <a href="{% url 'update-address' add.id %}" class="float-right"><i class="fa fa-edit"></i></a>
                            </div>
                            {% endif %}
                        </div>
```

```
            </div>
        {% endfor %}
        </div>
      </div>
    </div>
</main>
{% endblock %}
```

## Edit.html



```
{% extends 'base.html' %}
{% load static %}
{% block title %}User Profile Dashboard | SpeakAndShop{% endblock %}
{% block content %}

<main class="container my-4">
  <!-- Featured Products -->
  <h3 class="my-4 border-bottom pb-1">Edit Profile</h3>
  <div class="row">
    <div class="col-md-3">
      {% include 'user/user-sidebar.html' %}
    </div>
    <div class="col-md-9">
      <table class="table table-bordered">
        <form method="post">
          {% csrf_token %}
          {{form.as_table}}
```

```
            <tr>
              <td colspan="2">
                <input type="submit" value="Save" class="btn btn-primary" />
              </td>
            </tr>
          </form>
        </table>
      </div>
    </div>
</main>
{% endblock %}
```

## Register.html

```
{% extends "base.html" %}
{% block title %}User Registration{% endblock %}
```



```
{% block content %}
<!-- signup page -->
<div class="section bg-gray-100">
  <div class="container">
    <div class="justify-content-center row">
```

```html
        <div class="col-md-10 col-lg-8 col-xxl-6">
          <div class="card">
            <div class="card-header bg-transparent py-3 border-bottom">
              <h3 class="h4 mb-0">Register </h3>
            </div>
            <div class="card-body">
              <form class="" method="POST" >
                {% csrf_token %}
                <div class="row">
                  <div class="col-md-6">
                    <div class="form-group mb-3">
                      <label for="name_1" class="form-label">First Name<span class="text-danger">*</span></label>
                      {{form.first_name}}
                      {% if  form.errors.first_name%}
                        <label class="text-danger">{{form.errors.first_name}}</label>
                      {% endif %}
                    </div>
                  </div>
                  <div class="col-md-6">
                    <div class="form-group mb-3">
                      <label for="name_1" class="form-label">Last Name<span class="text-danger">*</span></label>
                      {{form.last_name}}
                      {% if  form.errors.last_name%}
                        <label class="text-danger">{{form.errors.last_name}}</label>
                      {% endif %}
                    </div>
                  </div>
                  <div class="col-md-12">
                    <div class="form-group mb-3">
                      <label for="name_1" class="form-label">Email<span class="text-danger">*</span></label>
                      {{form.email}}
                      {% if  form.errors.email%}
                        <label class="text-danger">{{form.errors.email}}</label>
                      {% endif %}
                    </div>
                  </div>
                  <div class="col-md-12">
                    <div class="form-group mb-3">
                      <label for="name_1" class="form-label">Username<span class="text-danger">*</span></label>
                      {{form.username}}
                      {% if  form.errors.username%}
```
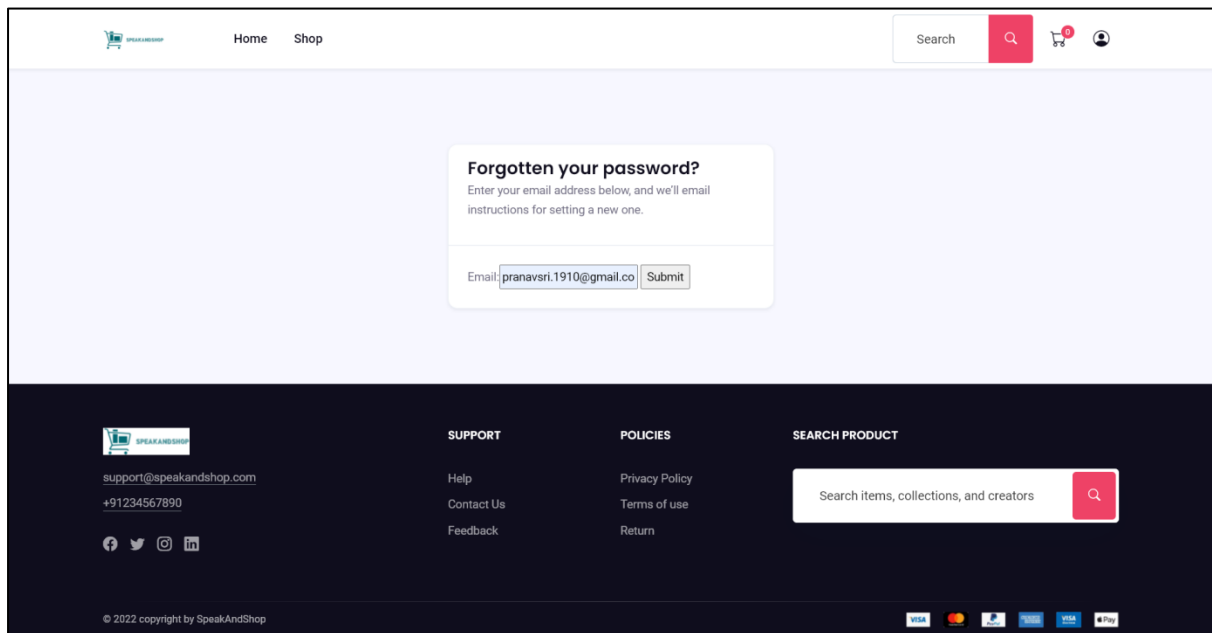
```html
                <label class="text-danger">{{form.errors.username}}</label>
                {% endif %}
              </div>
            </div>
            <div class="col-md-6">
              <div class="form-group mb-3">
                <label for="name_1" class="form-label">Password<span class="text-danger">*</span></label>
                  {{form.password1}}
                  {% if  form.errors.password1 %}
                    <label class="text-danger">{{form.errors.password1}}</label>
                  {% endif %}
              </div>
            </div>
            <div class="col-md-6">
              <div class="form-group mb-3">
                <label for="name_1" class="form-label">Confirm Password<span class="text-danger">*</span></label>
                  {{form.password2}}
                  {% if  form.errors.password2%}
                    <label class="text-danger">{{form.errors.password2}}</label>
                  {% endif %}
              </div>
            </div>
          </div>
          <div class="form-check mb-4">
            <input class="form-check-input" type="checkbox" value="" id="form2Example3" checked />
            <label class="form-check-label" for="form2Example3"> By register, I read & accept the terms. </label>
          </div>
          <div class="form-group row align-items-center">
            <div class="col">
              <button type="submit" class="btn btn-primary " >
                Create Account
              </button>
            </div>
            <div class="col-12 col-sm text-sm-end mt-3 mt-sm-0">
              <span class="text-muted">Have an account? <a href="{%url 'login'%}">Login here</a></span>
            </div>
          </div>
        </form>
      </div>
    </div>
```

```
                </div>
            </div>
        </div>
    </div>
<!-- end signup -->
{% endblock %}
```



## Forget.html

```
{% extends "base.html" %}
{%load static%}

{% block title %}Forgot Password | SpeakAndShop{% endblock %}
{% block content %}
<!-- login page -->
<div class="section bg-gray-100">
    <div class="container">
        <div class="justify-content-center row">
            <div class="col-md-7 col-lg-5 col-xxl-4">
                <div class="card">
                    <div class="card-header bg-transparent py-3 border-bottom">
                        <h3 class="h4 mb-0">Forgotten your password?<br></h3>
                        <p> Enter your email address below, and we'll email instructions for setting a new one.
<p>
                    </div>
                    <div class="card-body">
                        {% comment %} <form class="" method="POST">
                            {% csrf_token %}
```

```
                    <div class="form-group">
                    </div>
                    <div class="col-md-12">
                      <div class="form-group mb-3">
                        <label for="name_1" class="form-label">Username<span class="text-
danger">*</span></label>
                          <input type="text" id="username_id" name="username" class="form-control"
>
                      </div>
                    </div>
                    <div class="col-md-12">
                      <div class="form-group mb-3">
                        <label for="name_1" class="form-label">Password<span class="text-
danger">*</span></label>
                          <input type="password" id="password_id" name="password" class="form-
control" >
                      </div>
                    </div>
                    <div class="form-group text-center">
                      <button type="submit" value="Login" class="btn btn-primary w-100">
                        Log in
                      </button>
                    </div>
                  </form> {% endcomment %}
                  <form method="post">
                    {% csrf_token %}
                    {{form}}
                    <input type="Submit" name="Send email">
                  </form>
                </div>
              </div>
            </div>
          </div>
        </div>
<!-- end login -->
{% endblock %}
```

## Password_reset_sent.html

```
{% extends "base.html" %}
{%load static%}
{% block title %}Reset Password | SpeakAndShop{% endblock %}
{% block content %}
```

```
<div class="container m-5">
<h3>Password reset sent</h3>
<p>We've emailed you instructions for setting your password, if an account exists with the email you
entered. You should receive them shortly.</p>

<p>If you don't receive an email, please make sure you've entered the address you registered with,
and check your spam folder.</p>
```
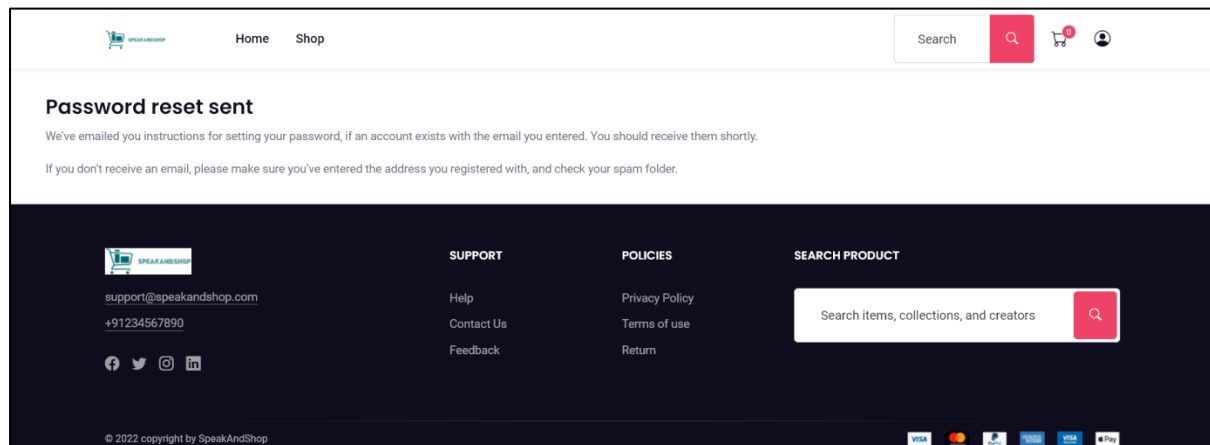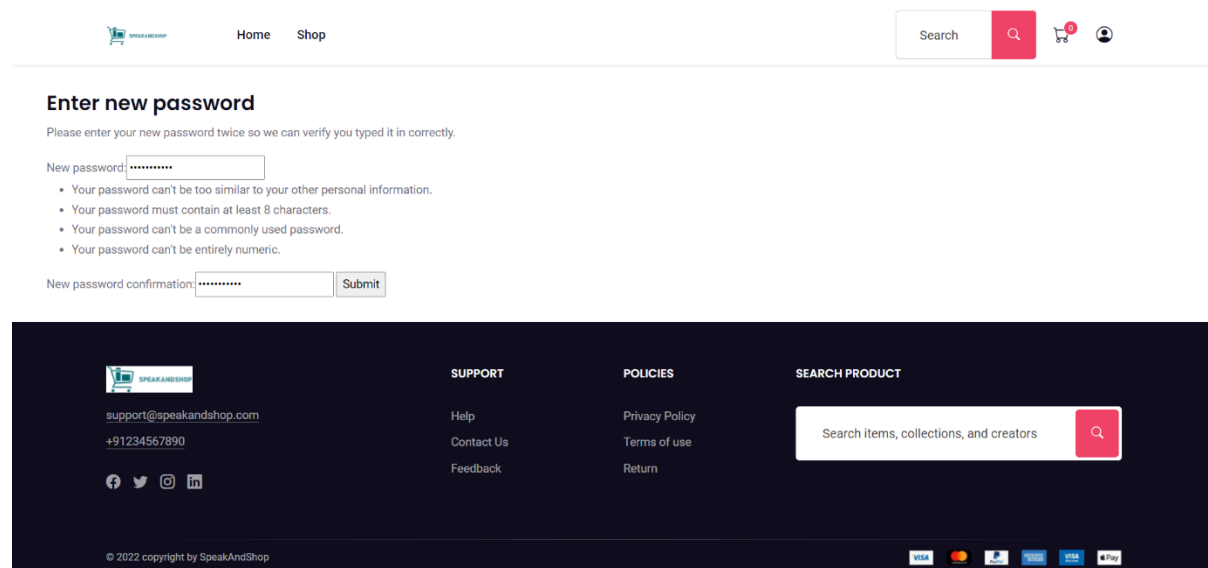


```
</div>
{% endblock %}
```
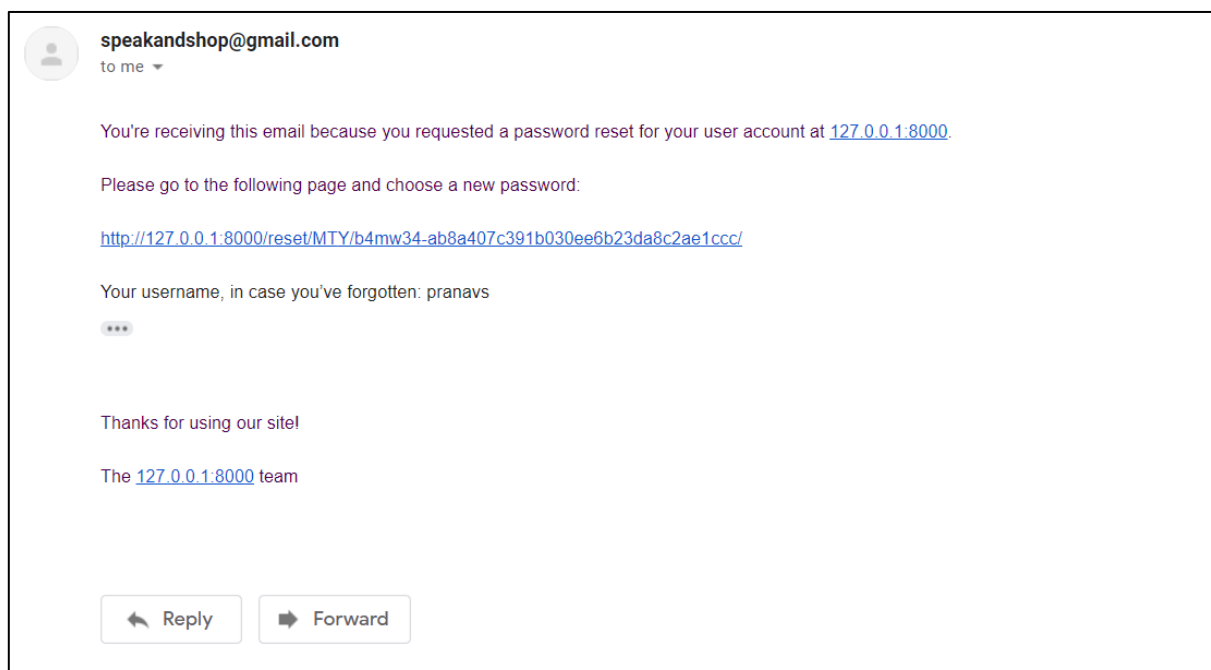
## Reset Password form.html

{% extends "base.html" %}
{% load static %}
{% block title %}Change Password | SpeakAndShop{% endblock %}
{% block content %}
<div class="container m-5">
<h3>Enter new password</h3>
<p>Please enter your new password twice so we can verify you typed it in correctly.</p>
<form method="post">
    {% csrf_token %}
    {{form}}
    <input type="Submit" name="Update Password">
</form>
</div>
{% endblock %}

# SELLER PART



## Register.html

{% extends "base.html" %}
{% block title %}Seller Registration{% endblock %}
{% block content %}
<!-- signup page -->
<div class="section bg-gray-100">
   <div class="container">
      <div class="justify-content-center row">
         <div class="col-md-10 col-lg-8 col-xxl-6">
            <div class="card">
               <div class="card-header bg-transparent py-3 border-bottom">
                  <h3 class="h4 mb-0">Seller Register </h3>
               </div>
               <div class="card-body">
                  <form class="" method="POST" >
                     {% csrf_token %}

```html
<div class="row">
  <div class="col-md-6">
    <div class="form-group mb-3">
      <label for="name_1" class="form-label">First Name<span class="text-danger">*</span></label>
      {{form.first_name}}
      {% if  form.errors.first_name%}
        <label class="text-danger">{{form.errors.first_name}}</label>
      {% endif %}
    </div>
  </div>
  <div class="col-md-6">
    <div class="form-group mb-3">
      <label for="name_1" class="form-label">Last Name<span class="text-danger">*</span></label>
      {{form.last_name}}
      {% if  form.errors.last_name%}
        <label class="text-danger">{{form.errors.last_name}}</label>
      {% endif %}
    </div>
  </div>
  <div class="col-md-12">
    <div class="form-group mb-3">
      <label for="name_1" class="form-label">Email<span class="text-danger">*</span></label>
      {{form.email}}
      {% if  form.errors.email%}
        <label class="text-danger">{{form.errors.email}}</label>
      {% endif %}
    </div>
  </div>
  <div class="col-md-12">
    <div class="form-group mb-3">
      <label for="name_1" class="form-label">Username<span class="text-danger">*</span></label>
      {{form.username}}
      {% if  form.errors.username%}
        <label class="text-danger">{{form.errors.username}}</label>
      {% endif %}
    </div>
  </div>
  <div class="col-md-6">
    <div class="form-group mb-3">
      <label for="name_1" class="form-label">Password<span class="text-danger">*</span></label>
```

```
                {{form.password1}}
                {% if  form.errors.password1 %}
                   <label class="text-danger">{{form.errors.password1}}</label>
                {% endif %}
             </div>
          </div>
          <div class="col-md-6">
             <div class="form-group mb-3">
                <label for="name_1" class="form-label">Confirm Password<span class="text-
danger">*</span></label>
                {{form.password2}}
                {% if  form.errors.password2%}
                   <label class="text-danger">{{form.errors.password2}}</label>
                {% endif %}
             </div>
          </div>
          <div class="form-check mb-4">
             <input class="form-check-input" type="checkbox" value="" id="form2Example3"
checked />
             <label class="form-check-label" for="form2Example3"> By register, I read & accept
the terms. </label>
          </div>
          <div class="form-group row align-items-center">
             <div class="col">
                <button type="submit" class="btn btn-primary " >
                   Create Account
                </button>
             </div>
             <div class="col-12 col-sm text-sm-end mt-3 mt-sm-0">
                <span class="text-muted">Have an account? <a href="{%url 'login'%}">Login
here</a></span>
             </div>
          </div>
       </form>
     </div>
   </div>
  </div>
 </div>
</div>
<!-- end signup -->
{% endblock %}
```

## Seller.py(model)

```python
from django.db import models
from django.utils.html import mark_safe
from django.urls import reverse
from django.conf.urls import url
class Seller(models.Model):
    first_name=models.CharField(max_length=50)
    last_name=models.CharField(max_length=50)
    email=models.CharField(max_length=50)
    username=models.CharField(max_length=250,default='')
    password=models.CharField(max_length=250)
    class Meta:
        db_table="store_seller"
    def register(self):
        self.save()
    @staticmethod
    def get_seller_by_username(usr):
        try:
            return Seller.objects.get(username=usr)
        except:
            return False
    @staticmethod
    def isExists(username):
        if Seller.objects.filter(username = username):
            return True
        return  False
```

## seller_form

```python
class Sellerform(forms.ModelForm):
    password1 = forms.CharField(
    help_text='Enter Password',
    required = True,
    widget=forms.PasswordInput(attrs={'class': 'form-control', 'placeholder': 'Password'}),
    )
    password2 = forms.CharField(
    required = True,
    help_text='Enter Password Again',
    widget=forms.PasswordInput(attrs={'class': 'form-control', 'placeholder': 'Password Again'}),
    )
    class Meta:
        model=Seller
        fields=['first_name','last_name','email','username','password1','password2']
        widgets={
            'first_name':forms.TextInput(attrs={'class':'form-control '}),
            'last_name':forms.TextInput(attrs={'class':'form-control '}),
```
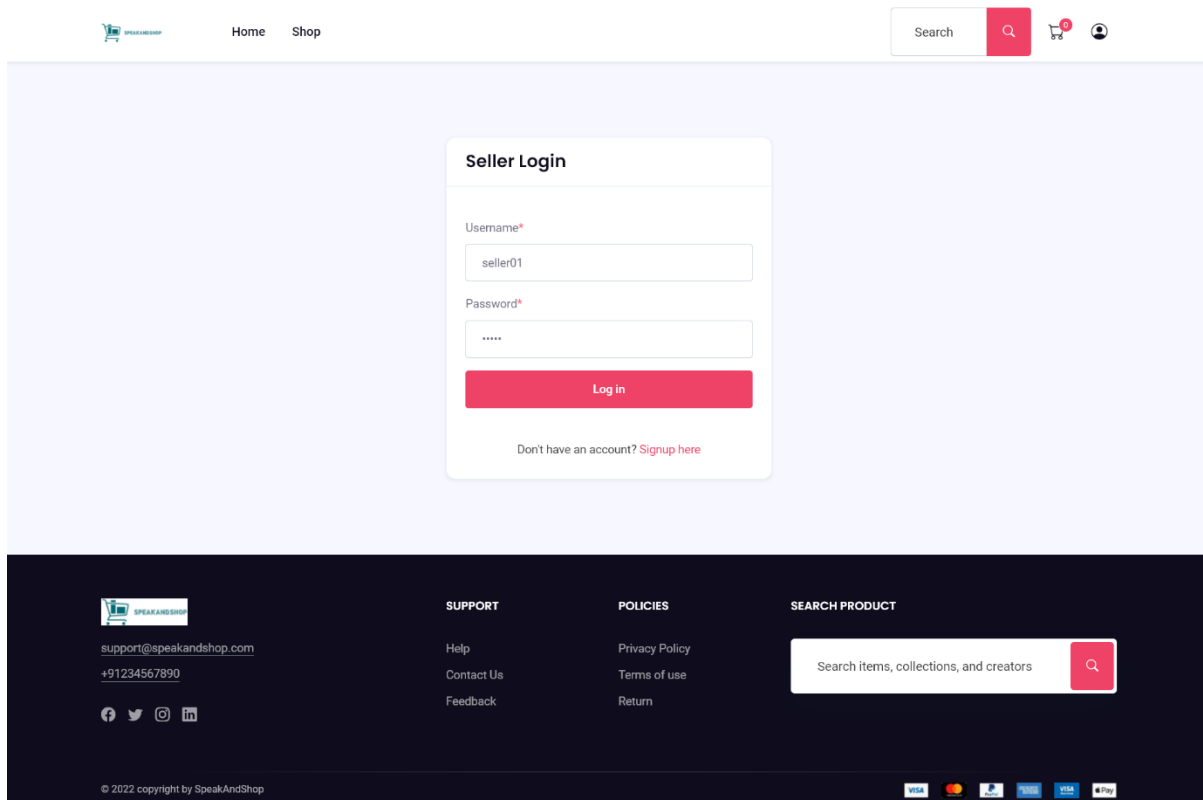
```
        'email':forms.EmailInput(attrs={'class':'form-control'}),
        'username':forms.TextInput(attrs={'class':'form-control'}),
    }
def save(self,commit=True):
    seller=super(Sellerform,self).save(commit=False)
    if commit:
        seller.save()
    return seller
```

## Login.html



```
{% extends "base.html" %}
{%load static%}
{% block title %}Seller Login | SpeakAndShop{% endblock %}
{% block content %}
<!-- login page -->
<div class="section bg-gray-100">
  <div class="container">
    <div class="justify-content-center row">
      <div class="col-md-7 col-lg-5 col-xxl-4">
        <div class="card">
          <div class="card-header bg-transparent py-3 border-bottom">
            <h3 class="h4 mb-0">Seller Login </h3>
          </div>
          <div class="card-body">
```

```
<form class="" method="POST">
  {% csrf_token %}
  <div class="form-group">
  </div>
  <div class="col-md-12">
    <div class="form-group mb-3">
      <label for="name_1" class="form-label">Username<span class="text-danger">*</span></label>
        <input type="text" id="username_id" name="username" class="form-control" >
    </div>
  </div>
  <div class="col-md-12">
    <div class="form-group mb-3">
      <label for="name_1" class="form-label">Password<span class="text-danger">*</span></label>
        <input type="password" id="password_id" name="password" class="form-control" >
    </div>
  </div>
  {% comment %} <div class="form-group mb-3">
    <label for="email_1" class="form-label">Username<span class="text-danger">*</span></label>
      {{form.username}}
  </div>
  <div class="form-group mb-3">
    <div class="row align-items-center">
      <label class="form-label col" for="exampleInputPassword01">Password<span class="text-danger">*</span></label>
        <span class="col ms-auto small text-end"><a href="#!">Forgot password?</a></span>
    </div>
    {{form.password}}
  </div>   {% endcomment %}
  <div class="form-group text-center">
    <button type="submit" value="Login" class="btn btn-primary w-100">
      Log in
    </button>
  </div>
</form>
{% comment %} <div class="pt text-center">
  <span class="text-muted"><a href="{% url 'reset_password'%}">Forgotten Password? </a></span>
</div> {% endcomment %}
<div class="pt-4 text-center">
```

```
                <span class="text-muted">Don't have an account? <a href="{% url
'seller_signup'%}">Signup here</a></span>
              </div>
            </div>
          </div>
        </div>
      </div>
</div>
<!-- end login -->
{% endblock %}
```

## Seller base.html

```
{% load static %}
<!DOCTYPE html>
<html lang="zxx">
<head>
  <!-- Title -->
  <title>{% block title %}
    {% endblock %}</title>
  <!-- Required Meta Tags Always Come First -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
  <meta http-equiv="x-ua-compatible" content="ie=edge">
  <!-- Favicon -->
  <link rel="shortcut icon" href="#">
  <!-- CSS Template -->
  <link href="{%static 'css/style.css'%}" rel="stylesheet">
  <!-- CSS Custom -->
  <link href="https://getbootstrap.com/docs/5.1/examples/sidebars/sidebars.css" rel="stylesheet">
  <!-- alertify CSS -->
  <link rel="stylesheet" href="//cdn.jsdelivr.net/npm/alertifyjs@1.13.1/build/css/alertify.min.css"/>
  <!-- alertify Default theme -->
  <link rel="stylesheet"
href="//cdn.jsdelivr.net/npm/alertifyjs@1.13.1/build/css/themes/default.min.css"/>
  <!-- jquery -->
  <script src="{%static '/js/jquery-3.5.1.min.js'%}"></script>
  <!--cart ajax js-->
  <style>
    .bd-placeholder-img {
     font-size: 1.125rem;
     text-anchor: middle;
     -webkit-user-select: none;
     -moz-user-select: none;
```

```
      user-select: none;
     }
     @media (min-width: 768px) {
      .bd-placeholder-img-lg {
       font-size: 3.5rem;
      }
     }
    </style>
</head>
<body>
   <script>
     alertify.set('notifier','position', 'top-right');
     {% for msg in messages %}
        alertify.success('{{msg}}');
     {% endfor %}
   </script>
   <svg xmlns="http://www.w3.org/2000/svg" style="display: none;">
     <symbol id="bootstrap" viewBox="0 0 118 94">
      <path fill-rule="evenodd" clip-rule="evenodd" d="M24.509 0c-6.733 0-11.715 5.893-11.492
12.284.214 6.14-.064 14.092-2.066 20.577C8.943 39.365 5.547 43.485 0 44.014v5.972c5.547.529
8.943 4.649 10.951 11.153 2.002 6.485 2.28 14.437 2.066 20.577C12.794 88.106 17.776 94 24.51
94H93.5c6.733 0 11.714-5.893 11.491-12.284-.214-6.14.064-14.092 2.066-20.577 2.009-6.504
5.396-10.624 10.943-11.153v-5.972c-5.547-.529-8.934-4.649-10.943-11.153-2.002-6.484-2.28-
14.437-2.066-20.577C105.214 5.894 100.233 0 93.5 0H24.508zM80 57.863C80 66.663 73.436 72
62.543 72H44a2 2 0 01-2-2V24a2 2 0 012-2h18.437c9.083 0 15.044 4.92 15.044 12.474 0 5.302-4.01
10.049-9.119 10.88v.277C75.317 46.394 80 51.21 80 57.863zM60.521
28.34H49.948v14.934h8.905c6.884 0 10.68-2.772 10.68-7.727 0-4.643-3.264-7.207-9.012-
7.207zM49.948 49.2v16.458H60.91c7.167 0 10.964-2.876 10.964-8.281 0-5.406-3.903-8.178-11.425-
8.178H49.948z"></path>
     </symbol>
     <symbol id="home" viewBox="0 0 16 16">
      <path d="M8.354 1.146a.5.5 0 0 0-.708 0l-6 6A.5.5 0 0 0 1.5 7.5v7a.5.5 0 0 0 .5.5h4.5a.5.5 0 0 0
.5-.5v-4h2v4a.5.5 0 0 0 .5.5H14a.5.5 0 0 0 .5-.5v-7a.5.5 0 0 0-.146-.354L13 5.793V2.5a.5.5 0 0 0-.5-
.5h-1a.5.5 0 0 0-.5.5v1.293L8.354 1.146zM2.5 14V7.707l5.5-5.5 5.5 5.5V14H10v-4a.5.5 0 0 0-.5-.5h-
3a.5.5 0 0 0-.5.5v4H2.5z"/>
     </symbol>
     <symbol id="speedometer2" viewBox="0 0 16 16">
      <path d="M8 4a.5.5 0 0 1 .5.5V6a.5.5 0 0 1-1 0V4.5A.5.5 0 0 1 8 4zM3.732 5.732a.5.5 0 0 1 .707
0l.915.914a.5.5 0 1 1-.708.708l-.914-.915a.5.5 0 0 1 0-.707zM2 10a.5.5 0 0 1 .5-.5h1.586a.5.5 0 0 1 0
1H2.5A.5.5 0 0 1 2 10zm9.5 0a.5.5 0 0 1 .5-.5h1.5a.5.5 0 0 1 0 1H12a.5.5 0 0 1-.5-.5zm.754-
4.246a.389.389 0 0 0-.527-.02L7.547 9.31a.91.91 0 1 0 1.302 1.258l3.434-4.297a.389.389 0 0 0-.029-
.518z"/>
      <path fill-rule="evenodd" d="M0 10a8 8 0 1 1 15.547 2.661c-.442 1.253-1.845 1.602-2.932
1.25C11.309 13.488 9.475 13 8 13c-1.474 0-3.31.488-4.615.911-1.087.352-2.49.003-2.932-
```

```
1.25A7.988 7.988 0 0 1 0 10zm8-7a7 7 0 0 0-6.603 9.329c.203.575.923.876 1.68.63C4.397 12.533
6.358 12 8 12s3.604.532 4.923.96c.757.245 1.477-.056 1.68-.631A7 7 0 0 0 8 3z"/>
    </symbol>
    <symbol id="table" viewBox="0 0 16 16">
     <path d="M0 2a2 2 0 0 1 2-2h12a2 2 0 0 1 2 2v12a2 2 0 0 1-2 2H2a2 2 0 0 1-2-2V2zm15 2h-
4v3h4V4zm0 4h-4v3h4V8zm0 4h-4v3h3a1 1 0 0 0 1-1v-2zm-5 3v-3H6v3h4zm-5 0v-3H1v2a1 1 0 0 0 1
1h3zm-4-4h4V8H1v3zm0-4h4V4H1v3zm5-3v3h4V4H6zm4 4H6v3h4V8z"/>
    </symbol>
    <symbol id="people-circle" viewBox="0 0 16 16">
     <path d="M11 6a3 3 0 1 1-6 0 3 3 0 0 1 6 0z"/>
     <path fill-rule="evenodd" d="M0 8a8 8 0 1 1 16 0A8 8 0 0 1 0 8zm8-7a7 7 0 0 0-5.468
11.37C3.242 11.226 4.805 10 8 10s4.757 1.225 5.468 2.37A7 7 0 0 0 8 1z"/>
    </symbol>
    <symbol id="grid" viewBox="0 0 16 16">
     <path d="M1 2.5A1.5 1.5 0 0 1 2.5 1h3A1.5 1.5 0 0 1 7 2.5v3A1.5 1.5 0 0 1 5.5 7h-3A1.5 1.5 0 0
1 1 5.5v-3zM2.5 2a.5.5 0 0 0-.5.5v3a.5.5 0 0 0 .5.5h3a.5.5 0 0 0 .5-.5v-3a.5.5 0 0 0-.5-.5h-
3zm6.5.5A1.5 1.5 0 0 1 10.5 1h3A1.5 1.5 0 0 1 15 2.5v3A1.5 1.5 0 0 1 13.5 7h-3A1.5 1.5 0 0 1 9 5.5v-
3zm1.5-.5a.5.5 0 0 0-.5.5v3a.5.5 0 0 0 .5.5h3a.5.5 0 0 0 .5-.5v-3a.5.5 0 0 0-.5-.5h-3zM1 10.5A1.5 1.5 0
0 1 2.5 9h3A1.5 1.5 0 0 1 7 10.5v3A1.5 1.5 0 0 1 5.5 15h-3A1.5 1.5 0 0 1 1 13.5v-3zm1.5-.5a.5.5 0 0 0-
.5.5v3a.5.5 0 0 0 .5.5h3a.5.5 0 0 0 .5-.5v-3a.5.5 0 0 0-.5-.5h-3zm6.5.5A1.5 1.5 0 0 1 10.5 9h3a1.5 1.5
0 0 1 1.5 1.5v3a1.5 1.5 0 0 1-1.5 1.5h-3A1.5 1.5 0 0 1 9 13.5v-3zm1.5-.5a.5.5 0 0 0-.5.5v3a.5.5 0 0 0
.5.5h3a.5.5 0 0 0 .5-.5v-3a.5.5 0 0 0-.5-.5h-3z"/>
    </symbol>
    <symbol id="collection" viewBox="0 0 16 16">
     <path d="M2.5 3.5a.5.5 0 0 1 0-1h11a.5.5 0 0 1 0 1h-11zm2-2a.5.5 0 0 1 0-1h7a.5.5 0 0 1 0 1h-
7zM0 13a1.5 1.5 0 0 0 1.5 1.5h13A1.5 1.5 0 0 0 16 13V6a1.5 1.5 0 0 0-1.5-1.5h-13A1.5 1.5 0 0 0 0
6v7zm1.5.5A.5.5 0 0 1 1 13V6a.5.5 0 0 1 .5-.5h13a.5.5 0 0 1 .5.5v7a.5.5 0 0 1-.5.5h-13z"/>
    </symbol>
    <symbol id="calendar3" viewBox="0 0 16 16">
     <path d="M14 0H2a2 2 0 0 0-2 2v12a2 2 0 0 0 2 2h12a2 2 0 0 0 2-2V2a2 2 0 0 0-2-2zM1
3.857C1 3.384 1.448 3 2 3h12c.552 0 1 .384 1 .857v10.286c0 .473-.448.857-1 .857H2c-.552 0-1-.384-
1-.857V3.857z"/>
     <path d="M6.5 7a1 1 0 1 0 0-2 1 1 0 0 0 0 2zm3 0a1 1 0 1 0 0-2 1 1 0 0 0 0 2zm3 0a1 1 0 1 0 0-2
1 1 0 0 0 0 2zm-9 3a1 1 0 1 0 0-2 1 1 0 0 0 0 2zm3 0a1 1 0 1 0 0-2 1 1 0 0 0 0 2zm3 0a1 1 0 1 0 0-2 1 1
0 0 0 0 2zm3 0a1 1 0 1 0 0-2 1 1 0 0 0 0 2zm-9 3a1 1 0 1 0 0-2 1 1 0 0 0 0 2zm3 0a1 1 0 1 0 0-2 1 1 0 0
0 0 2zm3 0a1 1 0 1 0 0-2 1 1 0 0 0 0 2z"/>
    </symbol>
    <symbol id="chat-quote-fill" viewBox="0 0 16 16">
     <path d="M16 8c0 3.866-3.582 7-8 7a9.06 9.06 0 0 1-2.347-.306c-.584.296-1.925.864-4.181
1.234-.2.032-.352-.176-.273-.362.354-.836.674-1.95.77-2.966C.744 11.37 0 9.76 0 8c0-3.866 3.582-7
8-7s8 3.134 8 7zM7.194 6.766a1.688 1.688 0 0 0-.227-.272 1.467 1.467 0 0 0-.469-.324l-.008-
.004A1.785 1.785 0 0 0 5.734 6C4.776 6 4 6.746 4 7.667c0 .92.776 1.666 1.734 1.666.343 0 .662-
.095.931-.26-.137.389-.39.804-.81 1.22a.405.405 0 0 0 .011.59c.173.16.447.155.614-.01 1.334-1.329
1.37-2.758.941-3.706a2.461 2.461 0 0 0-.227-.4zM11 9.073c-.136.389-.39.804-.81 1.22a.405.405 0 0
0 .012.59c.172.16.446.155.613-.01 1.334-1.329 1.37-2.758.942-3.706a2.466 2.466 0 0 0-.228-.4
```

1.686 1.686 0 0 0-.227-.273 1.466 1.466 0 0 0-.469-.324l-.008-.004A1.785 1.785 0 0 0 10.07 6c-.957 0-1.734.746-1.734 1.667 0 .92.777 1.666 1.734 1.666.343 0 .662-.095.931-.26z"/>
    </symbol>
    <symbol id="cpu-fill" viewBox="0 0 16 16">
     <path d="M6.5 6a.5.5 0 0 0-.5.5v3a.5.5 0 0 0 .5.5h3a.5.5 0 0 0 .5-.5v-3a.5.5 0 0 0-.5-.5h-3z"/>
     <path d="M5.5.5a.5.5 0 0 0-1 0V2A2.5 2.5 0 0 0 2 4.5H.5a.5.5 0 0 0 0 1H2v1H.5a.5.5 0 0 0 0 1H2v1H.5a.5.5 0 0 0 0 1H2v1H.5a.5.5 0 0 0 0 1H2A2.5 2.5 0 0 0 4.5 14v1.5a.5.5 0 0 0 1 0V14h1v1.5a.5.5 0 0 0 1 0V14h1v1.5a.5.5 0 0 0 1 0V14h1v1.5a.5.5 0 0 0 1 0V14a2.5 2.5 0 0 0 2.5-2.5h1.5a.5.5 0 0 0 0-1H14v-1h1.5a.5.5 0 0 0 0-1H14v-1h1.5a.5.5 0 0 0 0-1H14v-1h1.5a.5.5 0 0 0 0-1H14A2.5 2.5 0 0 0 11.5 2V.5a.5.5 0 0 0-1 0V2h-1V.5a.5.5 0 0 0-1 0V2h-1V.5a.5.5 0 0 0-1 0V2h-1V.5zm1 4.5h3A1.5 1.5 0 0 1 11 6.5v3A1.5 1.5 0 0 1 9.5 11h-3A1.5 1.5 0 0 1 5 9.5v-3A1.5 1.5 0 0 1 6.5 5z"/>
    </symbol>
    <symbol id="gear-fill" viewBox="0 0 16 16">
     <path d="M9.405 1.05c-.413-1.4-2.397-1.4-2.81 0l-.1.34a1.464 1.464 0 0 1-2.105.872l-.31-.17c-1.283-.698-2.686.705-1.987 1.987l.169.311c.446.82.023 1.841-.872 2.105l-.34.1c-1.4.413-1.4 2.397 0 2.81l.34.1a1.464 1.464 0 0 1 .872 2.105l-.17.31c-.698 1.283.705 2.686 1.987 1.987l.311-.169a1.464 1.464 0 0 1 2.105.872l.1.34c.413 1.4 2.397 1.4 2.81 0l.1-.34a1.464 1.464 0 0 1 2.105-.872l.31.17c1.283.698 2.686-.705 1.987-1.987l-.169-.311a1.464 1.464 0 0 1 .872-2.105l.34-.1c1.4-.413 1.4-2.397 0-2.81l-.34-.1a1.464 1.464 0 0 1-.872-2.105l.17-.31c.698-1.283-.705-2.686-1.987-1.987l-.311.169a1.464 1.464 0 0 1-2.105-.872l-.1-.34zM8 10.93a2.929 2.929 0 1 1 0-5.86 2.929 2.929 0 0 1 0 5.858z"/>
    </symbol>
    <symbol id="speedometer" viewBox="0 0 16 16">
     <path d="M8 2a.5.5 0 0 1 .5.5V4a.5.5 0 0 1-1 0V2.5A.5.5 0 0 1 8 2zM3.732 3.732a.5.5 0 0 1 .707 0l.915.914a.5.5 0 1 1-.708.708l-.914-.915a.5.5 0 0 1 0-.707zM2 8a.5.5 0 0 1 .5-.5h1.586a.5.5 0 0 1 0 1H2.5A.5.5 0 0 1 2 8zm9.5 0a.5.5 0 0 1 .5-.5h1.5a.5.5 0 0 1 0 1H12a.5.5 0 0 1-.5-.5zm.754-4.246a.389.389 0 0 0-.527-.02L7.547 7.31A.91.91 0 1 0 8.85 8.569l3.434-4.297a.389.389 0 0 0-.029-.518z"/>
     <path fill-rule="evenodd" d="M6.664 15.889A8 8 0 1 1 9.336.11a8 8 0 0 1-2.672 15.78zm-4.665-4.283A11.945 11.945 0 0 1 8 10c2.186 0 4.236.585 6.001 1.606a7 7 0 1 0-12.002 0z"/>
    </symbol>
    <symbol id="toggles2" viewBox="0 0 16 16">
     <path d="M9.465 10H12a2 2 0 1 1 0 4H9.465c.34-.588.535-1.271.535-2 0-.729-.195-1.412-.535-2z"/>
     <path d="M6 15a3 3 0 1 0 0-6 3 3 0 0 0 0 6zm0 1a4 4 0 1 1 0-8 4 4 0 0 1 0 8zm.535-10a3.975 3.975 0 0 1-.409-1H4a1 1 0 0 1 0-2h2.126c.091-.355.23-.69.41-1H4a2 2 0 1 0 0 4h2.535z"/>
     <path d="M14 4a4 4 0 1 1-8 0 4 4 0 0 1 8 0z"/>
    </symbol>
    <symbol id="tools" viewBox="0 0 16 16">
     <path d="M1 0L0 1l2.2 3.081a1 1 0 0 0 .815.419h.07a1 1 0 0 1 .708.293l2.675 2.675-2.617 2.654A3.003 3.003 0 0 0 0 13a3 3 0 1 0 5.878-.851l2.654-2.617.968.968-.305.914a1 1 0 0 0 .242 1.023l3.356 3.356a1 1 0 0 0 1.414 0l1.586-1.586a1 1 0 0 0 0-1.414l-3.356-3.356a1 1 0 0 0-1.023-.242L10.5 9.5l-.96-.96 2.68-2.643A3.005 3.005 0 0 0 16 3c0-.269-.035-.53-.102-.777l-2.14 2.141L12 4l-.364-1.757L13.777.102a3 3 0 0 0-3.675 3.68L7.462 6.46 4.793 3.793a1 1 0 0 1-.293-.707v-.071a1 1

```
0 0 0-.419-.814L1 0zm9.646 10.646a.5.5 0 0 1 .708 0l3 3a.5.5 0 0 1-.708.708l-3-3a.5.5 0 0 1 0-
.708zM3 11l.471.242.529.026.287.445.445.287.026.529L5 13l-.242.471-.026.529-.445.287-.287.445-
.529.026L3 15l-.471-.242L2 14.732l-.287-.445L1.268 14l-.026-.529L1 13l.242-.471.026-.529.445-
.287.287-.445.529-.026L3 11z"/>
    </symbol>
    <symbol id="chevron-right" viewBox="0 0 16 16">
      <path fill-rule="evenodd" d="M4.646 1.646a.5.5 0 0 1 .708 0l6 6a.5.5 0 0 1 0 .708l-6 6a.5.5 0 0
1-.708-.708L10.293 8 4.646 2.354a.5.5 0 0 1 0-.708z"/>
    </symbol>
    <symbol id="geo-fill" viewBox="0 0 16 16">
      <path fill-rule="evenodd" d="M4 4a4 4 0 1 1 4.5 3.969V13.5a.5.5 0 0 1-1 0V7.97A4 4 0 0 1 4
3.999zm2.493 8.574a.5.5 0 0 1-.411.575c-.712.118-1.28.295-1.655.493a1.319 1.319 0 0 0-
.37.265.301.301 0 0 0-.057.09V14l.002.008a.147.147 0 0 0 .016.033.617.617 0 0 0
.145.15c.165.13.435.27.813.395.751.25 1.82.414 3.024.414s2.273-.163 3.024-.414c.378-.126.648-
.265.813-.395a.619.619 0 0 0 .146-.15.148.148 0 0 0 .015-.033L12 14v-.004a.301.301 0 0 0-.057-.09
1.318 1.318 0 0 0-.37-.264c-.376-.198-.943-.375-1.655-.493a.5.5 0 1 1 .164-.986c.77.127 1.452.328
1.957.594C12.5 13 13 13.4 13 14c0 .426-.26.752-.544.977-.29.228-.68.413-1.116.558-.878.293-
2.059.465-3.34.465-1.281 0-2.462-.172-3.34-.465-.436-.145-.826-.33-1.116-.558C3.26 14.752 3
14.426 3 14c0-.599.5-1 .961-1.243.505-.266 1.187-.467 1.957-.594a.5.5 0 0 1 .575.411z"/>
    </symbol>
  </svg>
  <nav class="navbar navbar-dark bg-dark">
    <div class="container-fluid">
     <a class="navbar-brand h1 p-3" >Seller Dashboard</a>
     <a href="{%url 'seller_logout'%}" class="d-flex ">Logout</a>
    </div>
   </nav>
<main>
<div class="d-flex flex-column  flex-shrink-0 p-3 text-white bg-dark" style="width: 280px; height:
90.5vh;">
  <ul class="nav nav-pills flex-column mb-auto">
   <li class="nav-item">
    <a href="{%url 'seller_dashboard'%}" class="nav-link text-white {% if request.path ==
'/seller/dashboard/' %}active {% endif %}" aria-current="page">
     <svg class="bi me-2" width="16" height="16"><use xlink:href="#home"></use></svg>
     Dashboard
    </a>
   </li>
   <li>
    <a href="{% url 'seller_profile'%}" class="nav-link text-white {% if request.path ==
'/seller/profile/' %}active {% endif %}">
     <svg class="bi me-2" width="16" height="16"><use xlink:href="#people-circle"></use></svg>
     Profile
    </a>
   </li>
```

```html
    <li>
      <a href="{%url 'seller_product'%}" class="nav-link text-white {% if request.path ==
'/seller/product/' %}active {% endif %}">
        <svg class="bi me-2" width="16" height="16"><use xlink:href="#grid"></use></svg>
        Add Product
      </a>
    </li>
    <li>
      <a href="{%url 'seller_product_list'%}" class="nav-link text-white {% if request.path ==
'/seller/product/list/' %}active {% endif %}">
        <svg class="bi me-2" width="16" height="16"><use xlink:href="#collection"></use></svg>
        View Product
      </a>
    </li>
    <li>
      <a href="{%url 'seller_order'%}" class="nav-link text-white {% if request.path == '/seller/orders/'
%}active {% endif %}">
        <svg class="bi me-2" width="16" height="16"><use xlink:href="#table"></use></svg>
        Orders
      </a>
    </li>
    {% comment %} <li>
      <a href="#" class="nav-link text-white">
        <svg class="bi me-2" width="16" height="16"><use xlink:href="#grid"></use></svg>
        Products
      </a>
    </li>
    <li>
      <a href="#" class="nav-link text-white">
        <svg class="bi me-2" width="16" height="16"><use xlink:href="#people-circle"></use></svg>
        Customers
      </a>
    </li> {% endcomment %}
  </ul>
</div>
<div class="container m-3">
  {% block content %}
  {% endblock %}
</div>
</main>
<!--bootstrap-->
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"
integrity="sha384-ka7Sk0Gln4gmtz2MlQnikT1wXgYsOg+OMhuP+IlRH9sENBO0LRn5q+8nbTov4+1p"
crossorigin="anonymous"></script>
<!-- owl-carousel -->
```
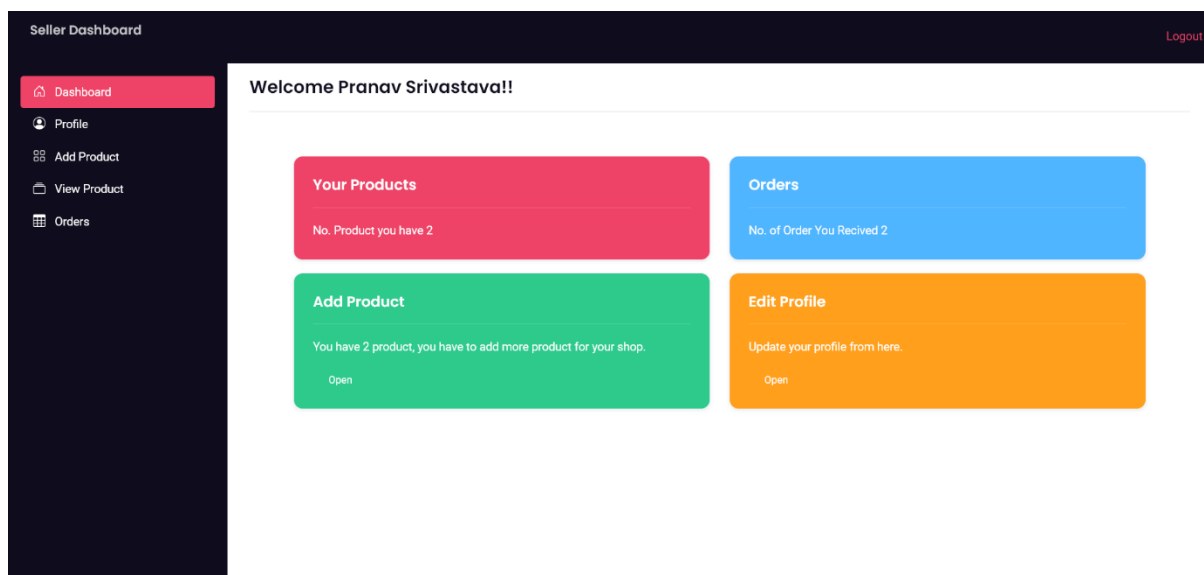
```html
<script src="https://getbootstrap.com/docs/5.1/examples/sidebars/sidebars.js"></script>
<script src="{%static '/vendor/owl-carousel/js/owl.carousel.min.js'%}"></script>
<!-- magnific -->
<script src="{%static '/vendor/magnific/jquery.magnific-popup.min.js'%}"></script>
<!-- working form -->
<script src="{%static '/vendor/mail/js/form.min.js'%}"></script>
<script src="{%static '/vendor/mail/js/script.js'%}"></script>
<!-- Theme Js -->
<script src="{%static '/js/custom.js'%}"></script>
<script src="{%static '/js/voice.js'%}"></script>
</body>
<!-- end body -->
</html>
```

## Dashboard.html



```
{% extends "seller/sellerbase.html" %}
{%load static%}
{% block title %}Seller Dashboard | SpeakAndShop{% endblock %}
{% block content %}
{% for dt in data %}
   <h4>Welcome {{dt.first_name}}{{" "}} {{dt.last_name}}!!</h4>
   {% endfor %}
<hr>
<main class="container my-4">
  <div class="row m-5">
     <div class="col-sm-6">
      <div class="card text-white bg-primary mb-3">
       <div class="card-body">
         <h5 class="card-title">Your Products</h5>
```

```
          <hr>
          <p class="card-text">No. Product you have {{pd}}</p>
        </div>
      </div>
    </div>
    <div class="col-sm-6">
      <div class="card text-white bg-info mb-3">
        <div class="card-body">
          <h5 class="card-title">Orders</h5>
          <hr>
          <p class="card-text">No. of Order You Recived {{ct}}</p>
        </div>
      </div>
    </div>
    <div class="col-sm-6">
      <div class="card text-white bg-success mb-3">
        <div class="card-body">
          <h5 class="card-title">Add Product</h5>
          <hr>
          <p class="card-text">You have {{pd}} product, you have to add more product for your
shop.</p>
          <a href="{% url 'seller_product'%}" class="btn-sm text-white">Open</a>
        </div>
      </div>
    </div>
    <div class="col-sm-6">
      <div class="card text-white bg-warning mb-3">
        <div class="card-body">
          <h5 class="card-title">Edit Profile</h5>
          <hr>
          <p class="card-text">Update your profile from here. </p>
          <a href="{% url 'seller_profile_edit'%}" class="btn-sm text-white">Open</a>
        </div>
      </div>
    </div>
  </div>
</main>
{% endblock %}
```

## Product.py(model)

```
from django.db import models
from E_shop.utils import unique_slug_generator
from .category import Category
from django.utils.html import mark_safe
```

```python
from django.urls import reverse
from django.utils.text import slugify
import itertools
from django.db.models.signals import pre_save, post_save

class Product(models.Model):
    name=models.CharField(max_length=50)
    slugs=models.CharField(null=True,blank=True,max_length=250)
    price=models.IntegerField(default=0)
    category=models.ForeignKey(Category,on_delete=models.CASCADE,default=1)
    description=models.CharField(default='',null=True,blank=True,max_length=250)
    image=models.ImageField(upload_to='uploads/products/')
    sid=models.IntegerField(default=1)
    class Meta:
        db_table="store_product"
        verbose_name_plural='2. Products'

    def __str__(self):
        return self.name

    def image_tags(self):
        return mark_safe('<img src="%s" width="50" height="50"/>' % (self.image.url))

    @staticmethod
    def get_all_product():
        return Product.objects.all()

    @staticmethod
    def get_all_product():
        return Product.objects.all()

    @staticmethod
    def get_limited_product():
        return Product.objects.all()[:12:2]

    @staticmethod
    def get_all_product_list():
        return Product.objects.all().values()

    @staticmethod
    def get_latest_product():
        return Product.objects.order_by('-id')[:3]

    @staticmethod
    def get_seller_product(sid):
```

```python
        return Product.objects.filter(sid=sid).order_by('-id')

    @staticmethod
    def search_product(q):
        return Product.objects.filter(name__icontains=q)

    @staticmethod
    def get_all_product_by_categoryid(category_id):
        if category_id:
            return Product.objects.filter(category=category_id)
        else:
            return Product.objects.all()

    @staticmethod
    def get_all_product_by_categoryid_onfilter(category_id):
        if category_id:
            return Product.objects.filter(category=category_id).values()
        else:
            return Product.objects.all()

    @staticmethod
    def get_product_detail(i):
        return Product.objects.filter(id=i)

def rl_pre_save_receiver(sender, instance, *args, **kwargs):
    if not instance.slugs:
        instance.slugs = unique_slug_generator(instance)

pre_save.connect(rl_pre_save_receiver, sender=Product)
```

## Category.py(model)

```python
from tabnanny import verbose
from django.db import models

class Category(models.Model):
    name=models.CharField(max_length=50)

    class Meta:
        db_table="store_category"
        verbose_name_plural='1. Categories'
    def __str__(self):
        return self.name
    @staticmethod
    def get_all_category():
```

```
return Category.objects.all()
```

## Profile.html

```
{% extends "seller/sellerbase.html" %}
{%load static%}
```
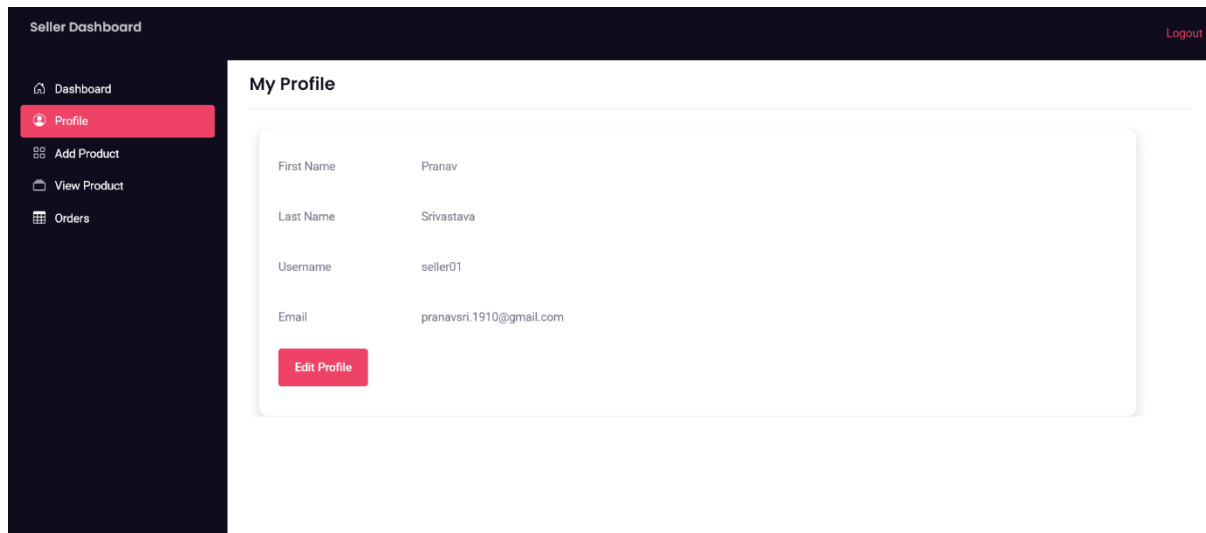


```
{% block title %}Seller Profile | SpeakAndShop{% endblock %}
{% block content %}
<h4>My Profile</h4>
<hr>
{% for dt in data %}
<main class="container my-4">
   <div class="card shadow" style="width: 70rem; height:23rem">
     <div class="card-body">
       <div class="mb-3 row">
         <label for="staticEmail" class="col-sm-2 col-form-label">First Name</label>
         <div class="col-md-10">
          <input type="text" readonly class="form-control-plaintext" id="staticEmail"
value="{{dt.first_name}}">
         </div>
       </div>
       <div class="mb-3 row">
         <label for="staticEmail" class="col-sm-2 col-form-label">Last Name</label>
         <div class="col-md-10">
          <input type="text" readonly class="form-control-plaintext" id="staticEmail"
value="{{dt.last_name}}">
         </div>
       </div>
       <div class="mb-3 row">
         <label for="staticEmail" class="col-sm-2 col-form-label">Username</label>
         <div class="col-md-10">
```

```
            <input type="text" readonly class="form-control-plaintext" id="staticEmail"
value="{{dt.username}}">
        </div>
    </div>
    <div class="mb-3 row">
        <label for="staticEmail" class="col-sm-2 col-form-label">Email</label>
        <div class="col-md-10">
            <input type="text" readonly class="form-control-plaintext" id="staticEmail"
value="{{dt.email}}">
        </div>
    </div>
    <a href="{%url 'seller_profile_edit'%}" class="btn btn-primary">Edit Profile</a>
    </div>
    </div>
</main>
{% endfor %}
{% endblock %}
```
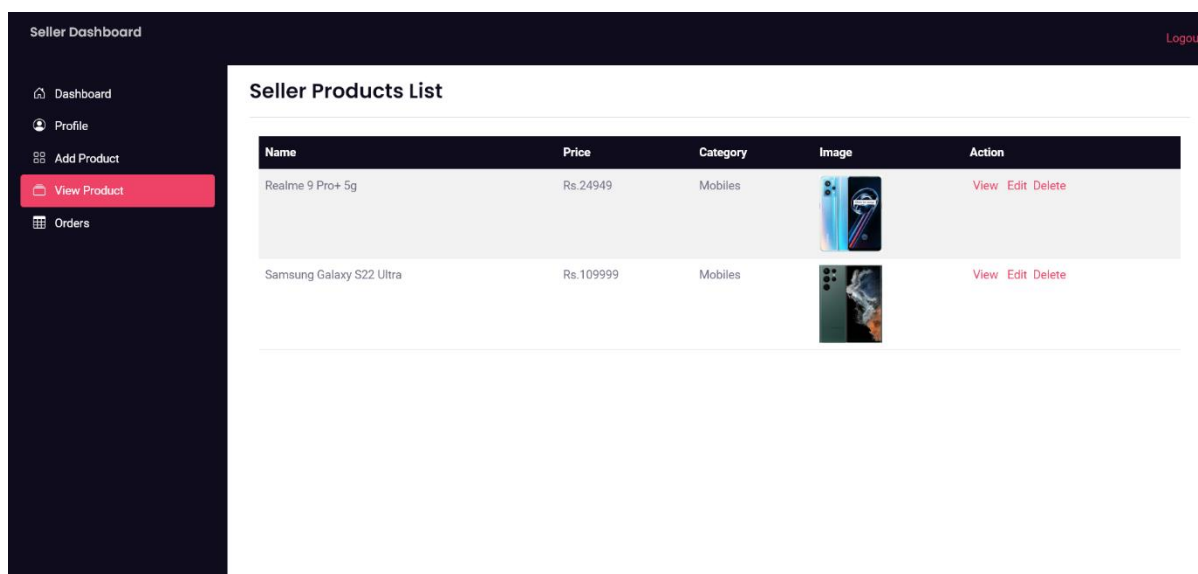
## Product_list.html

```
{% extends "seller/sellerbase.html" %}
{%load static%}
```



```
{% block title %}Seller Product List| SpeakAndShop{% endblock %}
{% block content %}
<h3>Seller Products List</h3>
<hr>
<main class="container-fluid my-4">
 <table class="table table-striped">
    <thead class="table-dark">
     <tr>
```

```html
        <th scope="col">Name</th>
        <th scope="col">Price</th>
        <th scope="col">Category</th>
        <th scope="col">Image</th>
        <th scope="col">Action</th>
      </tr>
    </thead>
    <tbody>
      {%for product in products%}
      <tr>
      <td>{{product.name}}</td>
      <td>Rs.{{product.price}}</td>
      <td>{{product.category}}</td>
      <td><img src="/images/{{product.image}}" width="80" /></td>
      <td><a href="/product/{{product.slugs}}" class="mx-1 ">View</a>
      <a href="/seller/product/edit/{{product.id}}" class="mx-1 ">Edit</a>
      <a class="btn-del" data-pid="{{product.id}}" class="mx-1 ">Delete</a></td>
      </tr>
      {% endfor %}
    </tbody>
  </table>

</main>
<script>
 $("tbody").on("click",".btn-del",function(){
   let id=$(this).attr("data-pid");
   let csr=$("input[name=csrfmiddlewaretoken]").val();
   console.log(id);
   console.log(csr);

   mydata={pid:id};
   mythis=this;
   $.ajax({
     url:"{% url 'delete_product' %}",
     method:"POST",
     data:mydata,
     success:function(data){
       if(data.status==1){
          $(mythis).closest("tr").fadeOut();
           console.log("delete")
       }
     }
   })
});
```

```
</script>
{% endblock %}
```

## Orders.html

```
{% extends "seller/sellerbase.html" %}
{%load static%}
```
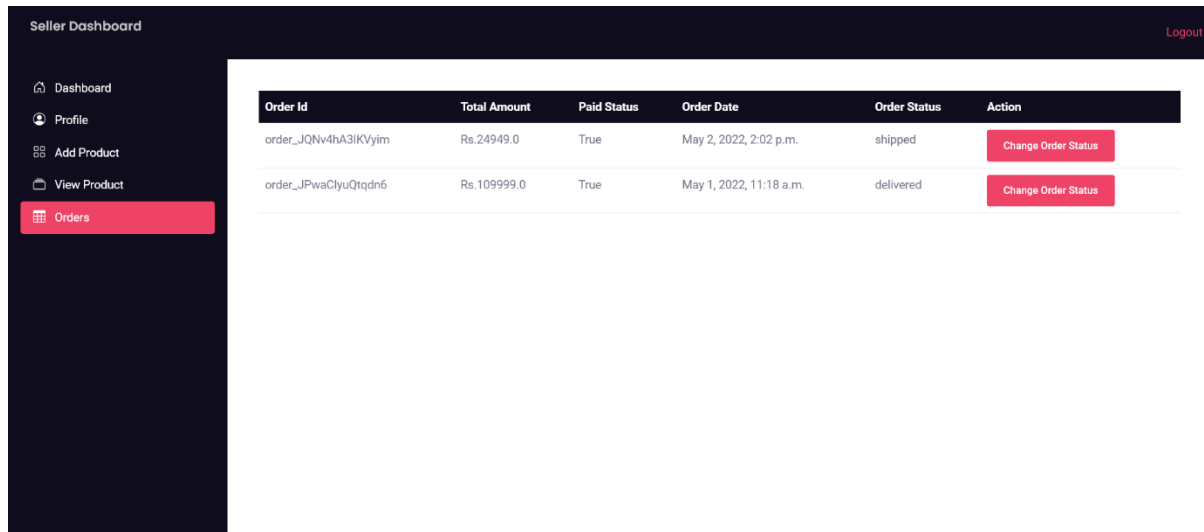


```
{% block title %}Seller Orders | SpeakAndShop{% endblock %}
{% block content %}
<main class="container my-4">
   <table class="table">
     <thead class="table-dark">
      <tr>
       <th scope="col">Order Id</th>
       <th scope="col">Total Amount</th>
       <th scope="col">Paid Status</th>
       <th scope="col">Order Date</th>
       <th scope="col">Order Status</th>
       <th scope="col">Action</th>
      </tr>
     </thead>
     <tbody>
       {%for od in orders%}
      <tr>
       <td>{{od.order_id}}</td>
       <td>Rs.{{od.total_amt}}</td>
       <td>{{od.paid_status}}</td>
       <td>{{od.order_dt}}</td>
       <td>{{od.order_status}}</td>
       <td><a type="button" id="bt1" class="btn btn-primary btn-sm"
href="/seller/order/{{od.order_id}}"  data-id="{{od.id}}">
```

```
        Change Order Status
      </a></td>
     </tr>
    {% endfor %}
   </tbody>
  </table>
</main>
{% endblock %}
<script>
  $(document).ready(function(){

  });
</script>
```

## Orders.py

```python
from django.db import models
from django.contrib.auth.models import User
from django.utils.html import mark_safe
from django.urls import reverse
from django.conf.urls import url

status_choice=(
    ('process','In Process'),
    ('shipped','Shipped'),
    ('delivered','Delivered'),
  )
class CartOrder(models.Model):
  user=models.ForeignKey(User,on_delete=models.CASCADE)
  total_amt=models.FloatField()
  order_id=models.CharField(max_length=250,default='')
  paid_status=models.BooleanField(default=False)
  order_dt=models.DateTimeField(auto_now_add=True)
  order_status=models.CharField(choices=status_choice,default='process',max_length=150)
  sid=models.IntegerField(default=1)

  class Meta:
    verbose_name_plural='3. Orders'

  @staticmethod
  def get_seller_order(sid):
    return CartOrder.objects.filter(sid=sid).order_by('-id').values()

# OrderItems
class CartOrderItems(models.Model):
```

```python
    order=models.ForeignKey(CartOrder,on_delete=models.CASCADE)
    invoice_no=models.CharField(max_length=150)
    item=models.CharField(max_length=150)
    image=models.CharField(max_length=200)
    address=models.CharField(max_length=200,null=True)
    mobile=models.IntegerField(null=True)
    qty=models.IntegerField()
    price=models.FloatField()
    total=models.FloatField()

    class Meta:
        verbose_name_plural='4. Order Items'

    def image_tags(self):
        return mark_safe('<img src="/images/%s" width="50" height="50"/>' % (self.image))

class UserAddressBook(models.Model):
    user=models.ForeignKey(User,on_delete=models.CASCADE)
    mobile=models.CharField(max_length=50,null=True)
    address=models.TextField()
    status=models.BooleanField(default=False)

    class Meta:
        verbose_name_plural='5. AddressBook'
```

## Change_status.html

```html
{% extends "seller/sellerbase.html" %}
{%load static%}
{% block title %}Seller Dashboard | SpeakAndShop{% endblock %}
{% block content %}
<main class="container my-4">
 <script type="text/javascript">
  var selectFunction = function() {
     document.getElementById("inputGroupSelect01").value = "{{status}}";
  };
  window.onload = function() {
     selectFunction();
   };
</script>
<div class="card shadow m-5" style="width: 70rem; height:14rem">
    <div class="card-body">
     <form method="POST">
       {% csrf_token %}
       <h5 class="card-title mt-2">Change Status : {{id}} </h5>
```
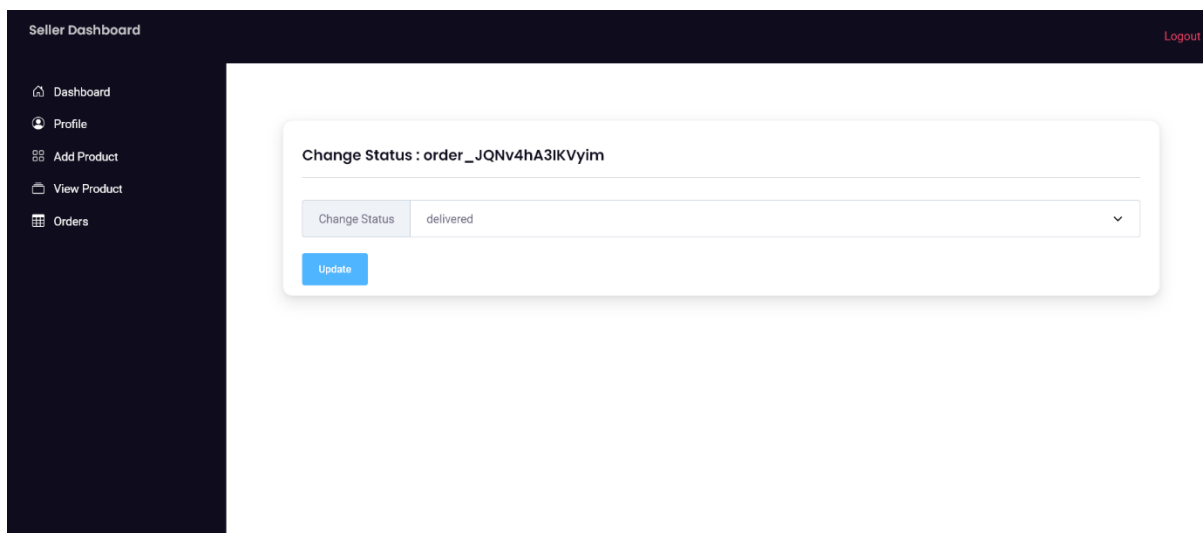
```
<hr>
<h6 class="card-subtitle mb-5 text-muted"></h6>
<div class="input-group mb-3">
  <label class="input-group-text" for="inputGroupSelect01" name="status">Change
Status</label>
  <select class="form-select" name="status" id="inputGroupSelect01">

    <option value="process">process</option>
    <option value="shipped">shipped</option>
    <option value="delivered">delivered</option>
  </select>
</div>
<input type="submit" value="Update" class="btn btn-sm btn-info mt-1"/>
</form>
```



```
      </div>
    </div>
</main>
{% endblock %}
```

# *ADMIN PART*

## Setting.py

```python
from pathlib import Path
# Build paths inside the project like this: BASE_DIR / 'subdir'.
BASE_DIR = Path(__file__).resolve().parent.parent


# Quick-start development settings - unsuitable for production
# See https://docs.djangoproject.com/en/3.2/howto/deployment/checklist/


# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = 'django-insecure-v@+*0)1azr+%*&(j1d+l(&tsh%2eop&&9=5x810syamk@4adxr'


# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True


ALLOWED_HOSTS = []
#APPEND_SLASH=False


# Application definition


INSTALLED_APPS = [
    'store',
    'jet.dashboard',
    'jet',
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
]


MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
]
```

```python
ROOT_URLCONF = 'E_shop.urls'

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    },
]

WSGI_APPLICATION = 'E_shop.wsgi.application'

# Database
# https://docs.djangoproject.com/en/3.2/ref/settings/#databases

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'e_shop',
        'USER': 'root',
        'PASSWORD': '',
        'HOST': '127.0.0.1',
        'PORT': '3306',
    }
}

# Password validation
# https://docs.djangoproject.com/en/3.2/ref/settings/#auth-password-validators

AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
    },
    {
```

```
        'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator',
    },
]


# Internationalization
# https://docs.djangoproject.com/en/3.2/topics/i18n/

LANGUAGE_CODE = 'en-us'
TIME_ZONE = 'UTC'
USE_I18N = True
USE_L10N = True
USE_TZ = True

# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/3.2/howto/static-files/

STATIC_URL = '/static/'
MEDIA_URL="/images/"
MEDIA_ROOT=BASE_DIR

# Default primary key field type
# https://docs.djangoproject.com/en/3.2/ref/settings/#default-auto-field

DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'

LOGIN_REDIRECT_URL='home'
LOGOUT_REDIRECT_URL='login'

RAZOR_KEY_ID = 'rzp_test_SFxoDXXyHgmCJj'
RAZOR_KEY_SECRET = 'SYdzJ7wr55dQ2uHZ2y3pIJiF'

EMAIL_BACKEND = 'django.core.mail.backends.smtp.EmailBackend'
EMAIL_USE_TLS = True
EMAIL_HOST = 'smtp.gmail.com'
EMAIL_HOST_USER = 'speakandshop@gmail.com'
EMAIL_HOST_PASSWORD = 'Sas@2022'
EMAIL_PORT = 587
```

## Urls.py

```
from django.contrib import admin
from django.urls import path,include
```

```
from . import settings
from django.conf.urls.static import static
admin.site.site_header="Speak and Shop Admin Pannel"
admin.site.site_title="SAS Dashboard"
admin.site.index_title="Welcome to Admin Pannel"
urlpatterns = [
    path('jet/',include('jet.urls')),
    path('jet/dashboard/', include('jet.dashboard.urls', 'jet-dashboard')),
    path('admin/', admin.site.urls),
    path('',include('store.urls')),
    #path('accounts/',include('django.contrib.auth.urls'),name='login'),
    #path('accounts/password_change',include('django.contrib.auth.urls'),name='password_change'),
]+ static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

## Admin.py

```
from django.contrib import admin
from .models.product import Product
from .models.category import Category
from .models.order import *

# Register your models here.

class AdminProduct(admin.ModelAdmin):
    list_display=['id','name','slugs','price','description','image_tags','category']

class CategoryProduct(admin.ModelAdmin):
    list_display=['name']

admin.site.register(Product, AdminProduct)
admin.site.register(Category,CategoryProduct)

class CartOrderAdmin(admin.ModelAdmin):
    list_editable=('paid_status','order_status')
    list_display=('user','total_amt','order_id','paid_status','order_status','order_dt')
admin.site.register(CartOrder,CartOrderAdmin)

class CartOrderItemsAdmin(admin.ModelAdmin):
    list_display=('invoice_no','item','image_tags','qty','price','total')
admin.site.register(CartOrderItems,CartOrderItemsAdmin)

class UserAddressBookAdmin(admin.ModelAdmin):
    list_display=('user','address','status')
admin.site.register(UserAddressBook,UserAddressBookAdmin)
```

**Username:**
admin

**Password:**
••••••••••

LOG IN

---

QUICK LINKS

Return to site   Change password   Log out

∧ RECENT ACTIONS

✎ Store | cart order CartOrder object (48)
▣ Store | product Glass
✎ Store | product Red Shirt
✎ Store | product black sirt
✎ Store | product blue shirt
✎ Store | product blue tshirt
✎ Store | product dark green shirt
✎ Store | product b&G top
✎ Store | product black Sweatshiert
✎ Store | product white top

∧ APPLICATIONS

STORE
1. Categories
2. Products
3. Orders
4. Order Items
5. AddressBook

∨ LATEST DJANGO NEWS

∧ ADMINISTRATION

AUTHENTICATION AND AUTHORIZATION
Groups
Users

∧ SUPPORT
Django documentation
Django "django-users" mailing list
Django irc channel

---

≡  HOME  >  STORE  >  1. CATEGORIES                                admin ∨

＋ Add category

☐  NAME
☐  Groceries
☐  Cloths - Women
☐  Cloths - Men

[--------- ∨]  GO

3 1. Categories

≡ HOME > STORE > 2. PRODUCTS

admin ⌄

+ Add product

| ☐ | ID | NAME | SLUGS | PRICE | DESCRIPTION | IMAGE TAGS | CATEGORY |
|---|----|------|-------|-------|-------------|-----------|----------|
| ☐ | 24 | Glass | glass | 400 | - | | Cloths - Men |
| ☐ | 23 | Green Tshirt | green-tshirt | 400 | - | | Cloths - Men |
| ☐ | 22 | Black Tee | black-tee | 400 | zXCvdbf | | Cloths - Men |
| ☐ | 21 | navy blue banarasi sarree | navy-blue-banarasi-sarree | 9000 | - | | Cloths - Women |
| ☐ | 20 | green banarasi sarree | green-banarasi-sarree | 6000 | - | | Cloths - Women |
| ☐ | 19 | red Banarasi sarree | red-banarasi-sarree | 4500 | - | | Cloths - Women |

⌄ GO

24 2. Products

---

≡ HOME > STORE > 3. ORDERS

admin ⌄

+ Add cart order

| ☐ | USER | TOTAL AMT | ORDER ID | PAID STATUS | ORDER STATUS | | ORDER DT |
|---|------|-----------|----------|-------------|--------------|---|----------|
| ☐ | websec | 450.0 | order_JP0qxAEz3INvsU | ☑ | In Process | ⌄ | April 29, 2022, 2:50 a.m. |
| ☐ | websec | 400.0 | order_JO3mhYnnFSooHo | ☑ | In Process | ⌄ | April 26, 2022, 5:02 p.m. |
| ☐ | websec | 400.0 | order_JO3hGW0vd1K81W | ☑ | In Process | ⌄ | April 26, 2022, 4:57 p.m. |
| ☐ | pranavsri | 400.0 | order_JNx13nMF7ZHEgl | ☑ | In Process | ⌄ | April 26, 2022, 10:25 a.m. |
| ☐ | websec | 400.0 | order_JGKMvlDPmInUqe | ☑ | In Process | ⌄ | April 7, 2022, 4:03 a.m. |
| ☐ | tech | 400.0 | order_JG0ZSVMcZjdJIU | ☑ | Shipped | ⌄ | April 6, 2022, 8:41 a.m. |
| ☐ | websec | 500.0 | order_JG0Mcir3Jg0gcg | ☑ | In Process | ⌄ | April 6, 2022, 8:29 a.m. |
| ☐ | websec | 450.0 | order_JFyh7bRJUvJrWY | ☑ | In Process | ⌄ | April 6, 2022, 6:51 a.m. |
| ☐ | websec | 450.0 | order_JFyh5HORwHu0Llb | ☑ | In Process | ⌄ | April 6, 2022, 6:51 a.m. |

⌄ GO

SAVE

53 3. Orders

---

≡ HOME > STORE > 2. PRODUCTS > ADD PRODUCT

admin ⌄

Name:*            Jacket

Slugs:

Price:*            1200

Category:*         Cloths - Men  ⌄ ☑ +

Description:       Peter England Jacket

Image:*           Choose File  jacket.jpg

SAVE    Save and add another    Save and continue editing

admin ⌄

The product "Jacket" was added successfully.

➕ Add product

| | ID | NAME | SLUGS | PRICE | DESCRIPTION | IMAGE TAGS | CATEGORY |
|---|---|---|---|---|---|---|---|
| ☐ | 25 | Jacket | jacket | 1200 | Peter England Jacket | | Cloths - Men |
| ☐ | 24 | Glass | glass | 400 | - | | Cloths - Men |
| ☐ | 23 | Green Tshirt | green-tshirt | 400 | - | | Cloths - Men |
| ☐ | 22 | Black Tee | black-tee | 400 | zXCvdbf | | Cloths - Men |
| ☐ | 21 | navy blue banarasi sarree | navy-blue-banarasi-sarree | 9000 | - | | Cloths - Women |
| ☐ | 20 | green banarasi sarree | green-banarasi-sarree | 6000 | - | | Cloths - Women |

# Chapter 8: Testing

## 8.1 Testing Objective

Testing of Speak and Shop application will helps in the prevention of errors and adds value to the product by ensuring conformity to client requirements.

The objective of testing is to ensure:

- Software reliability
- Software quality
- System Assurance
- Optimum performance and capacity utilization

Setting up the system is a complex process and subject to many market-specific variables. To maintain the integrity of the E Commerce system, testing becomes compulsory.

## 8.2 Testing Design

### 8.2.1 White Box Testing

White Box Testing is software testing technique in which internal structure, design and coding of software are tested to verify flow of input-output and to improve design, usability and security. In white box testing, code is visible to testers so it is also called Clear box testing, Open box testing, Transparent box testing, Code-based testing and Glass box testing.

The term "WhiteBox" was used because of the see-through box concept. The clear box or WhiteBox name symbolizes the ability to see through the software's outer shell (or "box") into its inner workings.

White box testing involves the testing of the software code for the following:

- Internal security holes
- Broken or poorly structured paths in the coding processes
- The flow of specific inputs through the code
- Expected output
- The functionality of conditional loops
- Testing of each statement, object, and function on an individual basis

The testing can be done at system, integration and unit levels of software development. One of the basic goals of whitebox testing is to verify a working flow for an application. It involves testing a series of predefined inputs against expected or desired outputs so that when a specific input does not result in the expected output, you have encountered a bug.

### 8.2.2 Black Box Testing

Black Box Testing is a software testing method in which the functionalities of software applications are tested without having knowledge of internal code structure, implementation details and internal paths. Black Box Testing mainly focuses on input and output of software applications and it is entirely based on software requirements and specifications. It is also known as Behavioral Testing.

Here are the generic steps followed to carry out any type of Black Box Testing.

- Initially, the requirements and specifications of the system are examined.
- Tester chooses valid inputs (positive test scenario) to check whether SUT processes them correctly. Also, some invalid inputs (negative test scenario) are chosen to verify that the SUT is able to detect them.
- Tester determines expected outputs for all those inputs.
- Software tester constructs test cases with the selected inputs.
- The test cases are executed.
- Software tester compares the actual outputs with the expected outputs.
- Defects if any are fixed and re-tested.

Black box testing has its own life cycle called Software Testing Life Cycle (STLC) and it is relative to every stage of Software Development Life Cycle of Software Engineering.

- ➢ **Requirement** – This is the initial stage of SDLC and in this stage, a requirement is gathered. Software testers also take part in this stage.
- ➢ **Test Planning & Analysis** – Testing Types applicable to the project are determined. A Test Plan is created which determines possible project risks and their mitigation.
- ➢ **Design** – In this stage Test cases/scripts are created on the basis of software requirement documents
- ➢ **Test Execution**– In this stage Test Cases prepared are executed. Bugs if any are fixed and re-tested.

## 8.3 Test cases of project

### #1) Homepage

- o Is it going to auto scroll?
- o If yes, at what interval will the image be refreshed?
- o When the user hovers over it, is it still going to scroll to the next one?
- o Can it be clicked on?
- o If yes, is it taking you to the right page and right deal?
- o Is it loading along with the rest of the page or loads last in comparison to the other elements on the page?

o   Can the rest of the content be viewed?
o   Does it render the same way in different browsers and different screen resolutions?

**#2) Search**

Search algorithms are very important for the success of a retail site because we can't always place what the users want to see right in front of their eyes.

Common tests are:

o   Search based on the Product name, brand name, or something more broadly, the category. For example Camera, Canon EOS 700D, electronics, etc.
o   Search Results have to be relevant
o   Different sort options have to be available- based on Brand, Price, and Reviews/ratings etc.
o   How many results to display per page?
o   For multi-page results, are there options to navigate to them

Also, search happens in many places. Please take the search drilling down into multiple levels into consideration when validating this functionality.

**#3) Product Details Page**

Once a user finds a product either through search or by browsing or by clicking on it from the homepage, the user will be taken to the product information page.

Check:

o   Image or images of the product
o   Price of the product
o   Product specifications
o   Reviews
o   Check out options
o   Delivery options
o   Shipping information
o   In-stock/Out of stock
o   Multiple color or variations options
o   Breadcrumb navigation for the categories (highlighted in Red below). If navigation such as that is displayed, make sure every element of it is functional.

**#4) Shopping Cart**

This is the penultimate stage before the user commits to the purchase.

Test the following:

o   Add items to the cart and continue shopping

- o If the user adds the same item to the cart while continuing to shop, the item count in the shopping cart should get incremented
- o All items and their totals should be displayed in the cart
- o Taxes as per location should be applied
- o A user can add more items to the cart- total should reflect the same
- o Update the contents added to the cart- total should reflect that too
- o Remove items from the cart
- o Proceed to checkout
- o Calculate Shipping costs with different shipping options
- o Apply coupons
- o Don't check out, close the site, and come back later. The site should retain the items in the cart

## #5) Payments

Payment methods

Check different payment options

- o If allowing check out as Guest, simply finish the purchase and provide an option to register at the end
- o Returning customers – Login to check out
- o User sign up
- o If storing customer Credit card or any other financial information, perform security testing around this to make sure it is secure.(PCI compliance is a must)
- o If the user is signed up for a long time, make sure the session is timed out or not. Every site has a different threshold. For some, it is 10 minutes. For some, it might be different.
- o Emails/Text confirmation with the order number generated

## #6) Categories/Featured Products/Related or Recommended Products

- o The most popular FAQ I get from E-commerce testers is: Do I have to test every category/every product?
- o The answer is NO.
- o If you are a returning customer you will be shown some recommended products on the home page or in your shopping cart.
- o Featured products also change almost every day.
- o Since these are dynamic elements, the best way to test these parts of the application is to test the algorithm based on which these sections are populated.
- o Check your Data mining/BI systems and check from the backend the queries that populate these sections.

## #7) After-Order Tests

- o Change the Order
- o Cancel the Order
- o Track the Order
- o Returns

## #8) Other Tests

- o Login
- o Contact Us page
- o Customer Service page etc.

# Chapter 9: Implementation & Maintenance

## 9.1 Implementation

Implementing an e-commerce shopping cart gives your business the ability to sell your products online day and night, reach new clients, target your ideal market, establish a strong brand, and build closer relationships with your customers by improving their purchasing experience.

Whether you're setting up an online store for the first time or updating your current platform, platform implementation is one of the most complex aspects of launching an e-commerce site. Without the guidance of a consultant with years of e-commerce experience, programming expertise and deep knowledge, your efforts to set up an e-commerce solution can become plagued by cost overruns, programming errors, and delays resulting in poor sales performance and reduced profits.

### IMPLEMENTATION STRATEGIES:

- Proper testing & readiness of website
- Reliability of technology
- Readiness of ISP to load website & handle traffic
- Security of lines & payment system
- Customer service staff training
- Readiness of fulfillment process with vendor's warehouse
- Performance check of entire system before actual deployment
- Assignment of a Web team to monitor, maintain & upgrade the system

### IMPLEMENTATION MANAGEMENT:

- Project Management
- Estimating resource requirements
- Scheduling tasks & events
- Providing for training and site preparation
- Selecting qualified staff & supervising their work
- Monitoring project's program
- Documenting
- Periodic evaluating
- Contingency planning

### IMPLEMENTATION ISSUES:

- Performance
- Personnel

- Integrating front-end commerce site with back-end fulfillment vendors
- Maintenance & enhancements
- Post-implementation planning

**IMPLEMENTATION STRATEGIES DISASTER/RECOVERY PLAN:**

- Appoint a competent team of representatives (ie - Web designer, IT representatives, and customer service staff)
- Prepare planning
- Draft a disaster/recovery manual

## 9.2 Maintenance

Having an ecommerce website is much like this. You need to keep up the ecommerce maintenance on the site to make sure you have the best online presence possible. Replace products where needed and fix any problems that may occur.

Maintaining your site isn't just about changing the color scheme or marking things as 'out of stock'. It's about creating great content, keeping it safe from hackers, and making sure the marketing team is doing its job.

Ecommerce Website Maintenance Has Benefits:

a. **Reduced cart abandonment:** A customer is on your site and after scrolling through; they add three pairs to their cart. However, a technical glitch causes the basket to empty. Frustrated, the customer leaves the site and chooses a competitor instead. Keeping these glitches at bay, then, helps arrest your customer churn.
b. **Better user experience:** It may be that as you have added more products to your pages, things have become jumbled. So, it's important to maintain a clear website, ensuring navigation is straightforward. It's all about being clear and organized. Maintaining an ecommerce website ensures a smooth shopping journey and user experience from start to finish. Meaning customers will be much likelier to return.

If you don't maintain:

a. More prone to hackers
b. Likely to fail on other updates in the future
c. Considered suspicious by web browsers
d. Decrease site rankings on Google
e. Site traffic will slow down

3 Types of Ecommerce Website Maintenance:

a. **Security maintenance:** Security is the number one reason for website maintenance. Hackers are always searching for ways to try and break into a

company's site. Especially those websites with access to customer details and financial information. Like every ecommerce site.

b. **Marketing maintenance:**   Digital marketing not only drives people to your site, but it also keeps search engines happy. If your site is doing well on Google, then even more people will end up on your site. So, it's important to maintain marketing to make sure things are staying fresh and relevant.

c. **Storefront maintenance:**   Your virtual storefront shows you are selling what the customer wants. Look at this landing screen for the confectionery brand.

# Chapter 10: Conclusions & Future Aspect

- The biggest advantage of voice e-commerce is how easy it is to use.
- Online shopping more convenient, voice-enabled searches are becoming immensely popular and are being widely adopted.
- Voice search empowers you to communicate fast and obtain results more swiftly, which is an exciting feature for most users.
- Purchasing convenience is a major advantage for customers, allowing them to bypass a lengthy checkout process.
- Getting your customers to leave reviews can be difficult.
- With voice based commerce, the process is simplified.
- Assistive commerce is a broad term that refers to web app that enable people with disabilities to access technology.

# REFERENCES

**EXAMPLE OF REFERENCING AN ARTICLE FROM INTERNET**

- https://getbootstrap.com/docs/
- https://docs.python.org/3/
- https://www.w3schools.com/
- https://stackoverflow.com/

# APPENDIX

# BIBLIOGRAPHY

# RESEARCH PAPER