PYTHON PROGRAMMING
MINI PROJECT

Topic Name: Extracting Web Trends

Prepared by:

Marwin Shroff - 60004200097

Meet Patel - 60004200104

Kartik Jolapara - 60004200107

EXTRACTING TRENDS

Prepared by

Marwin Shroff- TE BTech Student – 60004200097

Meet Patel - TE BTech Student – 60004200104

Kartik Jolapara - TE BTech Student – 600042000107

Prepared for

Dr. Nilesh Patil

TABLE OF CONTENTS

	page
1. Problem Statement	1
2. Abstract	2
3. Functionalities of App	3
4. Implementation	4
5. Results	13
6. Conclusion	16

1. Problem Statement

In this social media savvy world, we have a lot of trends define the movement among GenZ and millennials. We received a challenge during the Flipkart hackathon which stated that as part of the challenge, teams are expected to identify trends from social media data. from all the products available on Flipkart identify products ,utilize all signals available (ex. posts, sessions, check-ins, social graphs, media content etc).

Deliverable 1:

Identification of trends from social media

1. Identify trends on social media based on category. Can restrict to Fashion as a category for the project. Ex: Polka dots dresses are trending on twitter.
2. Ranking/scoring logic for trends extracted.
3. Outcome format:
    a. Option1: List of trending keyword(s) along with list of sample images and respective links from which the trend is derived with most trending first:
    Example: Trends:[{Polka dot dresses, <list of links/images>,trending score}, {Bellbottom Jeans, <list of links/images>,trending score}..]
    b. Option 2: structured data according to flipkart category, sub category, vertical and product attributes
    Example: {category: Fashion, Sub-category: Women Western, vertical: Women dresses, trending attribute type: Pattern, trending attribute value: Polka Print, list of sample images and links from which the trend is derived}.
    Outcome with Option 2 format will be given bonus points.

Deliverable 2:

Mapping trends with Flipkart products:

1. Create mapping of extracted trending keyword(s) with Flipkart category, sub category, vertical and product attribute(s), search page links.
Example:{category: Fashion, Sub-category: Women Western, vertical: Women dresses, trending attribute type: Pattern, trending attribute value: Polka Print}
Note: Use category, Subcategory combination from the Flipkart Website
2. From a trending keyword, creating a corresponding searchable term on Flipkart which will lead to matching products.
Example: Tropical Tops keywords will not give right results directly on Flipkart but we can construct search query for it using some intelligence.
3. Points will be given based on similarity between sample images for trends and product results on Flipkart.

2. Abstract

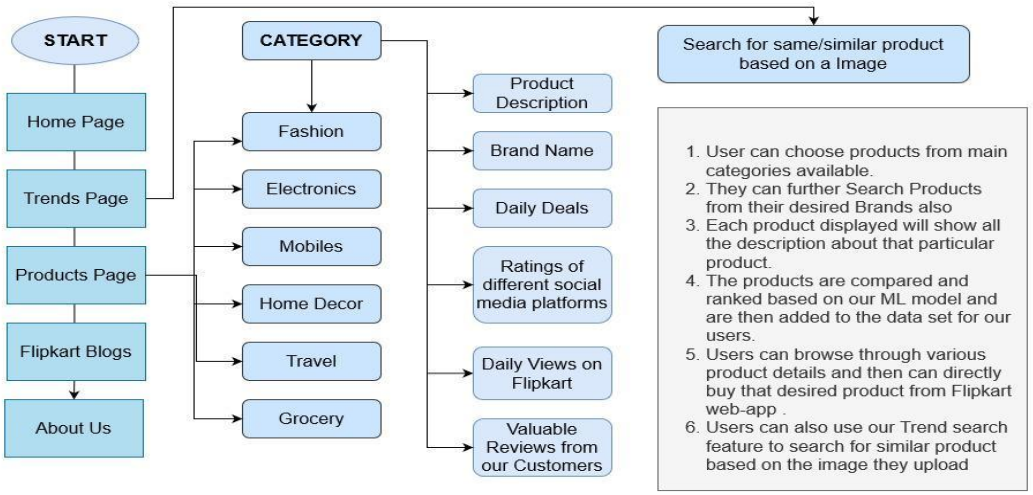
Social media generates a prodigious wealth of real-time content at an incessant rate. From all the content that people create and share, only a few topics manage to attract enough attention to rise to the top and become temporal trends which are displayed to users. The question of what factors cause the formation and persistence of trends is an important one that has not been answered yet. Social media nowadays is among the 'best possibilities available' to an item to get in touch with potential customers. Community social networking websites are the method to interact socially. These new media win the believe in of customers by linking with them at a deeper level. Community online marketing is the new mantra for several manufacturers since early a season ago. Promoters are considering many different social media possibilities and beginning to apply new social projects at a higher rate than ever before. Community online marketing and the companies that utilize it have become more sophisticated. One cannot afford to have no existence on the social programs if the competitor is creating waves with its solutions and items. The blast of social media trend is as amazing as that and the speed at which it is improving is frustrating. International companies have identified social media promotion as a potential promotion system, used them with enhancements to power their marketing with social media promotion.

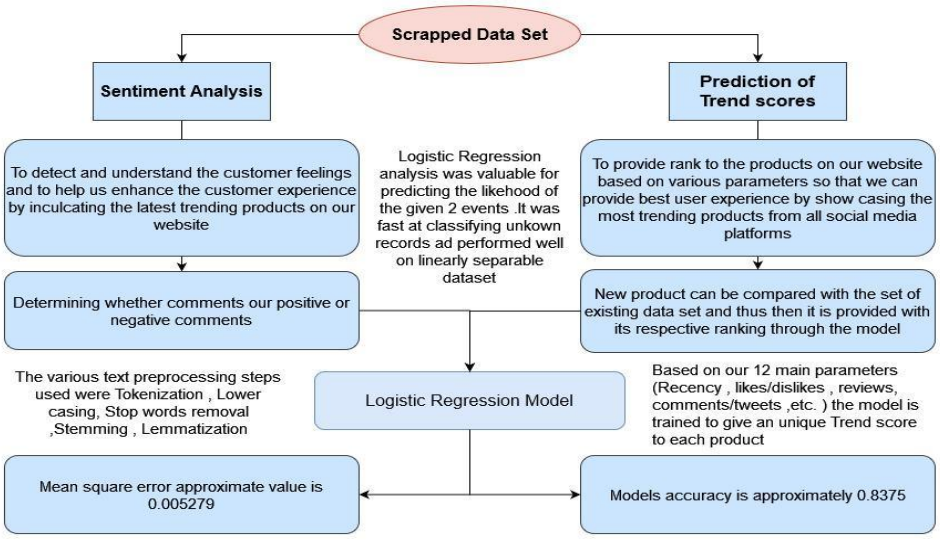
3. Functionality of App

1.Our Website aims at Providing the user with most accurate and time relevant data about the on-going trends and popular products on social media.
2.To achieve this the website facilitates in several ways –

- •Products have been divided into 6 main categories – Fashion , Mobile , Electronics , Travel , Home Décor and Grocery.
- •Each of this category is further sub divided into smaller other categories for easy browsing of trending product by the user .
- •For 24 hours of our online servicing our website contains a chatbot which can solve users query in an automated fashion .
- •The chatbot provides with the details of the latest trends , actions , suggestions about the ongoing trends etc.

3. The backend of the website is fuelled by two servers and complemented by various deep learning and machine learning model
4. Tech stacks used are ReactJs , Django rest framework





4. Implementation Details

Some snippets –

1) Views.py :

```
from django.http import JsonResponse
from django.conf import settings
from rest_framework.generics import GenericAPIView
from rest_framework import status
from rest_framework.decorators import api_view

from api.models import Blog, Brand, Category, Product, SubCategory, Video
from api.serializers import BlogSerializer, BrandSerializer, CategorySerializer, ProductSerializer, SubCategorySerializer, VideoSerializer

import csv, os
import urllib
import pickle
import numpy as np
import pandas as pd
from urllib.parse import urlparse
import urllib.request
from bs4 import BeautifulSoup
from django.core.files import File
from django.core.files.temp import NamedTemporaryFile

# Create your views here.
class Home(GenericAPIView):
    def get(self,request):
        return JsonResponse({'success' : 'success'})

class Videos(GenericAPIView):
    serializer_class = VideoSerializer
```

```
    queryset = Video.objects.all()

    def get(self,request, pk):
        if pk == '0':
            serializer = self.serializer_class(Video.objects.all(), many = True)
        else:
            video = Video.objects.filter(pk=pk).first()
            if video is None:
                return JsonResponse({'failure':'No such video exists'},status =
status.HTTP_404_NOT_FOUND , safe = False)
            video.views += 1
            video.save()
            serializer = self.serializer_class(video)
          return JsonResponse(serializer.data, status = status.HTTP_200_OK,
safe = False)

class Blogs(GenericAPIView):
    serializer_class = BlogSerializer
    queryset = Blog.objects.all()

    def get(self,request, pk):
        if pk == '0':
            serializer = self.serializer_class(Blog.objects.all(), many = True)
        else:
            obj = Blog.objects.filter(pk=pk).first()
            if obj is None:
                return JsonResponse({'failure':'No such blog exists'},status =
status.HTTP_404_NOT_FOUND , safe = False)
            serializer = self.serializer_class(obj)
          return JsonResponse(serializer.data, status = status.HTTP_200_OK,
safe = False)

    def post(self,request, pk):
        serializer = self.serializer_class(data=request.data)
```

```
        if serializer.is_valid():
            serializer.save()
                                return JsonResponse(serializer.data, status =
status.HTTP_201_CREATED, safe = False)
                            return JsonResponse(serializer.errors, status =
status.HTTP_400_BAD_REQUEST , safe = False)

@api_view(['POST'])
def all_products(request):
    filter = request.data['filter']
    if len(filter) == 1:
        category = Category.objects.get(name = filter[0])
                                products = Product.objects.filter(category =
category).order_by('rank')
        serializer = ProductSerializer(products, many = True)
    elif len(filter) == 2:
        subcategory = SubCategory.objects.get(name = filter[1])
                            products = Product.objects.filter(subcategory =
subcategory).order_by('rank')
        serializer = ProductSerializer(products, many = True)
    elif len(filter) == 3:
        category = Category.objects.get(name = filter[0])
        brand = Brand.objects.filter(name = filter[2]).first()
        if brand is None:
                return JsonResponse({'failure':'No such brand exists'},status =
status.HTTP_404_NOT_FOUND , safe = False)
            products = Product.objects.filter(category = category, brand =
brand).order_by('rank')
        serializer = ProductSerializer(products, many = True)
    else:
            serializer = ProductSerializer(Product.objects.all().order_by('rank'),
many = True)
    return JsonResponse(serializer.data, status = status.HTTP_200_OK, safe
= False)
```

```
@api_view(['POST'])
def brands(request):
    filter = request.data['filter']
    subcategory = SubCategory.objects.get(name = filter[1])
    products = Product.objects.filter(subcategory = subcategory)
    brands = Brand.objects.filter(id__in = products)
    serializer = BrandSerializer(brands, many = True)
    return JsonResponse(serializer.data, status = status.HTTP_200_OK, safe
= False)


@api_view(['GET',])
def get_trends(request):

    path = os.path.join(settings.BASE_DIR,"data.csv")

    with open(path, 'r') as csvfile:
        reader = csv.reader(csvfile)
        for row in reader:
            if row[0] == 'product name':
                continue
            if row[0] == '':
                break
            product = Product.objects.filter(name = row[0])
            discount = True
            if row[9]==row[13]:
                discount = False
            if len(product)==0:
                category,k =  Category.objects.get_or_create(name = row[10])
                brand,k =  Brand.objects.get_or_create(name = row[12])
                                                                    data      =
{'category':category.id,'brand':brand.id,'name':row[0],'price':float(row[9]),'
discount':discount,
```

```
                                                    'offer_price':float(row[13]),
'stock':row[14],'url':row[15],'hastags':row[8],'buyers':int(row[7])+int(row[6
]),

'rating':int(row[16]),'searches':int(row[2]),'viewers':int(row[17]),'rank':float
(row[19]),'image':None,
                }

                        req = urllib.request.Request(url=row[15], headers
={'User-Agent': 'Mozilla / 5.0 (X11 Linux x86_64) AppleWebKit / 537.36
(KHTML, like Gecko) Chrome / 52.0.2743.116 Safari / 537.36
PostmanRuntime/7.29.0'})
            response = urllib.request.urlopen(req)
            html_doc = response.read()
            soup = BeautifulSoup(html_doc, 'html.parser')
            json_object = soup.find(property='twitter:image')
            image_url = json_object.attrs['content']
            json_object = soup.find(property="og:description")
            description = json_object.attrs['content']
            data['description'] = description
            if category.id != 2 and category.id != 5:
                subcategory,k =  SubCategory.objects.get_or_create(name =
row[11],category=category.id)
                data['subcategory']=subcategory.id
            serializer = ProductSerializer(data=data)
            if serializer.is_valid(raise_exception=True):
                serializer.save()
            name = urlparse(image_url).path.split('/')[-1]
            img_temp = NamedTemporaryFile()
                    req = urllib.request.Request(url = image_url, headers=
{'User-Agent': 'Mozilla / 5.0 (X11 Linux x86_64) AppleWebKit / 537.36
(KHTML, like Gecko) Chrome / 52.0.2743.116 Safari / 537.36
PostmanRuntime/7.29.0'})
            img_temp.write(urllib.request.urlopen(req).read())
```

```
                img_temp.flush()
                try:
                    recipe = Product.objects.get(name = row[0])
                    recipe.image.save(name, File(img_temp))
                    recipe.save()
                except:
                    pass
            else:
                print(product)
                product[0].rank = float(row[18])
                product[0].save()
    content = {"detail":"Trends synchronized"}
    return JsonResponse(content, safe = False)

@api_view(['GET',])
def get_data(request):
    path = os.path.join(settings.BASE_DIR,"data.csv")
    pickled_model = pickle.load(open('./models/model_final_ann.pkl', 'rb'))
    df= pd.read_csv(path)
        df = df.drop(['product name', 'category','brand','url','category','sub
category', 'R', 'W', 'N', 'discount', 'stock', 'score'], axis=1)
    X = df.loc[:]
    result = pickled_model.predict(X[1:2])
    print(result)
    content = {"detail":"result"}
    #https://www.flipkart.com/search?q=tropical%20tops
    return JsonResponse(content, safe = False)

@api_view(['GET',])
def top_items(request):
    response_dict = {}
    categories = Category.objects.all()
    for category in categories:
                                            response_dict[category.name]      =
```

```
ProductSerializer(Product.objects.filter(category =
category).order_by('-buyers')[:4], many=True).data
    return JsonResponse(response_dict, safe = False)
```

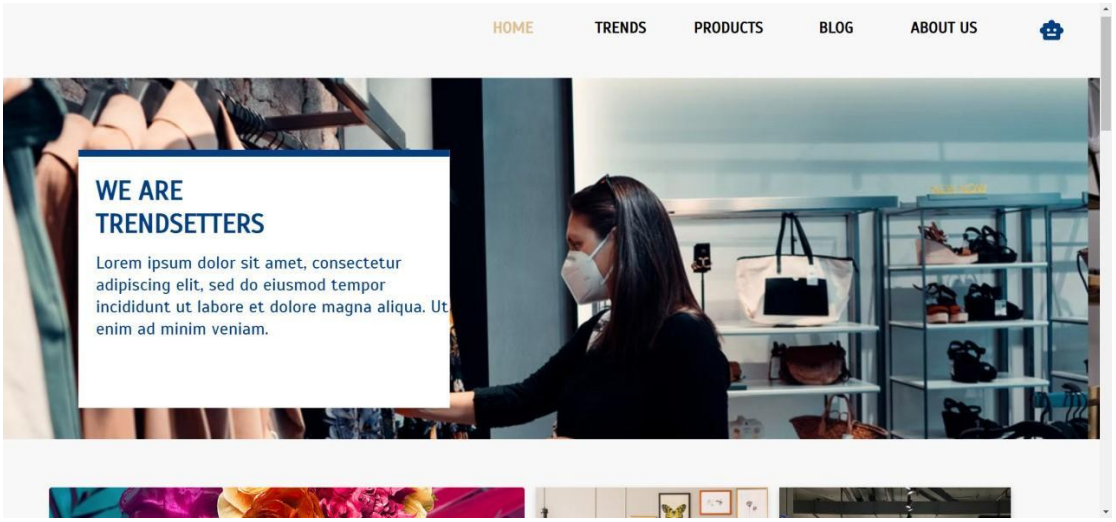
2) Models.py –

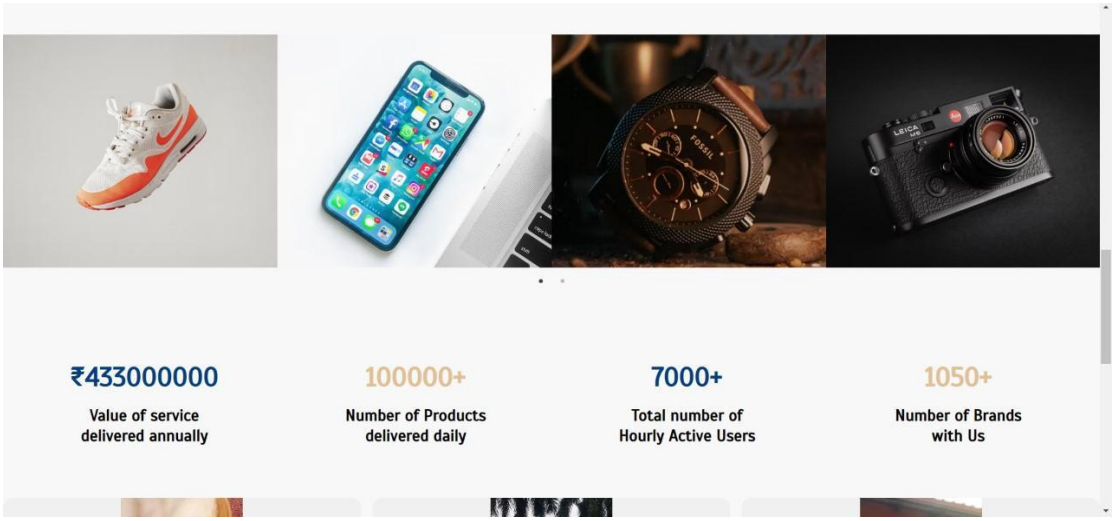
```
from django.db import models

class Category(models.Model):
    name = models.CharField(max_length = 25, unique=True)
        description = models.TextField(max_length=255, default = 'Best
products can be found here!')

    class Meta:
        verbose_name_plural = 'Categories'

    def __str__(self):
        return self.name

class SubCategory(models.Model):
                    category = models.ForeignKey(Category,
on_delete=models.CASCADE)
    name = models.CharField(max_length = 25, unique=True)
        description = models.TextField(max_length=255, default = 'Best
products can be found here!')

    class Meta:
        verbose_name_plural = 'Sub-Categories'

    def __str__(self):
        return self.name

class Brand(models.Model):
    name = models.CharField(max_length = 30, unique=True)
```

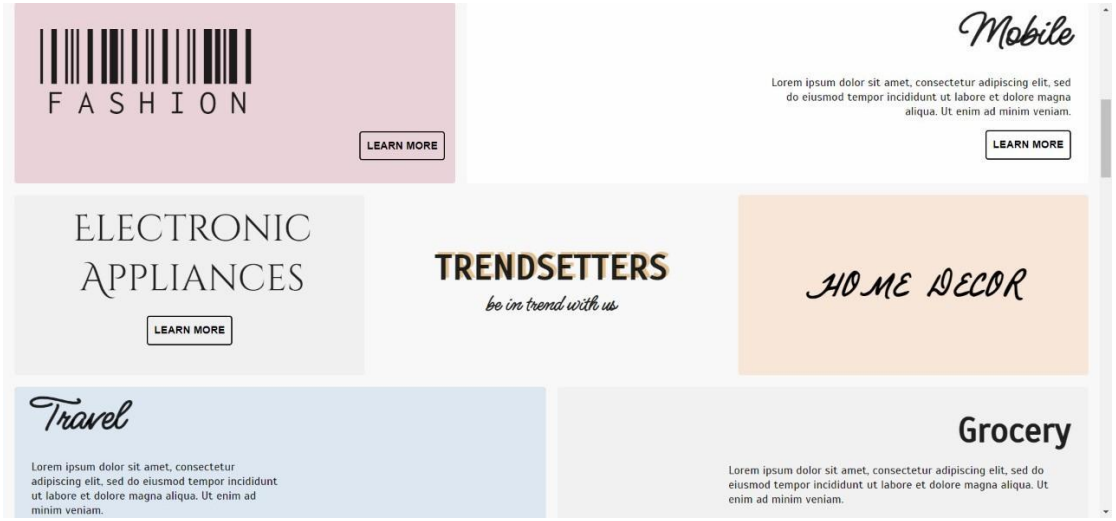
```
    def __str__(self):
        return self.name

def upload_path_handler(instance, filename):
    return "images/products/{label}/{file}".format(
        label=instance.name, file=filename
    )

class Product(models.Model):
                    category = models.ForeignKey(Category, on_delete=models.CASCADE)
                    subcategory = models.ForeignKey(SubCategory, on_delete=models.CASCADE, blank= True, null= True)
    brand = models.ForeignKey(Brand, on_delete=models.CASCADE)
    name = models.CharField(max_length = 1000, unique=True)
    price = models.FloatField(default = 0.0)
    discount = models.BooleanField(default = False)
    offer_price = models.FloatField(default=0.0)
    stock = models.PositiveIntegerField(default = 0)
     description = models.CharField(max_length = 2000, blank= True, null= True, default = 'Selling fast, Hurry up!')
       image = models.ImageField(upload_to = upload_path_handler,null = True, blank = True)
         url = models.URLField(default='www.example.com', max_length =2000)

    hastags = models.PositiveIntegerField(default = 0)
    buyers = models.PositiveIntegerField(default = 0)
    rating = models.PositiveIntegerField(default = 0)
    searches = models.PositiveIntegerField(default = 0)
    viewers = models.PositiveIntegerField(default = 0)
    rank = models.FloatField(default = 0.0)

    def __str__(self):
```

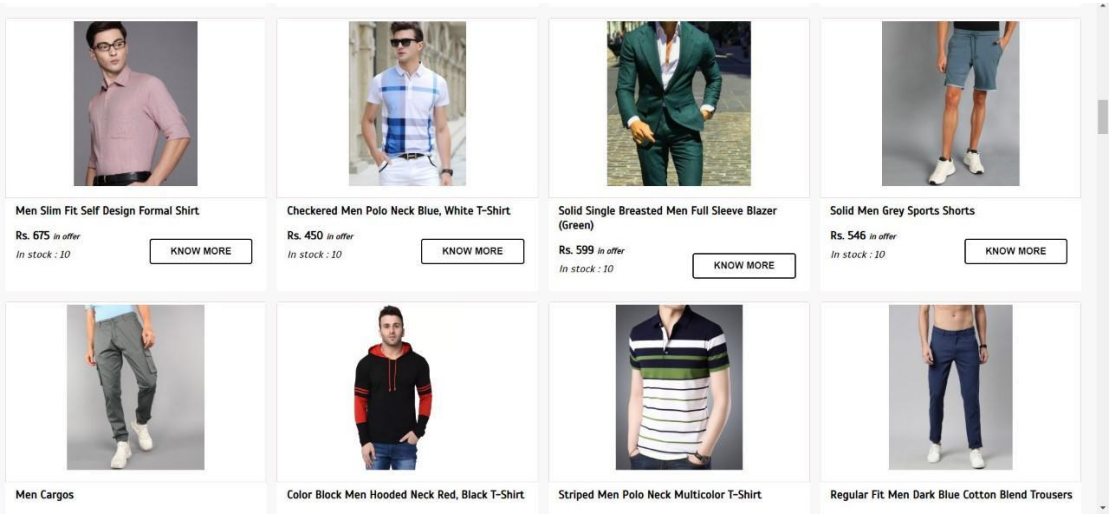
```
        return self.name

def upload_path_handler(instance, filename):
    return "videos/{label}/{file}".format(
        label=instance.name, file=filename
    )

class Video(models.Model):
    name = models.CharField(max_length = 100)
      file = models.FileField(upload_to = upload_path_handler,null = True,
blank = True)
    views = models.PositiveIntegerField(default = 0)

    def __str__(self):
        return self.name

def upload_path_handler(instance, filename):
    return "images/blogs/{label}/{file}".format(
        label=instance.title, file=filename
    )

class Blog(models.Model):
    title = models.CharField(max_length=100)
    content = models.TextField(max_length=1000)
      image = models.ImageField(upload_to = upload_path_handler,null =
True, blank = True)

    def __str__(self):
        return self.title
```

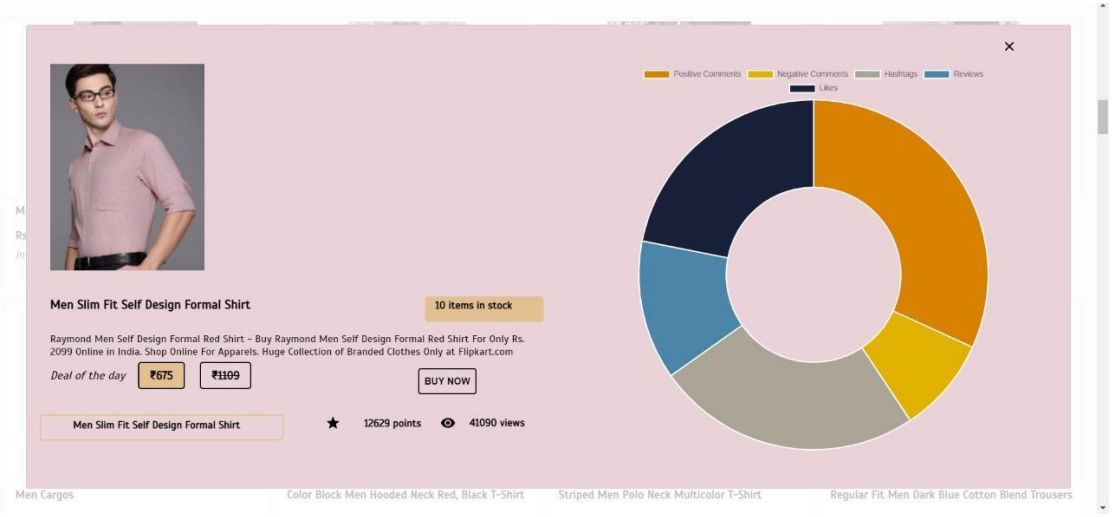
5. Results

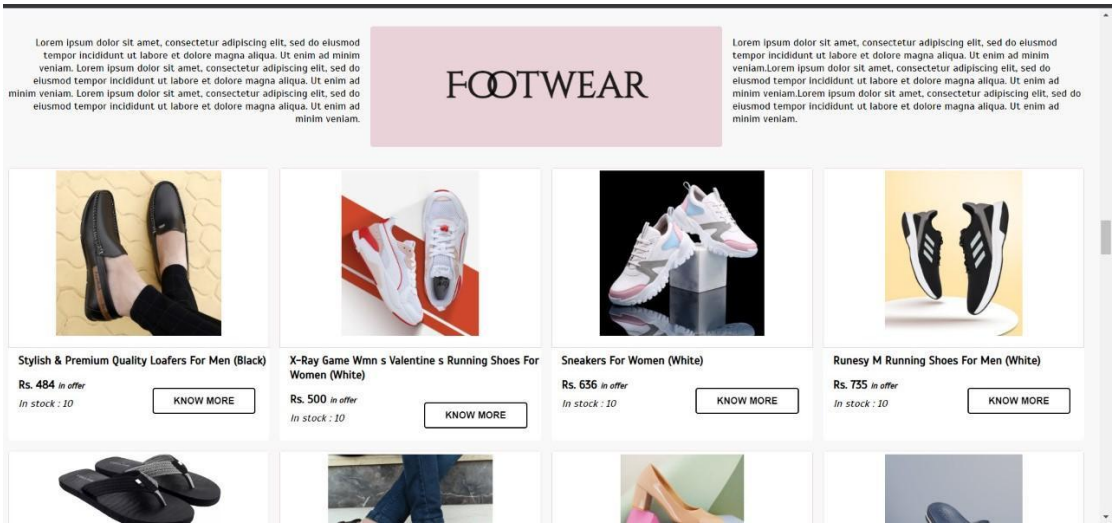

HOME
TRENDS
PRODUCTS
BLOG
ABOUT US
WE ARE TRENDSETTERS
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam.

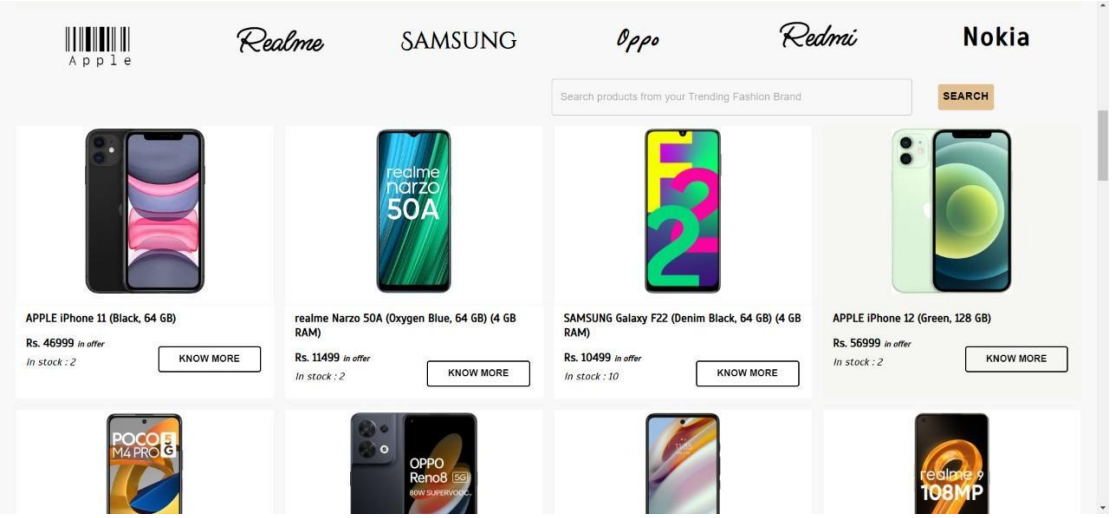

₹433000000
Value of service delivered annually
100000+
Number of Products delivered daily
7000+
Total number of Hourly Active Users
1050+
Number of Brands with Us

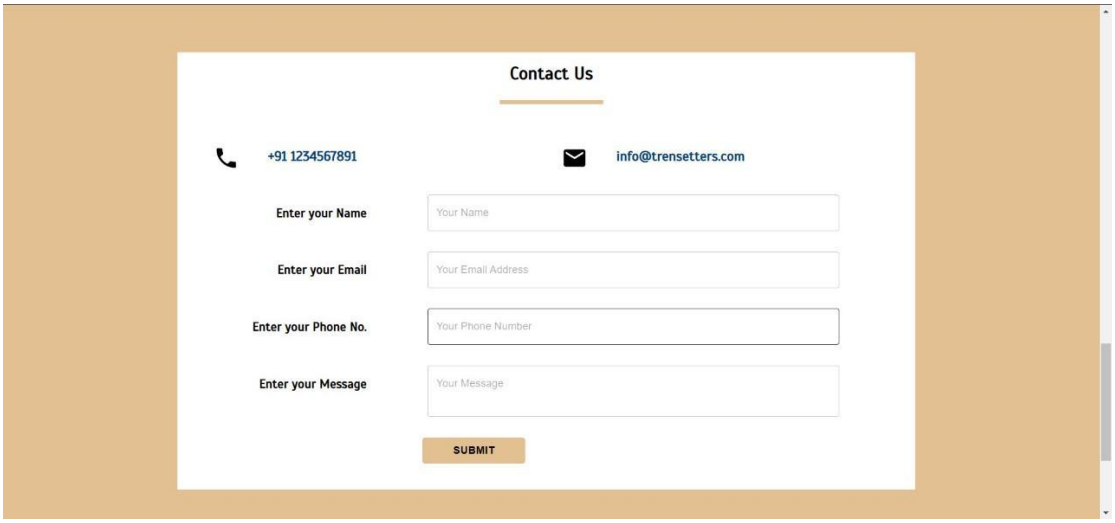

FASHION
LEARN MORE
Mobile
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam.
LEARN MORE
ELECTRONIC APPLIANCES
LEARN MORE
TRENDSETTERS
be in trend with us
HOME DECOR
Travel
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam.
Grocery
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam.

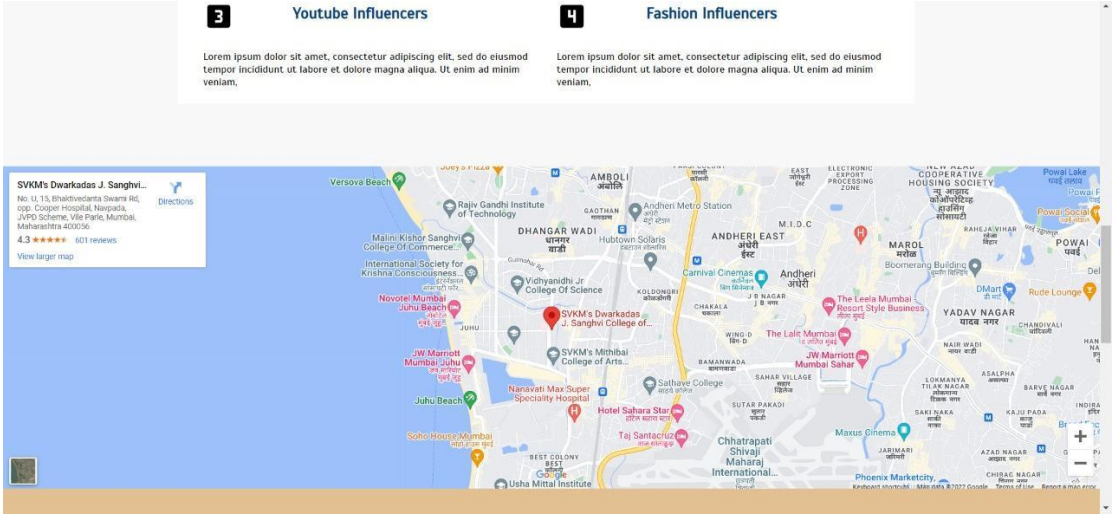

Men Slim Fit Self Design Formal Shirt
Rs. 675 in offer
In stock : 10
KNOW MORE
Checkered Men Polo Neck Blue, White T-Shirt
Rs. 450 in offer
In stock : 10
KNOW MORE
Solid Single Breasted Men Full Sleeve Blazer (Green)
Rs. 599 in offer
In stock : 10
KNOW MORE
Solid Men Grey Sports Shorts
Rs. 546 in offer
In stock : 10
KNOW MORE
Men Cargos
Color Block Men Hooded Neck Red, Black T-Shirt
Striped Men Polo Neck Multicolor T-Shirt
Regular Fit Men Dark Blue Cotton Blend Trousers


Positive Comments
Negative Comments
Hashtags
Reviews
Likes
Men Slim Fit Self Design Formal Shirt
10 items in stock
Raymond Men Self Design Formal Red Shirt - Buy Raymond Men Self Design Formal Red Shirt For Only Rs. 2099 Online in India. Shop Online For Apparels. Huge Collection of Branded Clothes Only at Flipkart.com
Deal of the day
₹675
₹1109
BUY NOW
Men Slim Fit Self Design Formal Shirt
12629 points
41090 views


Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam.
FOOTWEAR
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam.Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam.
Stylish & Premium Quality Loafers For Men (Black)
Rs. 484 in offer
In stock : 10
KNOW MORE
X-Ray Game Wmn s Valentine s Running Shoes For Women (White)
Rs. 500 in offer
In stock : 10
KNOW MORE
Sneakers For Women (White)
Rs. 636 in offer
In stock : 10
KNOW MORE
Runesy M Running Shoes For Men (White)
Rs. 735 in offer
In stock : 10
KNOW MORE


Apple
Realme
SAMSUNG
Oppo
Redmi
Nokia
Search products from your Trending Fashion Brand
SEARCH
APPLE iPhone 11 (Black, 64 GB)
Rs. 46999 in offer
In stock : 2
KNOW MORE
realme Narzo 50A (Oxygen Blue, 64 GB) (4 GB RAM)
Rs. 11499 in offer
In stock : 2
KNOW MORE
SAMSUNG Galaxy F22 (Denim Black, 64 GB) (4 GB RAM)
Rs. 10499 in offer
In stock : 10
KNOW MORE
APPLE iPhone 12 (Green, 128 GB)
Rs. 56999 in offer
In stock : 2
KNOW MORE


Contact Us
+91 1234567891
info@trensetters.com
Enter your Name
Your Name
Enter your Email
Your Email Address
Enter your Phone No.
Your Phone Number
Enter your Message
Your Message
SUBMIT


3
Youtube Influencers
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam,
4
Fashion Influencers
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam,
SVKM's Dwarkadas J. Sanghvi...
Directions
View larger map
Versova Beach
Rajiv Gandhi Institute of Technology
Andheri Metro Station
DHANGAR WADI
Hubtown Solaris
ANDHERI EAST
Malini Kishor Sanghvi College Of Commerce...
International Society for Krishna Consciousness
Vidhyanidhi Jr College Of Science
Carnival Cinemas
Andheri
MAROL
Boomerang Building
Novotel Mumbai Juhu Beach
SVKM's Dwarkadas J. Sanghvi College of...
The Leela Mumbai Resort Style Business
DMart
Rude Lounge
YADAV NAGAR
The Lalit Mumbai
JW Marriott Mumbai Juhu
SVKM's Mithibai College of Arts...
JW Marriott Mumbai Sahar
Sathaye College
Nanavati Max Super Speciality Hospital
Hotel Sahara Star
Juhu Beach
Taj Santacruz
Maxus Cinema
Soho House Mumbai
Chhatrapati Shivaji Maharaj International...
Phoenix Marketcity
Usha Mittal Institute

6. Conclusion

In times where Internet is a vast repository of almost everything and E-Commerce has replaced our next-door grocery, it is too difficult for shoppers to pick the actual best. So, a Django and react based web application of implemented which extract trends from inserted data which includes the product buyer statistics, reviews, ratings and social media trends. This application picks the top or “trending” products using the data and displays them on dashboard with a direct Flipkart link to buy it. The features and use cases of Django framework were studied by the means of this project.