

# **SMART FASHION RECOMMENDER APPLICATION**

## **A PROJECT REPORT**

**Submitted by**

**D PRASANIYA (191031068)**

**B SANJANA (191031082)**

**V YAMINI (191031118)**

**K SIVASHREE (191031097)**

**In Partial fulfillment for the award of the degree**

**of**

**BACHELOR OF ENGINEERING**

**IN**

**COMPUTER SCIENCE ENGINEERING**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**MAHENDRA ENGINEERING COLLEGE**

**(Autonomus)**

**Mahendrapuri, Mallasamudram, Namakkal DT-637 503**

**NOVEMBER – 2022**

**MAHENDRA ENGINEERING COLLEGE**

**(Autonomous)**

**Mahendrapuri, Mallasamudram, Namakkal DT-637 503**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**BONAFIDE CERTIFICATE**

Certified that this project report “**SMART FASHION RECOMMENDER APPLICATION**” is a bonafide work “**D PRASANIYA (191031068), B SANJANA(191031082) ,V YAMINI(191031118), K SIVASHREE(191031097)**” who carried out this project work under my supervision.

**SIGNATURE**

Prof.DR.M.Kannan,M.E,M.S,Ph.D.

HEAD OF THE DEPARTMENT

**SIGNATURE**

DR.A.Neelamadheswari

Associate Professor

M.E.,Ph.D

**DR.D.Sujeetha,M.E**

Assistant Professor

Supervisor

**Date:**

**Date:**

**MAHENDRA ENGINEERING COLLEGE**

**(Autonomous)**

**Mahendrapuri, Mallasamudram, Namakkal DT-637 503**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**CERTIFICATE OF PROJECT APPROVAL**

This is to certify that the project report titled **“SMART FASHION RECOMMENDER APPLICATION”** is an approved record of work done by **“D PRASANIYA (191031068), B SANJANA (191031082), V YAMINI (191031118), K SIVASHREE (191031097)”** in partial fulfillment for the award of the degree of B.E Computer Science & Engineering during the Academic year **2022-2023**.

**SUPERVISOR**

**DR.A.Neelamadheswari  
M.E,Ph.D**

**HEAD OF THE DEPARTMENT**

**Prof.DR.M.Kannan,M.E.,M.S.,Ph.D.**

Submitted for the end semester project viva-voice exam held  
on\_\_\_\_\_ at\_\_\_\_\_

**INTERNAL EXAMINER**

**EXTERNAL EXAMINER**

**Date:**

**Date:**

## ACKNOWLEDGEMENT

We express our earnest thanks with deepest respect and gratitude in our Honourable Chairman

“**Shri.M.G.BHARATHKUMAR,M.A.,B.Ed.**” and respected Managing Directors “**Er.BA.MAHENDIRAN,B.E.,M.I.S.T.E.**, and “**Er.BA.MAHA AJAY PRASATH,MS** who has provide excellent facilities for us.

We feel happy to convey our kind regards and sincere thanks to our be loved **DR.V.SHANMUGAM.,M.E.,Ph.D.**, who provide his kind of concern for carring out this project work and providing suitable environment to work.

We wish to express our sense of gratitude and sincere that's to our **HEAD OF THE DEAPARTMENT Prof.DR.M.Kannan,M.E.,M.S.,Ph.D. of COMPUTER SCIENCE AND ENGINEERING** for his valuable guidance in the preparation and presentation of this project.

We have express our profound and sincere thanks to our mini project guide **DR.A.Neelamadheswari Associate Professor M.E.,Ph.D of COMPUTER SCIENCE AND ENGINEERING** for having given constant encouragement with helping a lots also we wish to record our indebtedness to here for courages support and guidance on various stages of this project

I Specially thanks all our friends, parents, teaching and non teaching staffs and our well wishers for their constant support all the time.

## Content

CHAPTER No.		TITLE
		<b>Introduction</b>
<b>1.1</b>		<b>Project Overview</b>
<b>1.2</b>		<b>Purpose</b>
<b>2</b>		<b>Literature Survey</b>
<b>2.1</b>		<b>Existing Problem</b>
<b>2.2</b>		<b>Reference</b>
<b>2.3</b>		<b>Problem Statement definition</b>
<b>5.1</b>		<b>Data flow Diagrams</b>
<b>5.2</b>		<b>Solutions and Technical Architecture</b>
<b>5.3</b>		<b>User stories</b>
<b>6</b>		<b>Project planning &amp;Scheduling</b>
<b>6.1</b>		<b>Sprint Planning and Estimation</b>
<b>6.2</b>		<b>Sprint Delivery schedule</b>
<b>6.3</b>		<b>Reports from JIRA</b>
<b>7</b>		<b>Coding &amp; Solutioning</b>
<b>7.1</b>		<b>Python code</b>
<b>7.2</b>		<b>Html coding</b>
<b>8</b>		<b>Testing</b>
<b>8.1</b>		<b>Test cases</b>
<b>8.2</b>		<b>User Acceptance Testing</b>
<b>9</b>		<b>Results</b>
<b>9.1</b>		<b>Performance metrics</b>
<b>9.1.1</b>		<b>Home page</b>
<b>9.1.2</b>		<b>Register page</b>
<b>9.1.3</b>		<b>Login page</b>
<b>9.1.4</b>		<b>User details</b>
<b>9.1.5</b>		<b>Product search</b>

<b>10</b>		<b>Advantages and Disadvantages</b>
<b>11</b>		<b>Conclusion</b>
<b>12</b>		<b>Future work</b>
<b>13</b>		<b>Appendix</b>

## **Introduction**

### **1.1 Project overview**

A recommendation system is an integral part of any modern online shopping or social network platform. The product recommendation system as a typical example of the legacy recommendation systems suffers from two major drawbacks: recommendation redundancy and unpredictability concerning new items (cold start). These limitations take place because the legacy recommendation systems rely only on the user's previous buying behavior to recommend new items. Incorporating the user's social features, such as personality traits and topical interest, might help alleviate the cold start and remove recommendation redundancy. we propose Meta-Interest, a personality-aware product recommendation system based on user interest mining and meta path discovery. Meta-Interest predicts the user's interest and the items associated with these interests, even if the user's history does not contain these items or similar ones. This is done by analyzing the user's topical interests and, eventually, recommending the items associated with the user's interest. The proposed system is personality-aware from two aspects; it incorporates the user's personality traits to predict his/her topics of interest and to match the user's personality facets with the associated items. The proposed system was compared against recent recommendation methods, such as AI based watson-based recommendation using flask framework is implemented here

### **1.2 Purpose**

The purpose of the work E- commerce system is implemented for product recommendation system, online based system application is used for purchase the product recommendation tools use a set of algorithms to show relevant products to your visitors. They do this by using data like location, gender, and more specific information like purchase intent.

## **2. Literature survey**

McAuley et al. [1] devised a parametric distance transformation that assigns a lower distance to garment pairings that fit well than to those that do not. And provided Image-based recommendations on styles and substitutes

Hu et al. [2] conducted a preliminary investigation into personalised outfit recommendation. To describe the user-item and item-item interactions, a functional tensor factorization method was presented. They proposed A functional tensor factorization approach

Thombre in [3] used image segmentation and Kalman filter to realize Human detection and tracking. Orrite-Urunuela proposed a statistical model for detection and tracking of human silhouette and the corresponding 3D skeletal structure in gait sequences

Veit et al. [4] learned feature transformation for a compatibility measure between pairs of objects using a Siamese CNN architecture. All of these works focused solely on the compatibility of two things. Furthermore, they simply modelled broad matching criteria and ignored the issue of personalisation

Ajmani et al. [5] present a novel method for contentbased recommendation of media-rich commodities with the use of probabilistic multimedia ontology. Proposed an ontology based personalized garment recommendation system.

Li et al. [6] utilized the HMM of recommended items to match customers' model according to customer data. The second method is the collaborative filtering-based recommendations algorithm. Proposed Content-Based Filtering Recommendation Algorithm

For instance, Nogueira et al. [7] presented a new collaborative filtering strategy that utilizes the visual attention to characterize images and alleviate the new item cold-start problem. The rule-based recommendation algorithm is the third method.

Hwang et al. [8] put forward a method to generate the automatic rules with the user's items and made a suggestion on the best rule. The fourth method is the utility-based recommendation. Scholz et al.

[9] found that exponential utility functions are better geared to predicting optimal recommendation ranks for products, and linear utility functions perform much better in estimating customers' willingness.

Koenig in [10] developed a system toward real-time human detection and tracking in diverse environments. However, mostly the researchers focus on the point of human detection and tracking in complex scene, while refined contour extraction of human in dynamic scene is still an open question.

## 2.1 EXISTING PROBLEM

In existing system only simple web application and their rating has been implemented in existing system, An ecommerce product recommendation engine is a piece of technology that displays recommended products to shoppers throughout your store. It uses machine learning to get smarter and show increasingly relevant products to shoppers based on their interests and previous browsing behavior

In existing model is content based filtering scheme has been employed in existing model **The content-based filtering method** analyzes customer data on the likes and dislikes of each user (cookies allow tracking over multiple visits), then makes recommendations based on the browsing history of that user. The idea behind content-based filtering is that if you enjoy a certain item, you'll likely also enjoy a similar item. An example of a content-based filtering system would be if you were listening to Pandora and consistently 'liked' downtempo jazz music.

**The collaborative-filtering method** incorporates data from users who have purchased similar products, then combines that information to make decisions about recommendations. The advantage to this filtering method is that it is capable of making complex recommendations on items such as music or movies without having to 'understand' what the item is. This method of filtering operates under the assumption that users will prefer recommendations that are based on purchases they made in the past. Here's an example: If customer A likes a specific line of products that customer B also likes (assuming they have similar interests), then collaborative-filtering would assume that customer A would like other products that customer B purchased and vice versa.

**A hybrid method** combines the content-based and collaborative-based methods to incorporate group decisions but focuses the output based on the attributes of a specific visitor. An example of a hybrid filtering system would be how Spotify curates its personalized 'Discover Weekly' playlists. If you've ever listened to a personalized Spotify playlist, it's shocking how accurately they're able to recommend songs based on what you like. The secret behind how they pull this off is through a complex hybrid filtering system that aggregates data on your listening habits as well as similar users' listening habits, to create a playlist of unique songs that align with your personal taste.



## 2.2 References

1. I. Esslimani, A. Brun, and A. Boyer, “A collaborative filtering approach combining clustering and navigational based correlations,” in Proc. 5th Int. Conf. Web Inf. Syst. Technol., 2009, pp. 364–369
2. S. Zhang, W. Wang, J. Ford, and F. Makedon, “Learning from incomplete ratings using non-negative matrix factorization,” in Proc. 6th SIAM Int. Conf. Data Mining, 2006, pp. 549–553.
3. T. Hofmann and J. Puzicha, “Latent class models for collaborative filtering,” in Proc. 6th Int. Joint Conf. Artif. Intell., 1999, pp. 688–693.
4. B. M. Sarwar, G. Karypis, J. A. Konstan, and J. Reidl, “Item-based collaborative filtering recommendation algorithms,” in Proc. 10th Int. World Wide Web Conf., 2001, pp. 285–295
5. T. George and S. Merugu, “A scalable collaborative filtering framework based on co-clustering,” in Proc. 5th IEEE Int. Conf. Data Mining, 2005, pp. 625–628

## 2.3 Problem statement definition

- The problem of the work is to design static web applications deployments with customer deployment
- Lack of interaction between application and user
- User need to navigate across multiple pages to choose right product
- Confusion in choosing product
- Lack of sales
- Complex User Interface.
- Lack of proper guidance.

### 3. Ideation and Proposed solution

#### 3.1 Empathy Map Canvas



### 3.2 Ideation And Brainstorming

## 2

### Brainstorm

Write down any ideas that come to mind that address your problem statement.

🕒 10 minutes

**Team member 1: Praveen D.**

Outlets	smart sock gen 3	Makeup things
Wear habits like rub	Wear habit like rye	Shirts Accessories
Body Accessories		

**Team member 1: Sargun B.**

Customer Interest	Customer suggestion	Customer Search engine
AIo smart device	smart coding jacket	food X yoga pants
Sports things		

**Team member 2: Vivek V.**

Maids Accessories	smart shirt	smart ballroom
smart coding	smart baby tie	smart coding jacket
spoonbender		

**Team member 2: Shashank K.**

smart device	smart home products	city camera
wear camera	Digital camera	smart bedies
smart baby		

**TIP**

You can select a sticky note and hit the pencil icon (switch to sketch) icon to start drawing!

### Improving product sales

Customer suggestion	Customer Interest	Quality products	Customer interested in collected places
---------------------	-------------------	------------------	---

### Employer details

Employee name	Email id and Gali	Contact number	Address
---------------	-------------------	----------------	---------

### List of Mandatory Credentials

Name	Contact Number	Address	Payment details
------	----------------	---------	-----------------

### Customer details

Name	Contact Number	Email id	Address
------	----------------	----------	---------

## 3

### Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you can break it up into smaller sub-groups.

🕒 20 minutes

### Usages

But, right now it is not ready	Used only by smart baby	already present and have already been used
Wearable in hospital or home doctor	This distribution will have priority	City, quality and not for production level

### Social media

Future design of people and company	Customizable and unique items
Group discussion	

### Features

Smart wear can be used in any situation	Customizable and unique items
Smart wear can be used in any situation	Smart wear can be used in any situation

### Recommendation

For quality and price	For quality and price
-----------------------	-----------------------

### Notification

Smart wear can be used in any situation
---

### Security

Secure personal information	Secure user self information
-----------------------------	------------------------------

**TIP**

Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas as themes within your mural.

### 3.3 Proposed solutions

The proposed solutions web-based chatbot based applications is implemented with dash board applications is implemented for web applications The aim of this research is to build a perfume recommendation system. This system will help the user to get required perfumes. For that user has to provide description as a search query about the perfume according to his interest. This description can contain feelings, emotions, description, likes, dislikes and brand of the perfume. A chat bot will help the user to get the input in the form of search query and then provide the output as a recommended perfume what user is looking for. Initial work for research is collecting a data. Data required for this research contained the details in the form of name, brand, text descriptions, reviews, a list of notes. As we are using natural language processing, the text data must be pre-processed. It covers some tasks like making text data to lower case, removing stop words, tokenization, stemming, etc. shows tasks of pre-processing of data. Lowercasing – Lowercasing is the first step in data

S.NO	Parameter	Description
1.	Problem Statement (Problem to be solved)	If there is money or products Problem (Customer satisfaction problem) during online shopping Like customers have to visit many Website for the different products
2.	Idea /Solution description	In view of rapid growth of multimedia data where visual information is a key component further research on application of multimodal fusion and multi task learning in fashion recommender
3.	Novelty/Uniqueness	Recommender cannot accurately judge item whether the user knows if the user never interacted with it the simplest way to quantify novelty is the average popularity of top recommended items.
4.	Social Impact /Customer Satisfaction	Customer satisfaction provides us with insight into things that need improvement or other ways we can improve our service or product to better service our customer.
5.	Business Model (Revenue Models)	To establish the best possible price for what they selling to maximize profits once the pricing strategy is set the revenue model will dictate how customers pay that price when they purchase.
6.	Scalability of Solution	At present research at home and aboard mainly focuses on recommendation quality and there is less discussion on scalability.
<div> <div>Define CS, TM into CC</div> <div>Focus on JSP, find into BE, understand RC</div> <div>Identify strong TR &amp; EM</div> </div> <div> <div>4. EMOTIONS: BEFORE / AFTER</div> <div>EM</div> <div>If the customer face a problem the customer should get anger but if they are fixed it could</div> </div>		

### 3.4 Problem solution fit

## 4.Requirement Analysis

## 4.1 Functional requirement

FR NO.	Functional Requirement (Epic)	Sub Requirement(story/sub-task)
FR-1	User Registration	Registration through Form Registration through Gmail
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User Login	Enter username and password
FR-4	User Search	Search for desired product
FR-5	User Authentication	The process of verifying the identity of user when that user login in to an Application.
FR-6	Reporting	It is essential to product performance or quality over any period of time.

## 4.2 non-functional requirement

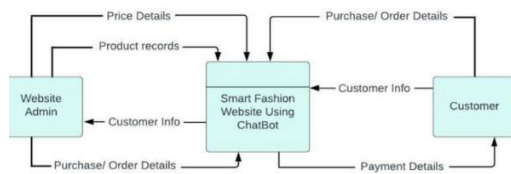
FR NO.	Non-Functional Requirement	Description
NFR-1	Usability	It software has a high level usability it means that users can handle a software interface easily.
NFR-2	Security	Access permissions for the particular system information may only be changed by the system data administrator
NFR-3	Reliability	Technology that is highly reliable functions with the same or similar efficiently after extensive use.
NFR-4	Performance	The website load time should not be more than one seconds for users.
NFR-5	Availability	Customers visit the website at any time during the day
NFR-6	Scalability	The ability of a system to handle a growing amount of work.

## 5.Project Design

### 5.1 Data flow diagrams

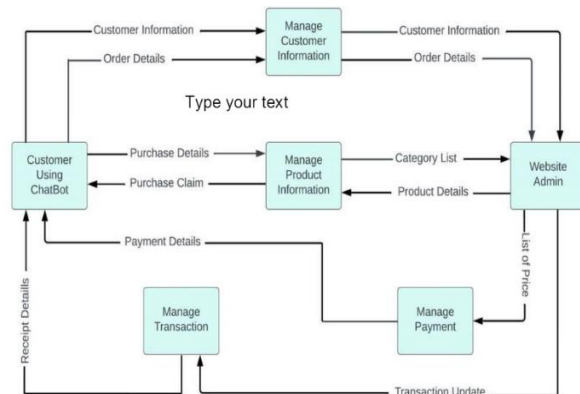
A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.

FLOW:  
DFD Level 0



- Admin is maintaining all the things that the users are purchasing
- To keep track of the stock information
- The chatbot can give payment details to the users.
- We can manage user selections and orders using chatbots

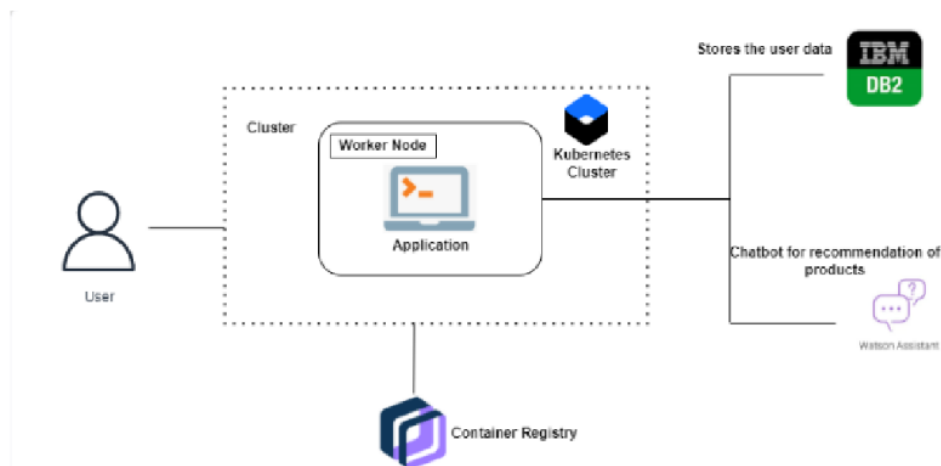
DFD Level 1



## 5.2 Solutions and Technical Architecture

Recent years have seen a startling increase in the popularity of online clothing purchasing. in light of the current coronavirus crisis, in particular. The potential of the global fashion e-commerce market is starting to be recognised. An industry in transition that has made significant progress but is still far from flawless. The explicit feedback, implicit feedback, and hybrid feedback are the three types of input that the recommendation system is built upon. Explicit feedback must be of the highest calibre since it includes users' direct comments indicating their interest in or preference for a production.

FRS is a method of matching features between fashion products and customers or users based on predetermined criteria. Various studies examined the design of colours, clothing forms, and other aspects of apparel.



S.NO	Component	Description	Technology
1.	User Interface	User interacts with application through web UI	HTML, Javascript , CSS
2.	Cloud Database	Cloud database can be offered as managed database as a service	IBM DB2, IBM Cloud ant
3.	File Storage	Necessary files for the application should be maintained at easily accessible place	IBM Block Storage or Other Storage Service or Local Filesystem
4.	Recommenders' products chatbot	A Products recommendation chatbot facilitates this experience. Instead of scrolling through a dozen product pages, just answer a few questions and get the perfect recommendation within minutes, if not seconds.	IBM Watson Assistant
5.	Database	It refers to a set of programs that allows users to access databases and manipulate, maintain, report on and link data.	MYSQL
6.	Infrastructure (server/cloud)	Application Deployment on local System /Cloud Local Server Configuration Cloud Server Configuration	Local Cloud Foundry, Kubernetes

## 5.3 User Stories

The operation of your online store is guided by correctly defined user stories, which also aid the development team's memory of the environment in which they are designing. User stories are explicitly created from an end-user perspective in agile software development, explaining the type of user, what they want, and why they want it.

User stories are frequently written on index cards or post-it notes in an effort to keep them concise and unambiguous and to reduce the amount of documentation to a framework that is solely focused on the needs of the customer.

Because, let's face it, how often do your teams read a sophisticated, long, technical requirements document?, this narrative card writing format is tremendously helpful. User stories help teams in charge of building and executing user-focused technologies communicate effectively despite the density.

## 6.Project planning &Scheduling

### 6.1 Sprint Planning and Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story, Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN 1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	V.Yamini, D.Prasaniya
Sprint-1	Confirmation	USN-2	As a user, I will receive confirmation email once I have registered for the application	1	High	K.Sivashree, B.Sanjana
Sprint-1	Login	USN-3	As a user, I can log into the application by entering email & password	1	High	B.Sanjana, D.Prasaniya
Sprint-2	Dashboard	USN-4	Bro-vse the products that are offered on the website. The customer can interact with chat bot directly about the products rather than navigate through various screens.	3	High	D.Prasaniya, V.Yamini
Sprint-3	Chat Bot	USN-4	The user can directly talk to Chatbot regarding the products. Get the recommendations based on information provided by the use	2	High	K.Sivashree, V.Yamini
Sprint-3	Tracking the order	USN-5	As a user, If I order any product, chat bot notifies it.	2	High	K.Sivashree, D.Prasaniya
Sprint-4	Flow of orders and check out	USN-6	As a user, I can track my order and collect information about shipping	2	High	B.Sanjana, V.Yamini
Sprint -4	Feedback, comment section	USN-7	As a user I can write a fashion review as both positive and negative.	2	High	D.Prasaniya, B.Sanjana
Sprint-4	Return the product if not satisfied	USN-8	As a user, If I am not satisfying with the product, I can return in 7 days from the date of delivery	2	Medium	V.Yamini, K.Sivashree

### 6.2 Sprint Delivery Schedule



Since sprints take place over a fixed period of time, it's critical to avoid wasting time during planning and development. And this is precisely where sprint scheduling enters the equation.

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	04	5 Days	28 Oct 2022	01 Nov 2022		01 Nov 2022
Sprint-2	03	5 Days	03 Nov 2022	07 Nov 2022		
Sprint-3	04	5 Days	09 Nov 2022	13 Nov 2022		
Sprint-4	06	5 Days	15 Nov 2022	19 Nov 2022		

Here on the above table it is mentioned what's the duration we planned the sprint and it's start date and end date and we completed based on what we planned.

### Velocity:

Sprint - 1: Average Velocity(AV)

$$Av = \text{Sprint Duration} / \text{Velocity} = 4/5 = 0.8$$

Sprint - 2: Average Velocity(AV)

$$Av = \text{Sprint Duration} / \text{Velocity} = 3/5 = 0.6$$

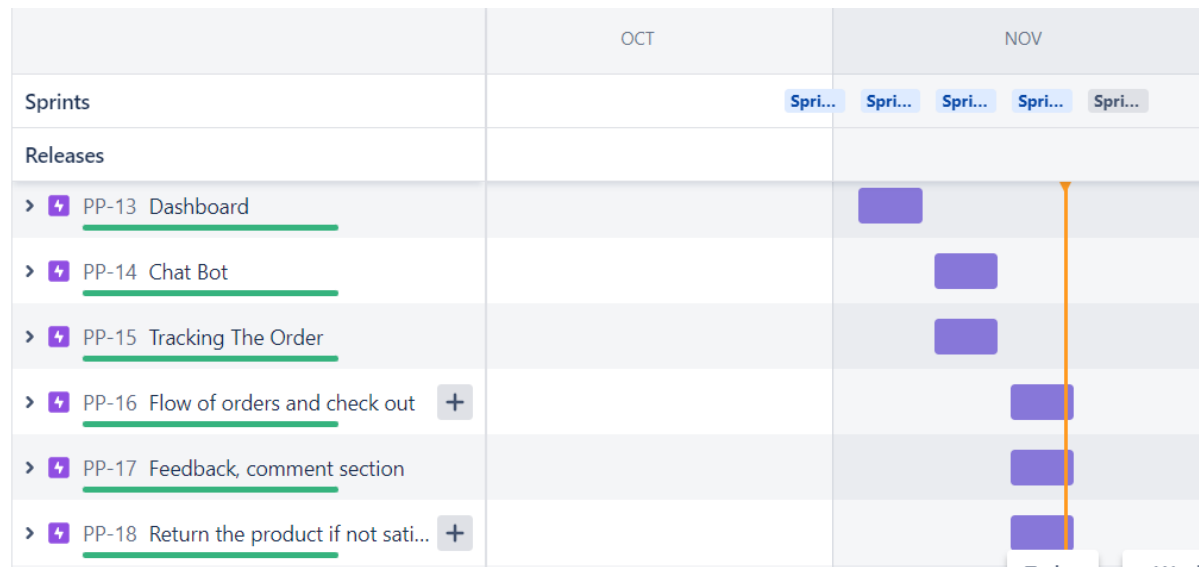
Sprint - 3: Average Velocity(AV)

$$Av = \text{Sprint Duration} / \text{Velocity} = 4/5 = 0.8$$

Sprint - 4: Average Velocity(AV)

$$Av = \text{Sprint Duration} / \text{Velocity} = 6/5 = 1.2$$

## ROADMAP



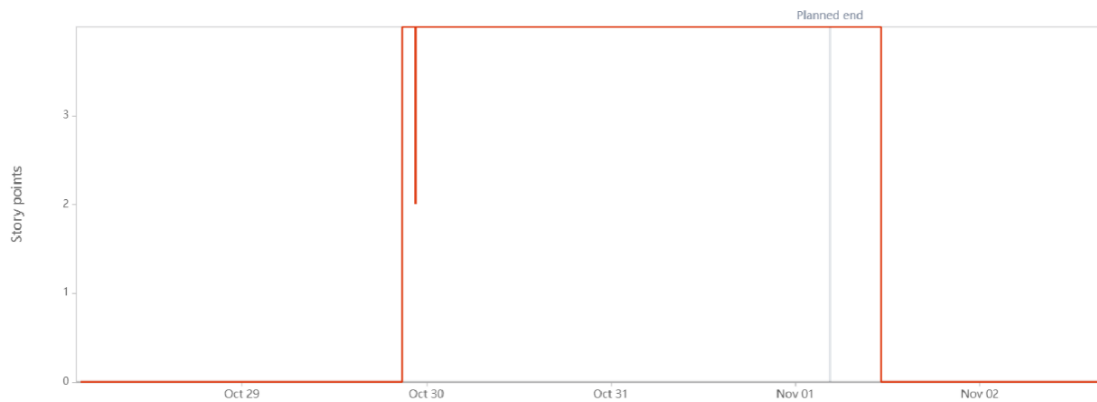
I have attached a roadmap that I made using JIRA software for the sprint and I have created story for those sprints.

### 6.3 Reports from JIRA

Jira Software is a member of a family of tools for managing projects in teams of all sizes. Jira was initially intended to be a bug and issue tracker. Today, however, Jira has developed into a potent work management solution for a variety of use cases, including agile software development and the management of requirements and test cases. In order to organise the work we are doing, we used the JIRA software to establish sprints on which we produced stories. We had kept track of it and had successfully completed all of the sprints on it.

#### Burndown Chart:

We have also created a burndown chart for the progress of the project. A burn down chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burn down charts can be applied to any project containing measurable progress over time.



## 7.Coding & Solutioning

### 7.1 Python code

```
from flask import Flask, render_template, flash, request, session
from flask import Flask, render_template, request, jsonify
import datetime
import re
```

```
import ibm_db
import pandas
import ibm_db_dbi
from sqlalchemy import create_engine
```

```
engine = create_engine('sqlite://',
                        echo = False)
```

```
dsn_hostname = "1bbf73c5-d84a-4bb0-85b9-ab1a4348f4a4.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud"
```

```
dsn_uid = "gqw63760"
```

```
dsn_pwd = "vV4w84opARRA2SxN"
```

```
dsn_driver = "{IBM DB2 ODBC DRIVER}"
```

```
dsn_database = "bludb"
```

```
dsn_port = "32286"
```

```
dsn_protocol = "TCPIP"
```

```
dsn_security = "SSL"
```

```
dsn = (
```

```
    "DRIVER={0};"
```

```
    "DATABASE={1};"
```

```
    "HOSTNAME={2};"
```

```
    "PORT={3};"
```

```
    "PROTOCOL={4};"
```

```
    "UID={5};"
```

```
    "PWD={6};"
```

```
    "SECURITY={7};").format(dsn_driver, dsn_database,  
dsn_hostname, dsn_port, dsn_protocol, dsn_uid,  
dsn_pwd,dsn_security)
```

```
try:
```

```
    conn = ibm_db.connect(dsn, "", "")
```

```
    print ("Connected to database: ", dsn_database, "as user: ",  
dsn_uid, "on host: ", dsn_hostname)
```

```
except:
```

```
    print ("Unable to connect: ", ibm_db.conn_errormsg() )
```

```
app = Flask(__name__)
app.config.from_object(__name__)
app.config['SECRET_KEY'] =
'7d441f27d441f27567d441f2b6176a'
```

```
@app.route("/")
def homepage():

    return render_template('index.html')
```

```
@app.route("/AdminLogin")
def AdminLogin():

    return render_template('AdminLogin.html')
```

```
@app.route("/NewUser")
def NewUser():

    return render_template('NewUser.html')
```

```
@app.route("/UserLogin")
def UserLogin():

    return render_template('UserLogin.html')
```

```

@app.route("/AdminHome")
def AdminHome():

    conn = ibm_db.connect(dsn, "", "")
    pd_conn = ibm_db_dbi.Connection(conn)

    selectQuery = "SELECT * from regtb "
    dataframe = pandas.read_sql(selectQuery, pd_conn)

    dataframe.to_sql('Employee_Data',
                    con=engine,
                    if_exists='append')

    # run a sql query
    data      =      engine.execute("SELECT      *      FROM
Employee_Data").fetchall()

    return render_template('AdminHome.html', data=data)

@app.route("/NewProduct")
def NewProduct():

    return render_template('NewProduct.html')

@app.route("/ProductInfo")
def ProductInfo():
    conn = ibm_db.connect(dsn, "", "")
    pd_conn = ibm_db_dbi.Connection(conn)

    selectQuery = "SELECT * from protb "

```

```

dataframe = pandas.read_sql(selectQuery, pd_conn)

dataframe.to_sql('Employee_Data',
                 con=engine,
                 if_exists='append')

# run a sql query
print(engine.execute("SELECT * FROM
Employee_Data").fetchall())

return render_template('ProductInfo.html',
data=engine.execute("SELECT * FROM Employee_Data").fetchall())

@app.route("/SalesInfo")
def SalesInfo():

    return render_template('SalesInfo.html')

@app.route("/Search")
def Search():

    conn = ibm_db.connect(dsn, "", "")
    pd_conn = ibm_db_dbi.Connection(conn)

    selectQuery = "SELECT * from protb "
    dataframe = pandas.read_sql(selectQuery, pd_conn)

```

```

        dataframe.to_sql('Employee_Data',
                        con=engine,
                        if_exists='append')

    # run a sql query
    print(engine.execute("SELECT * FROM
Employee_Data").fetchall())

    return render_template('ViewProduct.html',
data=engine.execute("SELECT * FROM Employee_Data").fetchall())

@app.route("/viewproduct", methods=['GET', 'POST'])
def viewproduct():

    searc = request.form['subcat']

    conn = ibm_db.connect(dsn, "", "")
    pd_conn = ibm_db_dbi.Connection(conn)

    selectQuery = "SELECT * from protb where SubCategory like
'" + searc + "'"
    dataframe = pandas.read_sql(selectQuery, pd_conn)

    dataframe.to_sql('Employee_Data',
                    con=engine,
                    if_exists='append')

    # run a sql query

```



```
print(engine.execute("SELECT * FROM Employee_Data").fetchall())
```

```
return render_template('ViewProduct.html',  
data=engine.execute("SELECT * FROM Employee_Data").fetchall())
```

```
@app.route("/RNewUser", methods=['GET', 'POST'])
```

```
def RNewUser():
```

```
    if request.method == 'POST':
```

```
        name1 = request.form['name']
```

```
        gender1 = request.form['gender']
```

```
        Age = request.form['age']
```

```
        email = request.form['email']
```

```
        address = request.form['address']
```

```
        pnumber = request.form['phone']
```

```
        uname = request.form['uname']
```

```
        password = request.form['psw']
```

```
        conn = ibm_db.connect(dsn, "", "")
```

```
        insertQuery = "INSERT INTO regtb VALUES ('" + name1  
+ "','" + gender1 + "','" + Age + "','" + email + "','" + pnumber + "','" +  
password + "','" + uname + "','" + address + "')
```

```
        insert_table = ibm_db.exec_immediate (conn, insertQuery)
```

```
        print(insert_table)
```

```
return render_template('userlogin.html')
```

```
@app.route("/RNewProduct", methods=['GET', 'POST'])
```

```
def RNewProduct():
```

```
    if request.method == 'POST':
```

```
        file = request.files['fileupload']
```

```
        file.save("static/upload/" + file.filename)
```

```
        ProductId =request.form['pid']
```

```
        Gender =request.form['gender']
```

```
        Category =request.form['cat']
```

```
        SubCategory=request.form['subcat']
```

```
        ProductType=request.form['ptype']
```

```
        Colour=request.form['color']
```

```
        Usage=request.form['usage']
```

```
        ProductTitle=request.form['ptitle']
```

```
        price = request.form['price']
```

```
        Image= file.filename
```

```
        ImageURL="static/upload/" + file.filename
```

```
conn = ibm_db.connect(dsn, "", "")
```

```
insertQuery = "INSERT INTO protb VALUES ('"+  
ProductId + "','" + Gender + "','" + Category + "','" + SubCategory + "','"  
+ ProductType + "','" + Colour + "','" + Usage + "','" + ProductTitle + "','" +  
Image + "','" + ImageURL + "','" + price + "')
```

```
insert_table = ibm_db.exec_immediate(conn, insertQuery)
```

```
data1 = 'Record Saved!'
```

```
return render_template('goback.html', data=data1)
```

```
@app.route("/userlogin", methods=['GET', 'POST'])
```

```
def userlogin():
```

```
    error = None
```

```
    if request.method == 'POST':
```

```
        username = request.form['uname']
```

```
        password = request.form['password']
```

```
        session['uname'] = request.form['uname']
```

```
conn = ibm_db.connect(dsn, "", "")
```

```
pd_conn = ibm_db_dbi.Connection(conn)
```

```
selectQuery = "SELECT * from regtb where uname='" +  
username + "' and password='" + password + "'"
```

```
dataframe = pandas.read_sql(selectQuery, pd_conn)
```

```
if dataframe.empty:
```

```
    data1 = 'Username or Password is wrong'
```

```
    return render_template('goback.html', data=data1)
```

```
else:
```

```
    print("Login")
```

```
selectQuery = "SELECT * from regtb where uname=" +  
username + " and password=" + password + ""  
dataframe = pandas.read_sql(selectQuery, pd_conn)
```

```
dataframe.to_sql('Employee_Data',  
con=engine,  
if_exists='append')
```

```
# run a sql query  
print(engine.execute("SELECT * FROM  
Employee_Data").fetchall())
```

```
return render_template('UserHome.html',  
data=engine.execute("SELECT * FROM Employee_Data").fetchall())
```

```
@app.route("/adminlogin", methods=['GET', 'POST'])  
def adminlogin():
```

```
error = None
```

```
if request.method == 'POST':
```

```
username = request.form['uname']
```

```
password = request.form['password']
```

```
conn = ibm_db.connect(dsn, "", "")
```

```
pd_conn = ibm_db_dbi.Connection(conn)
```

```
selectQuery = "SELECT * from adminth where username=" +  
+ username + " and password=" + password + ""
```

```

dataframe = pandas.read_sql(selectQuery, pd_conn)

if dataframe.empty:
    data1 = 'Username or Password is wrong'
    return render_template('goback.html', data=data1)
else:
    print("Login")
    selectQuery = "SELECT * from regtb "
    dataframe = pandas.read_sql(selectQuery, pd_conn)

    dataframe.to_sql('Employee_Data',
con=engine,if_exists='append')

    # run a sql query
    print(engine.execute("SELECT          *          FROM
Employee_Data").fetchall())

    return          render_template('AdminHome.html',
data=engine.execute("SELECT * FROM Employee_Data").fetchall())


@app.route("/Remove", methods=['GET'])
def Remove():

    pid = request.args.get('id')
    conn = ibm_db.connect(dsn, "", "")
    pd_conn = ibm_db_dbi.Connection(conn)

    insertQuery = "Delete from protb where id='"+ pid +'"

```

```

insert_table = ibm_db.exec_immediate(conn, insertQuery)

selectQuery = "SELECT * from protb "
dataframe = pandas.read_sql(selectQuery, pd_conn)

dataframe.to_sql('Employee_Data',
                 con=engine,
                 if_exists='append')

# run a sql query
print(engine.execute("SELECT * FROM
Employee_Data").fetchall())

return render_template('ProductInfo.html',
data=engine.execute("SELECT * FROM Employee_Data").fetchall())

@app.route("/fullInfo")
def fullInfo():
    pid = request.args.get('pid')
    session['pid'] = pid

    conn = ibm_db.connect(dsn, "", "")
    pd_conn = ibm_db_dbi.Connection(conn)

    selectQuery = "SELECT * FROM protb where ProductId=" +
pid + " "
    dataframe = pandas.read_sql(selectQuery, pd_conn)

    dataframe.to_sql('Employee_Data',

```

```

        con=engine,
        if_exists='append')

    # run a sql query
    print(engine.execute("SELECT * FROM
Employee_Data").fetchall())

    return render_template('ProductFullInfo.html',
data=engine.execute("SELECT * FROM Employee_Data").fetchall())

@app.route("/Book", methods=['GET', 'POST'])
def Book():
    if request.method == 'POST':

        username = session['uname']
        pid = session['pid']

        qty = request.form['qty']

        ctype = request.form['ctype']
        cardno = request.form['cardno']
        cvno = request.form['cvno']

        Bookingid = "
        ProductName = "
        uname= username
        Mobile="

```

```
Email="
Qty = qty
Amount="
```

```
CardType = ctype
CardNo = cardno
CvNo = cvno
date = datetime.datetime.now().strftime('%d-%b-%Y')
```

```
conn = ibm_db.connect(dsn, "", "")
pd_conn = ibm_db_dbi.Connection(conn)
selectQuery = "SELECT * FROM protb where ProductId="
+ pid + " "
dataframe = pandas.read_sql(selectQuery, pd_conn)
```

```
dataframe.to_sql('Employee_Data',con=engine,if_exists='append')
data = engine.execute("SELECT * FROM
Employee_Data").fetchall()
for item in data:
    ProductName = item[8]
    price = item[11]
    print(price)
    Amount = float(price) * float(Qty)

    print(Amount)
```

```
selectQuery1 = "SELECT * FROM regtb where uname="
+ username + ""
dataframe = pandas.read_sql(selectQuery1, pd_conn)
```



```
dataframe.to_sql('regtb', con=engine, if_exists='append')
data1 = engine.execute("SELECT * FROM
regtb").fetchall()
```

```
for item1 in data1:
    Mobile = item1[5]
    Email = item1[4]
```

```
selectQuery = "SELECT * FROM booktb"
dataframe = pandas.read_sql(selectQuery, pd_conn)
```

```
dataframe.to_sql('booktb', con=engine, if_exists='append')
data2 = engine.execute("SELECT * FROM
booktb").fetchall()
count = 0
```

```
for item in data2:
    count+=1
```

```
Bookingid="BOOKID00" + str(count)
```

```
insertQuery = "INSERT INTO booktb VALUES ('" +
Bookingid + "','" + ProductName + "','" + price + "','" + uname + "','" +
Mobile + "','" + Email + "','" + str(Qty) + "','" + str(Amount) + "','" +
str(CardType) + "','" + str(CardNo) + "','" + str(CvNo) + "','" + str(date)
+ "')
```

```
insert_table = ibm_db.exec_immediate(conn, insertQuery)
```

```
sendmsg(Email,"order received delivery in one week ")
```

```
selectQuery = "SELECT * FROM booktb where uname= "
+ uname + " "
```

```
dataframe = pandas.read_sql(selectQuery, pd_conn)
```

```
dataframe.to_sql('booktb1', con=engine, if_exists='append')
data      =      engine.execute("SELECT      *      FROM
booktb1").fetchall()
```

```
return render_template('UOrderInfo.html', data=data)
```

```
def sendmsg(Mailid,message):
```

```
    import smtplib
```

```
    from email.mime.multipart import MIMEMultipart
```

```
    from email.mime.text import MIMEText
```

```
    from email.mime.base import MIMEBase
```

```
    from email import encoders
```

```
    fromaddr = "sampletest685@gmail.com"
```

```
    toaddr = Mailid
```

```
    # instance of MIMEMultipart
```

```
msg = MIMEMultipart()

# storing the senders email address
msg['From'] = fromaddr

# storing the receivers email address
msg['To'] = toaddr

# storing the subject
msg['Subject'] = "Alert"

# string to store the body of the mail
body = message

# attach the body with the msg instance
msg.attach(MIMEText(body, 'plain'))

# creates SMTP session
s = smtplib.SMTP('smtp.gmail.com', 587)

# start TLS for security
s.starttls()

# Authentication
s.login(fromaddr, "hneucvnontsuwgpj")

# Converts the Multipart msg into a string
text = msg.as_string()

# sending the mail
s.sendmail(fromaddr, toaddr, text)

# terminating the session
```

```
s.quit()
```

```
@app.route("/UOrderInfo")
```

```
def UOrderInfo():
```

```
    uname = session['uname']
```

```
    conn = ibm_db.connect(dsn, "", "")
```

```
    pd_conn = ibm_db_dbi.Connection(conn)
```

```
    selectQuery = "SELECT * FROM booktb where uname= " +  
uname + " "
```

```
    dataframe = pandas.read_sql(selectQuery, pd_conn)
```

```
    dataframe.to_sql('booktb1', con=engine, if_exists='append')
```

```
    data = engine.execute("SELECT * FROM booktb1").fetchall()
```

```
    return render_template('UOrderInfo.html', data=data)
```

```
@app.route("/UserHome")
```

```
def UserHome():
```

```
    uname = session['uname']
```

```
    conn = ibm_db.connect(dsn, "", "")
```

```
    pd_conn = ibm_db_dbi.Connection(conn)
```

```
    selectQuery = "SELECT * FROM regtb where uname= " +  
uname + " "
```

```
    dataframe = pandas.read_sql(selectQuery, pd_conn)
```

```
    dataframe.to_sql('booktb1', con=engine, if_exists='append')
```

```
    data = engine.execute("SELECT * FROM booktb1").fetchall()
```

```

return render_template('UserHome.html', data=data)

@app.route("/ASalesInfo")
def ASalesInfo():
    conn = ibm_db.connect(dsn, "", "")
    pd_conn = ibm_db_dbi.Connection(conn)
    selectQuery = "SELECT * FROM booktb "
    dataframe = pandas.read_sql(selectQuery, pd_conn)
    dataframe.to_sql('booktb', con=engine, if_exists='append')
    data = engine.execute("SELECT * FROM booktb").fetchall()

    return render_template('ASalesInfo.html', data=data)

def main():
    app.run(debug=True, use_reloader=True)

if __name__ == '__main__':
    main()

```

## 7.2 Html coding

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Aroma Shop - Home</title>
    <link rel="icon" href="static/img/Favicon.png" type="image/png">
    <link rel="stylesheet" href="static/vendors/bootstrap/bootstrap.min.css">
    <link rel="stylesheet" href="static/vendors/fontawesome/css/all.min.css">

```

```

        <link rel="stylesheet" href="static/vendors/themify-icons/themify-icons.css">
        <link rel="stylesheet" href="static/vendors/nice-select/nice-select.css">
        <link rel="stylesheet" href="static/vendors/owl-carousel/owl.theme.default.min.css">
        <link rel="stylesheet" href="static/vendors/owl-carousel/owl.carousel.min.css">

        <link rel="stylesheet" href="static/css/style.css">
    </head>
    <body>
        <!--===== Start Header Menu Area =====-->
        <header class="header_area">
            <div class="main_menu">
                <nav class="navbar navbar-expand-lg navbar-light">
                    <div class="container">
                        <a class="navbar-brand logo_h" href="index.html"></a>
                        <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarSupportedContent"
                        aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle
navigation">
                            <span class="icon-bar"></span>
                            <span class="icon-bar"></span>
                            <span class="icon-bar"></span>
                        </button>
                        <div class="collapse navbar-collapse offset" id="navbarSupportedContent">
                            <ul class="nav navbar-nav menu_nav ml-auto mr-auto">
                                <li class="nav-item active"><a class="nav-link" href="index.html">Home</a></li>
                                <li class="nav-item submenu dropdown">
                                    <a href="#" class="nav-link dropdown-toggle" data-toggle="dropdown"
role="button" aria-haspopup="true"
                                    aria-expanded="false">Shop</a>
                                    <ul class="dropdown-menu">
                                        <li class="nav-item"><a class="nav-link" href="category.html">Shop
Category</a></li>
                                        <li class="nav-item"><a class="nav-link" href="single-product.html">Product
Details</a></li>
                                        <li class="nav-item"><a class="nav-link" href="checkout.html">Product
Checkout</a></li>
                                        <li class="nav-item"><a class="nav-link"
href="confirmation.html">Confirmation</a></li>
                                        <li class="nav-item"><a class="nav-link" href="cart.html">Shopping
Cart</a></li>
                                    </ul>
                                </li>
                                <li class="nav-item submenu dropdown">
                                    <a href="#" class="nav-link dropdown-toggle" data-toggle="dropdown"
role="button" aria-haspopup="true"
                                    aria-expanded="false">Blog</a>
                                    <ul class="dropdown-menu">
                                        <li class="nav-item"><a class="nav-link" href="blog.html">Blog</a></li>

```

```

        <li class="nav-item"><a class="nav-link" href="single-blog.html">Blog
Details</a></li>
    </ul>

    </li>
    <li class="nav-item submenu
dropdown">
        <a href="#" class="nav-link dropdown-toggle" data-toggle="dropdown"
role="button" aria-haspopup="true"
aria-expanded="false">Pages</a>
        <ul class="dropdown-menu">
            <li class="nav-item"><a class="nav-link" href="sign.html">Login</a></li>
            <li class="nav-item"><a class="nav-link" href="/register">Register</a></li>
            <li class="nav-item"><a class="nav-link" href="tracking-
order.html">Tracking</a></li>
        </ul>
    </li>
    <li class="nav-item"><a class="nav-link" href="contact.html">Contact</a></li>
</ul>

    <ul class="nav-shop">
        <li class="nav-item"><button><i class="ti-search"></i></button></li>
        <li class="nav-item"><button><i class="ti-shopping-cart"></i><span class="nav-
shop__circle">3</span></button> </li>
        <li class="nav-item"><a class="button button-header" href="#">Buy Now</a></li>
    </ul>
</div>
</div>
</nav>
</div>
</header>
    <!--===== End Header Menu Area =====>

<main class="site-main">

    <!--===== Hero banner start =====>
    <section class="hero-banner">
        <div class="container">
            <div class="row no-gutters align-items-center pt-60px">
                <div class="col-5 d-none d-sm-block">
                    <div class="hero-banner__img">
                        
                    </div>
                </div>
                <div class="col-sm-7 col-lg-6 offset-lg-1 pl-4 pl-md-5 pl-lg-0">
                    <div class="hero-banner__content">
                        <h4>Shop is fun</h4>
                        <h1>Browse Our Premium Product</h1>
                        <p>Us which over of signs divide dominion deep fill bring they're meat beho upon
own earth without morning over third. Their male dry. They are great appear whose land fly
grass.</p>

```

```

        <a class="button button-hero" href="#">Browse Now</a>
    </div>
</div>
</div>
</div>
</section>
<!--===== Hero banner start =====-->

<!--===== Hero Carousel start =====-->
<section class="section-margin mt-0">
    <div class="owl-carousel owl-theme hero-carousel">
        <div class="hero-carousel__slide">
            
            <a href="#" class="hero-carousel__slideOverlay">
                <h3>Wireless Headphone</h3>
                <p>Accessories Item</p>
            </a>
        </div>
        <div class="hero-carousel__slide">
            
            <a href="#" class="hero-carousel__slideOverlay">
                <h3>Wireless Headphone</h3>
                <p>Accessories Item</p>
            </a>
        </div>
        <div class="hero-carousel__slide">
            
            <a href="#" class="hero-carousel__slideOverlay">
                <h3>Wireless Headphone</h3>
                <p>Accessories Item</p>
            </a>
        </div>
    </div>
</section>
<!--===== Hero Carousel end =====-->

<!-- ===== trending product section start ===== -->
<section class="section-margin calc-60px">
    <div class="container">
        <div class="section-intro pb-60px">
            <p>Popular Item in the market</p>
            <h2>Trending <span class="section-intro__style">Product</span></h2>
        </div>
        <div class="row">
            <div class="col-md-6 col-lg-4 col-xl-3">
                <div class="card text-center card-product">
                    <div class="card-product__img">
                        
                    </div>
                    <ul class="card-product__imgOverlay">
                        <li><button><i class="ti-search"></i></button></li>
                    </ul>
                </div>
            </div>
        </div>
    </div>
</section>

```



```

        <li><button><i class="ti-shopping-cart"></i></button></li>
        <li><button><i class="ti-heart"></i></button></li>
    </ul>
</div>
<div class="card-body">
    <p>Accessories</p>
    <h4 class="card-product__title"><a href="single-product.html">Quartz Belt
Watch</a></h4>
    <p class="card-product__price">$150.00</p>
</div>
</div>
</div>
<div class="col-md-6 col-lg-4 col-xl-3">
    <div class="card text-center card-product">
        <div class="card-product__img">
            
            <ul class="card-product__imgOverlay">
                <li><button><i class="ti-search"></i></button></li>
                <li><button><i class="ti-shopping-cart"></i></button></li>
                <li><button><i class="ti-heart"></i></button></li>
            </ul>
        </div>
        <div class="card-body">
            <p>Beauty</p>
            <h4 class="card-product__title"><a href="single-product.html">Women
Freshwash</a></h4>
            <p class="card-product__price">$150.00</p>
        </div>
    </div>
</div>
</div>
<div class="col-md-6 col-lg-4 col-xl-3">
    <div class="card text-center card-product">
        <div class="card-product__img">
            
            <ul class="card-product__imgOverlay">
                <li><button><i class="ti-search"></i></button></li>
                <li><button><i class="ti-shopping-cart"></i></button></li>
                <li><button><i class="ti-heart"></i></button></li>
            </ul>
        </div>
        <div class="card-body">
            <p>Decor</p>
            <h4 class="card-product__title"><a href="single-product.html">Room Flash
Light</a></h4>
            <p class="card-product__price">$150.00</p>
        </div>
    </div>
</div>
</div>
<div class="col-md-6 col-lg-4 col-xl-3">
    <div class="card text-center card-product">

```

```

<div class="card-product__img">
  
  <ul class="card-product__imgOverlay">
    <li><button><i class="ti-search"></i></button></li>
    <li><button><i class="ti-shopping-cart"></i></button></li>
    <li><button><i class="ti-heart"></i></button></li>
  </ul>
</div>
<div class="card-body">
  <p>Decor</p>
  <h4 class="card-product__title"><a href="single-product.html">Room Flash
Light</a></h4>
  <p class="card-product__price">$150.00</p>
</div>
</div>
</div>
<div class="col-md-6 col-lg-4 col-xl-3">
  <div class="card text-center card-product">
    <div class="card-product__img">
      
      <ul class="card-product__imgOverlay">
        <li><button><i class="ti-search"></i></button></li>
        <li><button><i class="ti-shopping-cart"></i></button></li>
        <li><button><i class="ti-heart"></i></button></li>
      </ul>
    </div>
    <div class="card-body">
      <p>Accessories</p>
      <h4 class="card-product__title"><a href="single-product.html">Man Office
Bag</a></h4>
      <p class="card-product__price">$150.00</p>
    </div>
  </div>
</div>
<div class="col-md-6 col-lg-4 col-xl-3">
  <div class="card text-center card-product">
    <div class="card-product__img">
      
      <ul class="card-product__imgOverlay">
        <li><button><i class="ti-search"></i></button></li>
        <li><button><i class="ti-shopping-cart"></i></button></li>
        <li><button><i class="ti-heart"></i></button></li>
      </ul>
    </div>
    <div class="card-body">
      <p>Kids Toy</p>
      <h4 class="card-product__title"><a href="single-product.html">Charging
Car</a></h4>
      <p class="card-product__price">$150.00</p>
    </div>
  </div>
</div>

```

```

    </div>
</div>
<div class="col-md-6 col-lg-4 col-xl-3">
    <div class="card text-center card-product">
        <div class="card-product__img">
            
            <ul class="card-product__imgOverlay">
                <li><button><i class="ti-search"></i></button></li>
                <li><button><i class="ti-shopping-cart"></i></button></li>
                <li><button><i class="ti-heart"></i></button></li>
            </ul>
        </div>
        <div class="card-body">
            <p>Accessories</p>
            <h4 class="card-product__title"><a href="single-product.html">Bluetooth
Speaker</a></h4>
            <p class="card-product__price">$150.00</p>
        </div>
    </div>
</div>
<div class="col-md-6 col-lg-4 col-xl-3">
    <div class="card text-center card-product">
        <div class="card-product__img">
            
            <ul class="card-product__imgOverlay">
                <li><button><i class="ti-search"></i></button></li>
                <li><button><i class="ti-shopping-cart"></i></button></li>
                <li><button><i class="ti-heart"></i></button></li>
            </ul>
        </div>
        <div class="card-body">
            <p>Kids Toy</p>
            <h4 class="card-product__title"><a href="#">Charging Car</a></h4>
            <p class="card-product__price">$150.00</p>
        </div>
    </div>
</div>
</div>
</section>
<!-- ===== trending product section end ===== -->

<!-- ===== offer section start ===== -->
<section class="offer" id="parallax-1" data-anchor-target="#parallax-1" data-300-
top="background-position: 20px 30px" data-top-bottom="background-position: 0 20px">
    <div class="container">
        <div class="row">
            <div class="col-xl-5">
                <div class="offer__content text-center">

```

```

        <h3>Up To 50% Off</h3>
        <h4>Winter Sale</h4>
        <p>Him she'd let them sixth saw light</p>
        <a class="button button--active mt-3 mt-xl-4" href="#">Shop Now</a>
    </div>
</div>
</div>
</div>
</section>
<!-- ===== offer section end ===== -->

<!-- ===== Best Selling item carousel ===== -->
<section class="section-margin calc-60px">
    <div class="container">
        <div class="section-intro pb-60px">
            <p>Popular Item in the market</p>
            <h2>Best <span class="section-intro__style">Sellers</span></h2>
        </div>
        <div class="owl-carousel owl-theme" id="bestSellerCarousel">
            <div class="card text-center card-product">
                <div class="card-product__img">
                    
                    <ul class="card-product__imgOverlay">
                        <li><button><i class="ti-search"></i></button></li>
                        <li><button><i class="ti-shopping-cart"></i></button></li>
                        <li><button><i class="ti-heart"></i></button></li>
                    </ul>
                </div>
                <div class="card-body">
                    <p>Accessories</p>
                    <h4 class="card-product__title"><a href="single-product.html">Quartz Belt
Watch</a></h4>
                    <p class="card-product__price">$150.00</p>
                </div>
            </div>

            <div class="card text-center card-product">
                <div class="card-product__img">
                    
                    <ul class="card-product__imgOverlay">
                        <li><button><i class="ti-search"></i></button></li>
                        <li><button><i class="ti-shopping-cart"></i></button></li>
                        <li><button><i class="ti-heart"></i></button></li>
                    </ul>
                </div>
                <div class="card-body">
                    <p>Beauty</p>
                    <h4 class="card-product__title"><a href="single-product.html">Women
Freshwash</a></h4>
                    <p class="card-product__price">$150.00</p>
                </div>
            </div>
        </div>
    </div>

```

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

```
<div class="card text-center card-product">
  <div class="card-product__img">
    
    <ul class="card-product__imgOverlay">
      <li><button><i class="ti-search"></i></button></li>
      <li><button><i class="ti-shopping-cart"></i></button></li>
      <li><button><i class="ti-heart"></i></button></li>
    </ul>
  </div>
  <div class="card-body">
    <p>Decor</p>
    <h4 class="card-product__title"><a href="single-product.html">Room Flash
Light</a></h4>
    <p class="card-product__price">$150.00</p>
  </div>
</div>
```

```
<div class="card text-center card-product">
  <div class="card-product__img">
    
    <ul class="card-product__imgOverlay">
      <li><button><i class="ti-search"></i></button></li>
      <li><button><i class="ti-shopping-cart"></i></button></li>
      <li><button><i class="ti-heart"></i></button></li>
    </ul>
  </div>
  <div class="card-body">
    <p>Decor</p>
    <h4 class="card-product__title"><a href="single-product.html">Room Flash
Light</a></h4>
    <p class="card-product__price">$150.00</p>
  </div>
</div>
```

```
<div class="card text-center card-product">
  <div class="card-product__img">
    
    <ul class="card-product__imgOverlay">
      <li><button><i class="ti-search"></i></button></li>
      <li><button><i class="ti-shopping-cart"></i></button></li>
      <li><button><i class="ti-heart"></i></button></li>
    </ul>
  </div>
  <div class="card-body">
    <p>Accessories</p>
    <h4 class="card-product__title"><a href="single-product.html">Quartz Belt
Watch</a></h4>
```

```
<p class="card-product__price">$150.00</p>
</div>
</div>
```

```
<div class="card text-center card-product">
  <div class="card-product__img">
    
    <ul class="card-product__imgOverlay">
      <li><button><i class="ti-search"></i></button></li>
      <li><button><i class="ti-shopping-cart"></i></button></li>
      <li><button><i class="ti-heart"></i></button></li>
    </ul>
  </div>
  <div class="card-body">
    <p>Beauty</p>
    <h4 class="card-product__title"><a href="single-product.html">Women
Freshwash</a></h4>
    <p class="card-product__price">$150.00</p>
  </div>
</div>

<div class="card text-center card-product">
  <div class="card-product__img">
    
    <ul class="card-product__imgOverlay">
      <li><button><i class="ti-search"></i></button></li>
      <li><button><i class="ti-shopping-cart"></i></button></li>
      <li><button><i class="ti-heart"></i></button></li>
    </ul>
  </div>
  <div class="card-body">
    <p>Decor</p>
    <h4 class="card-product__title"><a href="single-product.html">Room Flash
Light</a></h4>
    <p class="card-product__price">$150.00</p>
  </div>
</div>

<div class="card text-center card-product">
  <div class="card-product__img">
    
    <ul class="card-product__imgOverlay">
      <li><button><i class="ti-search"></i></button></li>
      <li><button><i class="ti-shopping-cart"></i></button></li>
      <li><button><i class="ti-heart"></i></button></li>
    </ul>
  </div>
  <div class="card-body">
    <p>Decor</p>
```

```

        <h4 class="card-product__title"><a href="single-product.html">Room Flash
Light</a></h4>
        <p class="card-product__price">$150.00</p>
    </div>
</div>
</div>
</div>
</div>
</section>
<!-- ===== Best Selling item carousel end ===== -->

<!-- ===== Blog section start ===== -->
<section class="blog">
    <div class="container">
        <div class="section-intro pb-60px">
            <p>Popular Item in the market</p>
            <h2>Latest <span class="section-intro__style">News</span></h2>
        </div>

        <div class="row">
            <div class="col-md-6 col-lg-4 mb-4 mb-lg-0">
                <div class="card card-blog">
                    <div class="card-blog__img">
                        
                    </div>
                    <div class="card-body">
                        <ul class="card-blog__info">
                            <li><a href="#">By Admin</a></li>
                            <li><a href="#"><i class="ti-comments-smiley"></i> 2 Comments</a></li>
                        </ul>
                        <h4 class="card-blog__title"><a href="single-blog.html">The Richland Center
Shooping News and weekly shooper</a></h4>
                        <p>Let one fifth i bring fly to divided face for bearing divide unto seed. Winged
divided light Forth.</p>
                        <a class="card-blog__link" href="#">Read More <i class="ti-arrow-
right"></i></a>
                    </div>
                </div>

                <div class="col-md-6 col-lg-4 mb-4 mb-lg-0">
                    <div class="card card-blog">
                        <div class="card-blog__img">
                            
                        </div>
                        <div class="card-body">
                            <ul class="card-blog__info">
                                <li><a href="#">By Admin</a></li>
                                <li><a href="#"><i class="ti-comments-smiley"></i> 2 Comments</a></li>
                            </ul>

```

<h4 class="card-blog\_\_title"><a href="single-blog.html">The Shopping News also offers top-quality printing services</a></h4>

<p>Let one fifth i bring fly to divided face for bearing divide unto seed. Winged divided light Forth.</p>

<a class="card-blog\_\_link" href="#">Read More <i class="ti-arrow-right"></i></a>

</div>

</div>

</div>

<div class="col-md-6 col-lg-4 mb-4 mb-lg-0">

<div class="card card-blog">

<div class="card-blog\_\_img">



</div>

<div class="card-body">

<ul class="card-blog\_\_info">

<li><a href="#">By Admin</a></li>

<li><a href="#"><i class="ti-comments-smiley"></i> 2 Comments</a></li>

</ul>

<h4 class="card-blog\_\_title"><a href="single-blog.html">Professional design staff and efficient equipment you'll find we offer</a></h4>

<p>Let one fifth i bring fly to divided face for bearing divide unto seed. Winged divided light Forth.</p>

<a class="card-blog\_\_link" href="#">Read More <i class="ti-arrow-right"></i></a>

</div>

</div>

</div>

</div>

</div>

</section>

<!-- ===== Blog section end ===== -->

<!-- ===== Subscribe section start ===== -->

<section class="subscribe-position">

<div class="container">

<div class="subscribe text-center">

<h3 class="subscribe\_\_title">Get Update From Anywhere</h3>

<p>Bearing Void gathering light light his evening unto dont afraid</p>

<div id="mc\_embed\_signup">

<form target="\_blank" action="https://spondonit.us12.list-manage.com/subscribe/post?u=1462626880ade1ac87bd9c93a&id=92a4423d01" method="get" class="subscribe-form form-inline mt-5 pt-1">

<div class="form-group ml-sm-auto">

<input class="form-control mb-1" type="email" name="EMAIL" placeholder="Enter your email" onfocus="this.placeholder = ''" onblur="this.placeholder = 'Your Email Address '">

<div class="info"></div>

</div>



```
        <button class="button button-subscribe mr-auto mb-1" type="submit">Subscribe
Now</button>
```

```
        <div style="position: absolute; left: -5000px;">
            <input name="b_36c4fd991d266f23781ded980_aefe40901a" tabindex="-1"
value="" type="text">
        </div>
```

```
    </form>
</div>
```

```

</div>
</div>
</section>
```

```
<!-- ===== Subscribe section end ===== -->
```

```
</main>
```

```
<!--===== Start footer Area =====-->
```

```
    <footer class="footer">
        <div class="footer-area">
            <div class="container">
                <div class="row section_gap">
                    <div class="col-lg-3 col-md-6 col-sm-6">
                        <div class="single-footer-widget tp_widgets">
                            <h4 class="footer_title large_title">Our
Mission</h4>
```

```

                                <p>
                                    So seed seed green that winged
cattle in. Gathering thing made fly you're no
                                    divided deep moved us lan
Gathering thing us land years living.
```

```
                                </p>
                                <p>
                                    So seed seed green that winged
cattle in. Gathering thing made fly you're no divided deep moved
```

```
                                </p>
                            </div>
                        </div>
```

```
                    <div class="offset-lg-1 col-lg-2 col-md-6 col-sm-6">
                        <div class="single-footer-widget tp_widgets">
                            <h4 class="footer_title">Quick
```

```
Links</h4>
                                <ul class="list">
                                    <li><a href="#">Home</a></li>
                                    <li><a href="#">Shop</a></li>
                                    <li><a href="#">Blog</a></li>
```

[Product](#)


[Contact](#)

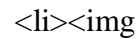
[Brand](#)

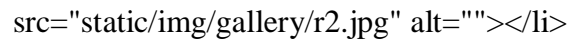
[Brand](#)

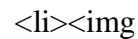
[Brand](#)

#### Gallery

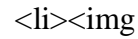


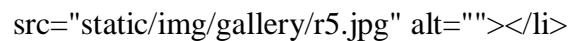


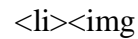


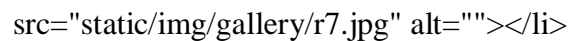


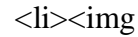




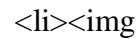












#### Contact

Us

location-arrow

Head Office

123, Main Street, Your

City

phone

Phone Number

+123 456 7890

+123 456 7890



The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

## 8.1 Test Case

S.NO	Action	TEST SCENARIO	TEST CASE	EXPECTED RESULT	ACTUAL RESULT	STATUS PASS/ FAIL
1.	Registration	Verify the email id is valid	Enter a valid email id with the domain name after @ symbol and the dot symbol after the domain name.	Successful Registration	Successful Registration	PASS
2.			Enter the email id without the domain name after @ symbol or the dot symbol after the domain name.	A message will be shown as "Enter a valid email id"		
3.	Login	Verify the user already exists	User data on the database already exists.	Successful Login	Successful Login	PASS
4.			User data on the database does not exist.	A message will be shown as "User does not exist"		
5.	Search	Verify whether the search option working correctly	Verify whether the search option is showing results for searched content.	Showing related results	Showing related results	PASS
6.	order	Verify order option is working correctly	Check on the data whether the product has been	Shows that the product has been ordered	Showing that product has been ordered	PASS

			ordered by the user			
7.	Review	I can review the product	Check the user can review the product again	Ordered products are shown		
8.	Send Confirmation	Check confirmation email sent by the website to the customer	Check whether customer receives email	Customer received email		
9.	Chatbot	Check whether the user is able to interact with the Watson assistant	Check whether the chatbot replies based on the question asked	Chatbot replies based on the question asked		

## Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed

## Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

## Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

## Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

## Features to be tested

- Verify that the entries are of the correct format

- No duplicate entries should be allowed
- All links should take the user to the correct page.

## **8.2 User Acceptance Testing**

User acceptability testing, also known as application testing or end-user testing, is a stage of software development when the target market tests the software in the real world. The final stage of software testing, known as UAT, is when the product is made available to the target market. UAT's objective is to make sure the programme meets development criteria and can handle tasks in the actual world.

### **Objectives:**

- All User stories are completed and signed off
- UI validations completed
- Completion of System and Integration Testing
- Ensured 100% Test Coverage in System Testing.
- Validations undertook against the Functional specifications.
- All environment and Access requests sorted out.

## **9.RESULTS**

### **9.1performance metrics**

## 9.1.1 Home Page



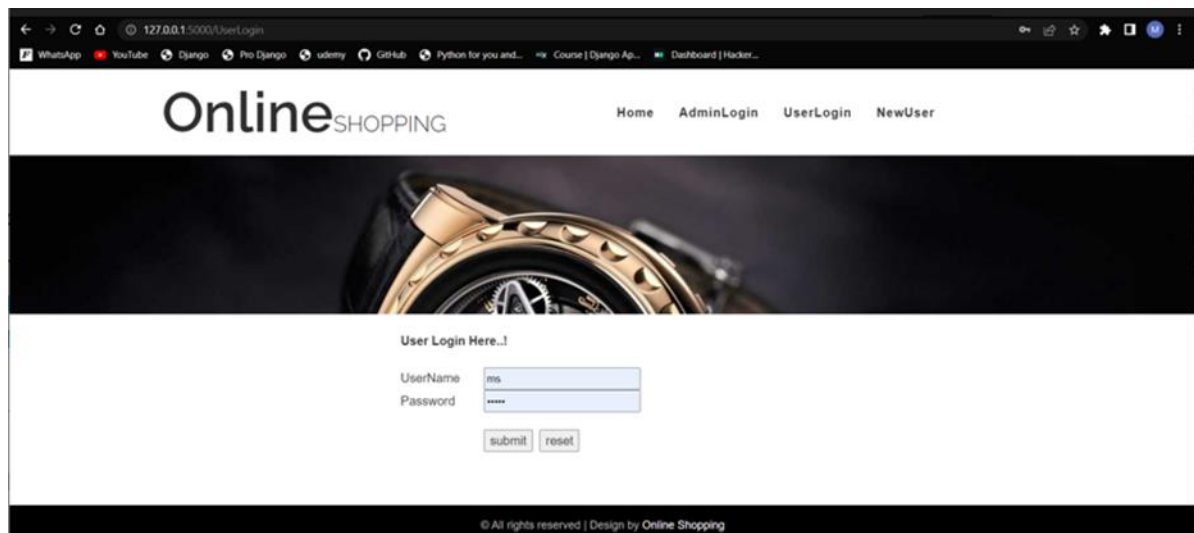
## 9.1.2 Register Page

A screenshot of a web browser displaying the 'New User Registration' page. The browser's address bar shows '127.0.0.1:5000/NewUser'. The page has a dark header image of a watch. The registration form includes the following fields and options:

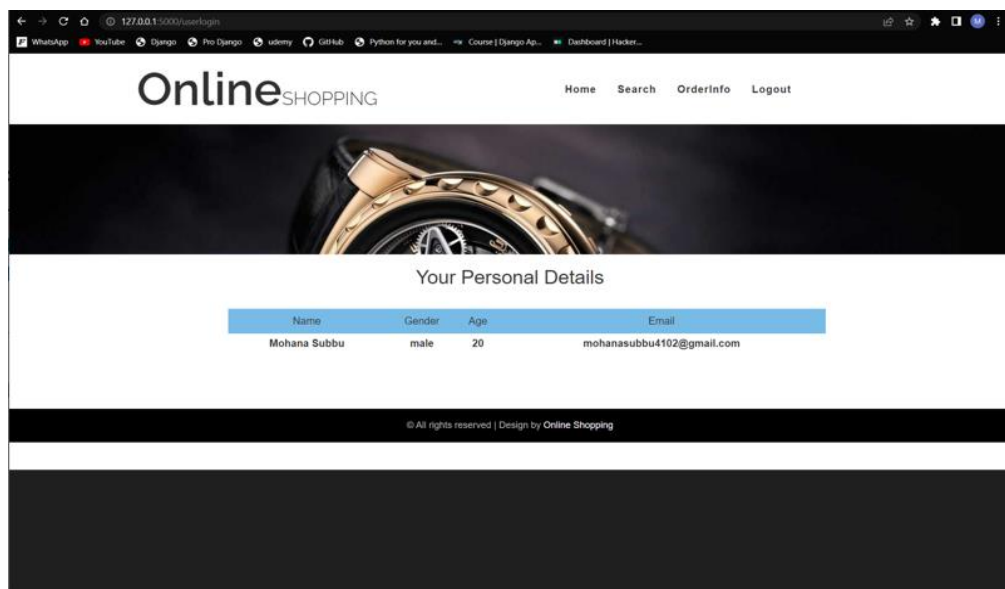
- Name: soundarya
- Gender: ☐ Male ☒ Female
- Age: 30
- Email Id: soundarya@gmail.com
- Phone Number: 7846132548
- Address: trichy
- User Name: sound
- Password: \*\*\*\*\*

At the bottom of the form are 'Submit' and 'Reset' buttons. A yellow cursor icon is pointing at the 'Submit' button. In the bottom right corner, there is a Windows watermark: 'Activate Windows Go to Settings to activate Windows.'

## 9.1.3 Login Page

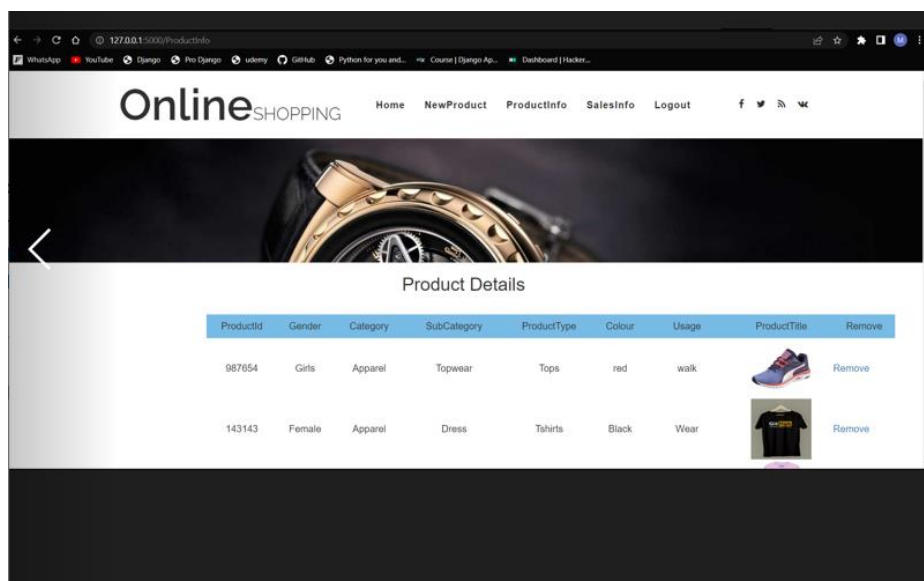
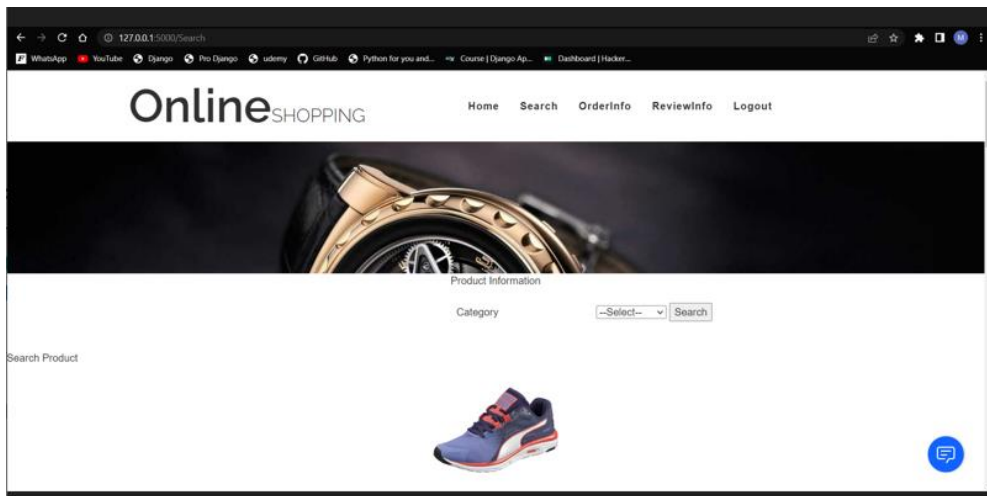


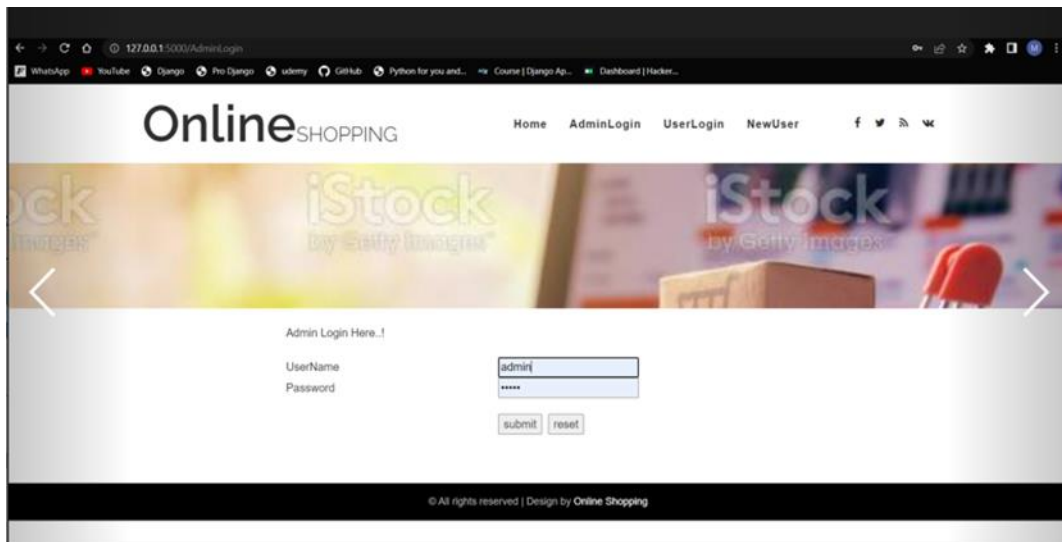
## 9.1.4 Users Details



## 9.1.2 Product search







## 10. Advantage & Disadvantages

1. Good customer satisfaction
2. The performance is better in terms of quality and time.
3. It provides the better use of the database which store user and product history.
4. Quality prediction, Scalability, Prediction, speed are the main advantages of the proposed scheme.

## 11. Conclusions

In we developed product recommendation system using chat bot-based applications we developed here recommend the product list , Recommender Systems have been widely used to exhibit the most appropriate items to users given their past consumption preferences. Recommendation systems are achieving great success in e-Commerce applications, during a live interaction with a customer; recommendation system may apply different techniques to solve the problem of making a correct and relevant product recommendation using chat bot Flask framework designed here for product recommendation system

## 12. Future work

In future work we designed the android based mobile applications for product recommendation .

## 13.Appendix

### Source code:

Python code:

```
from flask import Flask, render_template, flash, request,session
from flask import Flask, render_template, request, jsonify
import datetime
import re
```

```
import ibm_db
import pandas
import ibm_db_dbi
from sqlalchemy import create_engine
```

```
engine = create_engine('sqlite://',
                        echo = False)
```

```
dsn_hostname          =          "1bbf73c5-d84a-4bb0-85b9-
ab1a4348f4a4.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud"
dsn_uid = "gqw63760"
dsn_pwd = "vV4w84opARRA2SxN"
```

```
dsn_driver = "{IBM DB2 ODBC DRIVER}"
dsn_database = "bludb"
dsn_port = "32286"
dsn_protocol = "TCPIP"
dsn_security = "SSL"
```

```
dsn = (
```

```

"DRIVER={0};"
"DATABASE={1};"
"HOSTNAME={2};"
"PORT={3};"
"PROTOCOL={4};"
"UID={5};"
"PWD={6};"
"SECURITY={7};").format(dsn_driver,      dsn_database,
dsn_hostname,      dsn_port,      dsn_protocol,      dsn_uid,
dsn_pwd,dsn_security)

```

```

try:
    conn = ibm_db.connect(dsn, "", "")
    print ("Connected to database: ", dsn_database, "as user: ",
dsn_uid, "on host: ", dsn_hostname)

```

```

except:
    print ("Unable to connect: ", ibm_db.conn_errormsg() )

```

```

app = Flask(__name__)
app.config.from_object(__name__)
app.config['SECRET_KEY'] =
'7d441f27d441f27567d441f2b6176a'

```

```

@app.route("/")
def homepage():

```

```

    return render_template('index.html')

```

```
@app.route("/AdminLogin")
def AdminLogin():

    return render_template('AdminLogin.html')
```

```
@app.route("/NewUser")
def NewUser():

    return render_template('NewUser.html')
@app.route("/UserLogin")
def UserLogin():
```

```
    return render_template('UserLogin.html')
```

```
@app.route("/AdminHome")
def AdminHome():
```

```
    conn = ibm_db.connect(dsn, "", "")
    pd_conn = ibm_db_dbi.Connection(conn)
```

```
    selectQuery = "SELECT * from regtb "
    dataframe = pandas.read_sql(selectQuery, pd_conn)
```

```
    dataframe.to_sql('Employee_Data',
```

```

        con=engine,
        if_exists='append')

    # run a sql query
    data = engine.execute("SELECT * FROM
Employee_Data").fetchall()

    return render_template('AdminHome.html', data=data)

@app.route("/NewProduct")
def NewProduct():

    return render_template('NewProduct.html')

@app.route("/ProductInfo")
def ProductInfo():
    conn = ibm_db.connect(dsn, "", "")
    pd_conn = ibm_db_dbi.Connection(conn)

    selectQuery = "SELECT * from protb "
    dataframe = pandas.read_sql(selectQuery, pd_conn)

    dataframe.to_sql('Employee_Data',
        con=engine,
        if_exists='append')

    # run a sql query
    print(engine.execute("SELECT * FROM
Employee_Data").fetchall())

    return render_template('ProductInfo.html',
data=engine.execute("SELECT * FROM Employee_Data").fetchall())

```

```
@app.route("/SalesInfo")
def SalesInfo():

    return render_template('SalesInfo.html')
```

```
@app.route("/Search")
def Search():

    conn = ibm_db.connect(dsn, "", "")
    pd_conn = ibm_db_dbi.Connection(conn)

    selectQuery = "SELECT * from protb "
    dataframe = pandas.read_sql(selectQuery, pd_conn)

    dataframe.to_sql('Employee_Data',
                    con=engine,
                    if_exists='append')

    # run a sql query
    print(engine.execute("SELECT * FROM
Employee_Data").fetchall())

    return render_template('ViewProduct.html',
data=engine.execute("SELECT * FROM Employee_Data").fetchall())
```

```

@app.route("/viewproduct", methods=['GET', 'POST'])
def viewproduct():

    searc = request.form['subcat']

    conn = ibm_db.connect(dsn, "", "")
    pd_conn = ibm_db_dbi.Connection(conn)

    selectQuery = "SELECT * from protb where SubCategory like
'" + searc + "%' "
    dataframe = pandas.read_sql(selectQuery, pd_conn)

    dataframe.to_sql('Employee_Data',
                    con=engine,
                    if_exists='append')

    # run a sql query
    print(engine.execute("SELECT * FROM
Employee_Data").fetchall())

    return render_template('ViewProduct.html',
data=engine.execute("SELECT * FROM Employee_Data").fetchall())

@app.route("/RNewUser", methods=['GET', 'POST'])
def RNewUser():
    if request.method == 'POST':

```



```
name1 = request.form['name']
gender1 = request.form['gender']
Age = request.form['age']
email = request.form['email']
address = request.form['address']
pnumber = request.form['phone']
uname = request.form['uname']
password = request.form['psw']
```

```
conn = ibm_db.connect(dsn, "", "")
```

```
insertQuery = "INSERT INTO regtb VALUES ('" + name1
+ "','" + gender1 + "','" + Age + "','" + email + "','" + pnumber + "','" +
password + "','" + uname + "','" + address + "')
```

```
insert_table = ibm_db.exec_immediate (conn, insertQuery)
print(insert_table)
```

```
return render_template('userlogin.html')
```

```
@app.route("/RNewProduct", methods=['GET', 'POST'])
```

```
def RNewProduct():
```

```
    if request.method == 'POST':
```

```
        file = request.files['fileupload']
        file.save("static/upload/" + file.filename)
```

```
ProductId =request.form['pid']
Gender =request.form['gender']
Category =request.form['cat']
SubCategory=request.form['subcat']
ProductType=request.form['ptype']
Colour=request.form['color']
Usage=request.form['usage']
ProductTitle=request.form['ptitle']
price = request.form['price']
Image= file.filename
ImageURL="static/upload/" + file.filename
```

```
conn = ibm_db.connect(dsn, "", "")
```

```
insertQuery = "INSERT INTO protb VALUES ('"+
ProductId + "','" + Gender + "','" + Category + "','" + SubCategory + "','"
+ ProductType + "','" + Colour + "','" + Usage + "','" + ProductTitle + "','" +
Image + "','" + ImageURL + "','" + price + "')"
insert_table = ibm_db.exec_immediate(conn, insertQuery)
```

```
data1 = 'Record Saved!'
return render_template('goback.html', data=data1)
```

```
@app.route("/userlogin", methods=['GET', 'POST'])
```

```

def userlogin():
    error = None
    if request.method == 'POST':
        username = request.form['uname']
        password = request.form['password']
        session['uname'] = request.form['uname']

        conn = ibm_db.connect(dsn, "", "")
        pd_conn = ibm_db_dbi.Connection(conn)

        selectQuery = "SELECT * from regtb where uname='" +
username + "' and password='" + password + "'"
        dataframe = pandas.read_sql(selectQuery, pd_conn)

        if dataframe.empty:
            data1 = 'Username or Password is wrong'
            return render_template('goback.html', data=data1)
        else:
            print("Login")
            selectQuery = "SELECT * from regtb where uname='" +
username + "' and password='" + password + "'"
            dataframe = pandas.read_sql(selectQuery, pd_conn)

            dataframe.to_sql('Employee_Data',
                            con=engine,
                            if_exists='append')

            # run a sql query
            print(engine.execute("SELECT * FROM
Employee_Data").fetchall())

```

```
        return render_template('UserHome.html',
data=engine.execute("SELECT * FROM Employee_Data").fetchall())
```

```
@app.route("/adminlogin", methods=['GET', 'POST'])
def adminlogin():
    error = None
    if request.method == 'POST':
        username = request.form['uname']
        password = request.form['password']

        conn = ibm_db.connect(dsn, "", "")
        pd_conn = ibm_db_dbi.Connection(conn)

        selectQuery = "SELECT * from admintb where username=" +
+ username + " and password=" + password + ""
        dataframe = pandas.read_sql(selectQuery, pd_conn)

        if dataframe.empty:
            data1 = 'Username or Password is wrong'
            return render_template('goback.html', data=data1)
        else:
            print("Login")
            selectQuery = "SELECT * from regtb "
            dataframe = pandas.read_sql(selectQuery, pd_conn)

            dataframe.to_sql('Employee_Data',
con=engine,if_exists='append')
```

```
        # run a sql query
        print(engine.execute("SELECT * FROM
Employee_Data").fetchall())
```

```
        return render_template('AdminHome.html',
data=engine.execute("SELECT * FROM Employee_Data").fetchall())
```

```
@app.route("/Remove", methods=['GET'])
def Remove():
```

```
    pid = request.args.get('id')
    conn = ibm_db.connect(dsn, "", "")
    pd_conn = ibm_db_dbi.Connection(conn)
```

```
    insertQuery = "Delete from protb where id='"+ pid + "'"
    insert_table = ibm_db.exec_immediate(conn, insertQuery)
```

```
    selectQuery = "SELECT * from protb "
    dataframe = pandas.read_sql(selectQuery, pd_conn)
```

```
    dataframe.to_sql('Employee_Data',
                    con=engine,
                    if_exists='append')
```

```
        # run a sql query
        print(engine.execute("SELECT * FROM
Employee_Data").fetchall())
```

```
        return render_template('ProductInfo.html',
data=engine.execute("SELECT * FROM Employee_Data").fetchall())
```

```
@app.route("/fullInfo")
```

```
def fullInfo():
```

```
    pid = request.args.get('pid')
```

```
    session['pid'] = pid
```

```
    conn = ibm_db.connect(dsn, "", "")
```

```
    pd_conn = ibm_db_dbi.Connection(conn)
```

```
    selectQuery = "SELECT * FROM protb where ProductId=" +
pid + " "
```

```
    dataframe = pandas.read_sql(selectQuery, pd_conn)
```

```
    dataframe.to_sql('Employee_Data',
```

```
        con=engine,
```

```
        if_exists='append')
```

```
    # run a sql query
```

```
    print(engine.execute("SELECT * FROM
Employee_Data").fetchall())
```

```
        return render_template('ProductFullInfo.html',
data=engine.execute("SELECT * FROM Employee_Data").fetchall())
```

```
@app.route("/Book", methods=['GET', 'POST'])
def Book():
    if request.method == 'POST':

        username = session['uname']
        pid = session['pid']

        qty = request.form['qty']

        ctype = request.form['ctype']
        cardno = request.form['cardno']
        cvno = request.form['cvno']

        Bookingid = "
        ProductName = "
        uname= username
        Mobile="
        Email="
        Qty = qty
        Amount="

        CardType = ctype
        CardNo = cardno
        CvNo = cvno
        date = datetime.datetime.now().strftime('%d-%b-%Y')

        conn = ibm_db.connect(dsn, "", "")
        pd_conn = ibm_db_dbi.Connection(conn)
```

```
selectQuery = "SELECT * FROM protb where ProductId="
+ pid + " "
dataframe = pandas.read_sql(selectQuery, pd_conn)
```

```
dataframe.to_sql('Employee_Data',con=engine,if_exists='append')
data = engine.execute("SELECT * FROM
Employee_Data").fetchall()
for item in data:
    ProductName = item[8]
    price = item[11]
    print(price)
    Amount = float(price) * float(Qty)

    print(Amount)
```

```
selectQuery1 = "SELECT * FROM regtb where uname="
+ username + ""
dataframe = pandas.read_sql(selectQuery1, pd_conn)
```

```
dataframe.to_sql('regtb', con=engine, if_exists='append')
data1 = engine.execute("SELECT * FROM
regtb").fetchall()
```

```
for item1 in data1:
    Mobile = item1[5]
    Email = item1[4]
```

```
selectQuery = "SELECT * FROM booktb"
dataframe = pandas.read_sql(selectQuery, pd_conn)
```



```
dataframe.to_sql('booktb', con=engine, if_exists='append')
data2 = engine.execute("SELECT * FROM
booktb").fetchall()
count = 0
```

```
for item in data2:
    count+=1
```

```
Bookingid="BOOKID00" + str(count)
```

```
insertQuery = "INSERT INTO booktb VALUES ('" +
Bookingid + "','" + ProductName + "','" + price + "','" + uname + "','" +
Mobile + "','" + Email + "','" + str(Qty) + "','" + str(Amount) + "','" +
str(CardType) + "','" + str(CardNo) + "','" + str(CvNo) + "','" + str(date)
+ "')
```

```
insert_table = ibm_db.exec_immediate(conn, insertQuery)
```

```
sendmsg(Email,"order received delivery in one week ")
```

```
selectQuery = "SELECT * FROM booktb where uname=" "
+ uname + " "
dataframe = pandas.read_sql(selectQuery, pd_conn)
```

```
dataframe.to_sql('booktb1', con=engine, if_exists='append')
data      =      engine.execute("SELECT      *      FROM
booktb1").fetchall()
```

```
return render_template('UOrderInfo.html', data=data)
```

```
def sendmsg(Mailid,message):
    import smtplib
    from email.mime.multipart import MIMEMultipart
    from email.mime.text import MIMEText
    from email.mime.base import MIMEBase
    from email import encoders
```

```
fromaddr = "sampletest685@gmail.com"
toaddr = Mailid
```

```
# instance of MIMEMultipart
msg = MIMEMultipart()
```

```
# storing the senders email address
msg['From'] = fromaddr
```

```
# storing the receivers email address
msg['To'] = toaddr
```

```
# storing the subject
msg['Subject'] = "Alert"
```

```
# string to store the body of the mail
body = message
```

```
# attach the body with the msg instance
msg.attach(MIMEText(body, 'plain'))
```

```
# creates SMTP session
s = smtplib.SMTP('smtp.gmail.com', 587)
```

```
# start TLS for security
s.starttls()
```

```
# Authentication
s.login(fromaddr, "hneucvnontsuwgpj")
```

```
# Converts the Multipart msg into a string
text = msg.as_string()
```

```
# sending the mail
s.sendmail(fromaddr, toaddr, text)
```

```
# terminating the session
s.quit()
```

```
@app.route("/UOrderInfo")
def UOrderInfo():
```

```
    uname = session['uname']
```

```
    conn = ibm_db.connect(dsn, "", "")
```

```
    pd_conn = ibm_db_dbi.Connection(conn)
```

```
    selectQuery = "SELECT * FROM booktb where uname= '" +
uname + "' "
```

```
    dataframe = pandas.read_sql(selectQuery, pd_conn)
```

```
dataframe.to_sql('booktb1', con=engine, if_exists='append')
data = engine.execute("SELECT * FROM booktb1").fetchall()
```

```
return render_template('UOrderInfo.html', data=data)
```

```
@app.route("/UserHome")
```

```
def UserHome():
```

```
    uname = session['uname']
```

```
    conn = ibm_db.connect(dsn, "", "")
```

```
    pd_conn = ibm_db_dbi.Connection(conn)
```

```
    selectQuery = "SELECT * FROM regtb where  uname= " +
uname + " "
```

```
    dataframe = pandas.read_sql(selectQuery, pd_conn)
```

```
    dataframe.to_sql('booktb1', con=engine, if_exists='append')
```

```
    data = engine.execute("SELECT * FROM booktb1").fetchall()
```

```
    return render_template('UserHome.html', data=data)
```

```
@app.route("/ASalesInfo")
```

```
def ASalesInfo():
```

```
    conn = ibm_db.connect(dsn, "", "")
```

```
    pd_conn = ibm_db_dbi.Connection(conn)
```

```
    selectQuery = "SELECT * FROM booktb "
```

```
    dataframe = pandas.read_sql(selectQuery, pd_conn)
```

```
    dataframe.to_sql('booktb', con=engine, if_exists='append')
```

```
    data = engine.execute("SELECT * FROM booktb").fetchall()
```

```

        return render_template('ASalesInfo.html', data=data)

def main():
    app.run(debug=True, use_reloader=True)

if __name__ == '__main__':
    main()

```

HTML code:

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Aroma Shop - Home</title>
    <link rel="icon" href="static/img/Favicon.png" type="image/png">
    <link rel="stylesheet" href="static/vendors/bootstrap/bootstrap.min.css">
    <link rel="stylesheet" href="static/vendors/fontawesome/css/all.min.css">
    <link rel="stylesheet" href="static/vendors/themify-icons/themify-icons.css">
    <link rel="stylesheet" href="static/vendors/nice-select/nice-select.css">
    <link rel="stylesheet" href="static/vendors/owl-carousel/owl.theme.default.min.css">
    <link rel="stylesheet" href="static/vendors/owl-carousel/owl.carousel.min.css">

    <link rel="stylesheet" href="static/css/style.css">
</head>
<body>
    <!--===== Start Header Menu Area =====-->
    <header class="header_area">
    <div class="main_menu">
    <nav class="navbar navbar-expand-lg navbar-light">
    <div class="container">
    <a class="navbar-brand logo_h" href="index.html"></a>
    <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarSupportedContent"
    aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle
navigation">
    <span class="icon-bar"></span>
    <span class="icon-bar"></span>
    <span class="icon-bar"></span>
    </button>

```

```

<div class="collapse navbar-collapse offset" id="navbarSupportedContent">
  <ul class="nav navbar-nav menu_nav ml-auto mr-auto">
    <li class="nav-item active"><a class="nav-link" href="index.html">Home</a></li>
    <li class="nav-item submenu dropdown">
      <a href="#" class="nav-link dropdown-toggle" data-toggle="dropdown"
role="button" aria-haspopup="true"
      aria-expanded="false">Shop</a>
      <ul class="dropdown-menu">
        <li class="nav-item"><a class="nav-link" href="category.html">Shop
Category</a></li>
        <li class="nav-item"><a class="nav-link" href="single-product.html">Product
Details</a></li>
        <li class="nav-item"><a class="nav-link" href="checkout.html">Product
Checkout</a></li>
        <li class="nav-item"><a class="nav-link"
href="confirmation.html">Confirmation</a></li>
        <li class="nav-item"><a class="nav-link" href="cart.html">Shopping
Cart</a></li>
      </ul>
    </li>
    <li class="nav-item submenu dropdown">
      <a href="#" class="nav-link dropdown-toggle" data-toggle="dropdown"
role="button" aria-haspopup="true"
      aria-expanded="false">Blog</a>
      <ul class="dropdown-menu">
        <li class="nav-item"><a class="nav-link" href="blog.html">Blog</a></li>
        <li class="nav-item"><a class="nav-link" href="single-blog.html">Blog
Details</a></li>
      </ul>
    </li>
    <li class="nav-item submenu
dropdown">
      <a href="#" class="nav-link dropdown-toggle" data-toggle="dropdown"
role="button" aria-haspopup="true"
      aria-expanded="false">Pages</a>
      <ul class="dropdown-menu">
        <li class="nav-item"><a class="nav-link" href="sign.html">Login</a></li>
        <li class="nav-item"><a class="nav-link" href="/register">Register</a></li>
        <li class="nav-item"><a class="nav-link" href="tracking-
order.html">Tracking</a></li>
      </ul>
    </li>
    <li class="nav-item"><a class="nav-link" href="contact.html">Contact</a></li>
  </ul>
  <ul class="nav-shop">
    <li class="nav-item"><button><i class="ti-search"></i></button></li>
    <li class="nav-item"><button><i class="ti-shopping-cart"></i><span class="nav-
shop__circle">3</span></button> </li>
    <li class="nav-item"><a class="button button-header" href="#">Buy Now</a></li>

```

```

        </ul>
    </div>
</div>
</nav>
</div>
</header>
    <!--===== End Header Menu Area =====>

```

```

<main class="site-main">

```

```

    <!--===== Hero banner start =====>
    <section class="hero-banner">
        <div class="container">
            <div class="row no-gutters align-items-center pt-60px">
                <div class="col-5 d-none d-sm-block">
                    <div class="hero-banner__img">
                        
                    </div>
                </div>
                <div class="col-sm-7 col-lg-6 offset-lg-1 pl-4 pl-md-5 pl-lg-0">
                    <div class="hero-banner__content">
                        <h4>Shop is fun</h4>
                        <h1>Browse Our Premium Product</h1>
                        <p>Us which over of signs divide dominion deep fill bring they're meat beho upon
own earth without morning over third. Their male dry. They are great appear whose land fly
grass.</p>
                        <a class="button button-hero" href="#">Browse Now</a>
                    </div>
                </div>
            </div>
        </div>
    </section>

```

```

    <!--===== Hero banner start =====>

    <!--===== Hero Carousel start =====>
    <section class="section-margin mt-0">
        <div class="owl-carousel owl-theme hero-carousel">
            <div class="hero-carousel__slide">
                
                <a href="#" class="hero-carousel__slideOverlay">
                    <h3>Wireless Headphone</h3>
                    <p>Accessories Item</p>
                </a>
            </div>
            <div class="hero-carousel__slide">
                
                <a href="#" class="hero-carousel__slideOverlay">
                    <h3>Wireless Headphone</h3>
                    <p>Accessories Item</p>
                </a>
            </div>
        </div>
    </section>

```

```

</div>
<div class="hero-carousel__slide">
  
  <a href="#" class="hero-carousel__slideOverlay">
    <h3>Wireless Headphone</h3>
    <p>Accessories Item</p>
  </a>
</div>
</div>
</section>
<!--===== Hero Carousel end =====-->

<!-- ===== trending product section start ===== -->
<section class="section-margin calc-60px">
  <div class="container">
    <div class="section-intro pb-60px">
      <p>Popular Item in the market</p>
      <h2>Trending <span class="section-intro__style">Product</span></h2>
    </div>
    <div class="row">
      <div class="col-md-6 col-lg-4 col-xl-3">
        <div class="card text-center card-product">
          <div class="card-product__img">
            
            <ul class="card-product__imgOverlay">
              <li><button><i class="ti-search"></i></button></li>
              <li><button><i class="ti-shopping-cart"></i></button></li>
              <li><button><i class="ti-heart"></i></button></li>
            </ul>
          </div>
          <div class="card-body">
            <p>Accessories</p>
            <h4 class="card-product__title"><a href="single-product.html">Quartz Belt
Watch</a></h4>
            <p class="card-product__price">$150.00</p>
          </div>
        </div>
      </div>
      <div class="col-md-6 col-lg-4 col-xl-3">
        <div class="card text-center card-product">
          <div class="card-product__img">
            
            <ul class="card-product__imgOverlay">
              <li><button><i class="ti-search"></i></button></li>
              <li><button><i class="ti-shopping-cart"></i></button></li>
              <li><button><i class="ti-heart"></i></button></li>
            </ul>
          </div>
          <div class="card-body">
            <p>Beauty</p>

```



```

        <h4 class="card-product__title"><a href="single-product.html">Women
Freshwash</a></h4>
        <p class="card-product__price">$150.00</p>
    </div>
</div>
</div>
<div class="col-md-6 col-lg-4 col-xl-3">
    <div class="card text-center card-product">
        <div class="card-product__img">
            
            <ul class="card-product__imgOverlay">
                <li><button><i class="ti-search"></i></button></li>
                <li><button><i class="ti-shopping-cart"></i></button></li>
                <li><button><i class="ti-heart"></i></button></li>
            </ul>
        </div>
        <div class="card-body">
            <p>Decor</p>
            <h4 class="card-product__title"><a href="single-product.html">Room Flash
Light</a></h4>
            <p class="card-product__price">$150.00</p>
        </div>
    </div>
</div>
<div class="col-md-6 col-lg-4 col-xl-3">
    <div class="card text-center card-product">
        <div class="card-product__img">
            
            <ul class="card-product__imgOverlay">
                <li><button><i class="ti-search"></i></button></li>
                <li><button><i class="ti-shopping-cart"></i></button></li>
                <li><button><i class="ti-heart"></i></button></li>
            </ul>
        </div>
        <div class="card-body">
            <p>Decor</p>
            <h4 class="card-product__title"><a href="single-product.html">Room Flash
Light</a></h4>
            <p class="card-product__price">$150.00</p>
        </div>
    </div>
</div>
<div class="col-md-6 col-lg-4 col-xl-3">
    <div class="card text-center card-product">
        <div class="card-product__img">
            
            <ul class="card-product__imgOverlay">
                <li><button><i class="ti-search"></i></button></li>
                <li><button><i class="ti-shopping-cart"></i></button></li>
                <li><button><i class="ti-heart"></i></button></li>
            </ul>
        </div>
        <div class="card-body">
            <p>Decor</p>
            <h4 class="card-product__title"><a href="single-product.html">Room Flash
Light</a></h4>
            <p class="card-product__price">$150.00</p>
        </div>
    </div>
</div>

```

```

        </ul>
    </div>
    <div class="card-body">
        <p>Accessories</p>
        <h4 class="card-product__title"><a href="single-product.html">Man Office
Bag</a></h4>
        <p class="card-product__price">$150.00</p>
    </div>
</div>
</div>
<div class="col-md-6 col-lg-4 col-xl-3">
    <div class="card text-center card-product">
        <div class="card-product__img">
            
            <ul class="card-product__imgOverlay">
                <li><button><i class="ti-search"></i></button></li>
                <li><button><i class="ti-shopping-cart"></i></button></li>
                <li><button><i class="ti-heart"></i></button></li>
            </ul>
        </div>
        <div class="card-body">
            <p>Kids Toy</p>
            <h4 class="card-product__title"><a href="single-product.html">Charging
Car</a></h4>
            <p class="card-product__price">$150.00</p>
        </div>
    </div>
</div>
<div class="col-md-6 col-lg-4 col-xl-3">
    <div class="card text-center card-product">
        <div class="card-product__img">
            
            <ul class="card-product__imgOverlay">
                <li><button><i class="ti-search"></i></button></li>
                <li><button><i class="ti-shopping-cart"></i></button></li>
                <li><button><i class="ti-heart"></i></button></li>
            </ul>
        </div>
        <div class="card-body">
            <p>Accessories</p>
            <h4 class="card-product__title"><a href="single-product.html">Bluetooth
Speaker</a></h4>
            <p class="card-product__price">$150.00</p>
        </div>
    </div>
</div>
<div class="col-md-6 col-lg-4 col-xl-3">
    <div class="card text-center card-product">
        <div class="card-product__img">
            

```

```

        <ul class="card-product__imgOverlay">
            <li><button><i class="ti-search"></i></button></li>
            <li><button><i class="ti-shopping-cart"></i></button></li>
            <li><button><i class="ti-heart"></i></button></li>
        </ul>
    </div>
    <div class="card-body">
        <p>Kids Toy</p>
        <h4 class="card-product__title"><a href="#">Charging Car</a></h4>
        <p class="card-product__price">$150.00</p>
    </div>
</div>
</div>
</div>
</div>
</section>
<!-- ===== trending product section end ===== -->

```

```

<!-- ===== offer section start ===== -->
<section class="offer" id="parallax-1" data-anchor-target="#parallax-1" data-300-
top="background-position: 20px 30px" data-top-bottom="background-position: 0 20px">
    <div class="container">
        <div class="row">
            <div class="col-xl-5">
                <div class="offer__content text-center">
                    <h3>Up To 50% Off</h3>
                    <h4>Winter Sale</h4>
                    <p>Him she'd let them sixth saw light</p>
                    <a class="button button--active mt-3 mt-xl-4" href="#">Shop Now</a>
                </div>
            </div>
        </div>
    </div>
</section>
<!-- ===== offer section end ===== -->

```

```

<!-- ===== Best Selling item carousel ===== -->
<section class="section-margin calc-60px">
    <div class="container">
        <div class="section-intro pb-60px">
            <p>Popular Item in the market</p>
            <h2>Best <span class="section-intro__style">Sellers</span></h2>
        </div>
        <div class="owl-carousel owl-theme" id="bestSellerCarousel">
            <div class="card text-center card-product">
                <div class="card-product__img">
                    
                    <ul class="card-product__imgOverlay">
                        <li><button><i class="ti-search"></i></button></li>

```

```

        <li><button><i class="ti-shopping-cart"></i></button></li>
        <li><button><i class="ti-heart"></i></button></li>
    </ul>
</div>
<div class="card-body">
    <p>Accessories</p>
    <h4 class="card-product__title"><a href="single-product.html">Quartz Belt
Watch</a></h4>
    <p class="card-product__price">$150.00</p>
</div>
</div>

```

```

<div class="card text-center card-product">
    <div class="card-product__img">
        
        <ul class="card-product__imgOverlay">
            <li><button><i class="ti-search"></i></button></li>
            <li><button><i class="ti-shopping-cart"></i></button></li>
            <li><button><i class="ti-heart"></i></button></li>
        </ul>
    </div>
    <div class="card-body">
        <p>Beauty</p>
        <h4 class="card-product__title"><a href="single-product.html">Women
Freshwash</a></h4>
        <p class="card-product__price">$150.00</p>
    </div>
</div>

```

```

<div class="card text-center card-product">
    <div class="card-product__img">
        
        <ul class="card-product__imgOverlay">
            <li><button><i class="ti-search"></i></button></li>
            <li><button><i class="ti-shopping-cart"></i></button></li>
            <li><button><i class="ti-heart"></i></button></li>
        </ul>
    </div>
    <div class="card-body">
        <p>Decor</p>
        <h4 class="card-product__title"><a href="single-product.html">Room Flash
Light</a></h4>
        <p class="card-product__price">$150.00</p>
    </div>
</div>

```

```

<div class="card text-center card-product">
    <div class="card-product__img">
        
        <ul class="card-product__imgOverlay">

```

```

        <li><button><i class="ti-search"></i></button></li>
        <li><button><i class="ti-shopping-cart"></i></button></li>
        <li><button><i class="ti-heart"></i></button></li>
    </ul>
</div>
<div class="card-body">
    <p>Decor</p>
    <h4 class="card-product__title"><a href="single-product.html">Room Flash
Light</a></h4>
    <p class="card-product__price">$150.00</p>
</div>
</div>

```

```

<div class="card text-center card-product">
    <div class="card-product__img">
        
        <ul class="card-product__imgOverlay">
            <li><button><i class="ti-search"></i></button></li>
            <li><button><i class="ti-shopping-cart"></i></button></li>
            <li><button><i class="ti-heart"></i></button></li>
        </ul>
    </div>
    <div class="card-body">
        <p>Accessories</p>
        <h4 class="card-product__title"><a href="single-product.html">Quartz Belt
Watch</a></h4>
        <p class="card-product__price">$150.00</p>
    </div>
</div>

```

```

<div class="card text-center card-product">
    <div class="card-product__img">
        
        <ul class="card-product__imgOverlay">
            <li><button><i class="ti-search"></i></button></li>
            <li><button><i class="ti-shopping-cart"></i></button></li>
            <li><button><i class="ti-heart"></i></button></li>
        </ul>
    </div>
    <div class="card-body">
        <p>Beauty</p>
        <h4 class="card-product__title"><a href="single-product.html">Women
Freshwash</a></h4>
        <p class="card-product__price">$150.00</p>
    </div>
</div>

```

```

<div class="card text-center card-product">
    <div class="card-product__img">
        

```

```

        <ul class="card-product__imgOverlay">
            <li><button><i class="ti-search"></i></button></li>
            <li><button><i class="ti-shopping-cart"></i></button></li>
            <li><button><i class="ti-heart"></i></button></li>
        </ul>
    </div>
    <div class="card-body">
        <p>Decor</p>
        <h4 class="card-product__title"><a href="single-product.html">Room Flash
Light</a></h4>
        <p class="card-product__price">$150.00</p>
    </div>
</div>

<div class="card text-center card-product">
    <div class="card-product__img">
        
        <ul class="card-product__imgOverlay">
            <li><button><i class="ti-search"></i></button></li>
            <li><button><i class="ti-shopping-cart"></i></button></li>
            <li><button><i class="ti-heart"></i></button></li>
        </ul>
    </div>
    <div class="card-body">
        <p>Decor</p>
        <h4 class="card-product__title"><a href="single-product.html">Room Flash
Light</a></h4>
        <p class="card-product__price">$150.00</p>
    </div>
</div>
</div>
</div>
</section>
<!-- ===== Best Selling item carousel end ===== -->

<!-- ===== Blog section start ===== -->
<section class="blog">
    <div class="container">
        <div class="section-intro pb-60px">
            <p>Popular Item in the market</p>
            <h2>Latest <span class="section-intro__style">News</span></h2>
        </div>

        <div class="row">
            <div class="col-md-6 col-lg-4 mb-4 mb-lg-0">
                <div class="card card-blog">
                    <div class="card-blog__img">
                        
                    </div>
                    <div class="card-body">

```

```
<ul class="card-blog__info">
  <li><a href="#">By Admin</a></li>
  <li><a href="#"><i class="ti-comments-smiley"></i> 2 Comments</a></li>
</ul>
<h4 class="card-blog__title"><a href="single-blog.html">The Richland Center
Shooping News and weekly shooper</a></h4>
<p>Let one fifth i bring fly to divided face for bearing divide unto seed. Winged
divided light Forth.</p>
<a class="card-blog__link" href="#">Read More <i class="ti-arrow-
right"></i></a>
</div>
</div>
</div>
```

```
<div class="col-md-6 col-lg-4 mb-4 mb-lg-0">
<div class="card card-blog">
<div class="card-blog__img">

</div>
<div class="card-body">
<ul class="card-blog__info">
  <li><a href="#">By Admin</a></li>
  <li><a href="#"><i class="ti-comments-smiley"></i> 2 Comments</a></li>
</ul>
<h4 class="card-blog__title"><a href="single-blog.html">The Shopping News also
offers top-quality printing services</a></h4>
<p>Let one fifth i bring fly to divided face for bearing divide unto seed. Winged
divided light Forth.</p>
<a class="card-blog__link" href="#">Read More <i class="ti-arrow-
right"></i></a>
</div>
</div>
</div>
```

```
<div class="col-md-6 col-lg-4 mb-4 mb-lg-0">
<div class="card card-blog">
<div class="card-blog__img">

</div>
<div class="card-body">
<ul class="card-blog__info">
  <li><a href="#">By Admin</a></li>
  <li><a href="#"><i class="ti-comments-smiley"></i> 2 Comments</a></li>
</ul>
<h4 class="card-blog__title"><a href="single-blog.html">Professional design staff
and efficient equipment you'll find we offer</a></h4>
<p>Let one fifth i bring fly to divided face for bearing divide unto seed. Winged
divided light Forth.</p>
<a class="card-blog__link" href="#">Read More <i class="ti-arrow-
right"></i></a>
```

```

        </div>
    </div>
</div>
</div>
</div>
</section>
<!-- ===== Blog section end ===== -->

<!-- ===== Subscribe section start ===== -->
<section class="subscribe-position">
    <div class="container">
        <div class="subscribe text-center">
            <h3 class="subscribe__title">Get Update From Anywhere</h3>
            <p>Bearing Void gathering light light his evening unto dont afraid</p>
            <div id="mc_embed_signup">
                <form target="_blank" action="https://spondonit.us12.list-
manage.com/subscribe/post?u=1462626880ade1ac87bd9c93a&id=92a4423d01"
method="get" class="subscribe-form form-inline mt-5 pt-1">
                    <div class="form-group ml-sm-auto">
                        <input class="form-control mb-1" type="email" name="EMAIL"
placeholder="Enter your email" onfocus="this.placeholder = ''" onblur="this.placeholder =
'Your Email Address '" >
                            <div class="info"></div>
                        </div>
                        <button class="button button-subscribe mr-auto mb-1" type="submit">Subscribe
Now</button>
                            <div style="position: absolute; left: -5000px;">
                                <input name="b_36c4fd991d266f23781ded980_aefe40901a" tabindex="-1"
value="" type="text">
                                    </div>
                                </div>
                            </form>
                        </div>
                    </div>
                </div>
            </section>
            <!-- ===== Subscribe section end ===== -->

</main>

<!--===== Start footer Area =====-->
<footer class="footer">
    <div class="footer-area">
        <div class="container">
            <div class="row section_gap">
                <div class="col-lg-3 col-md-6 col-sm-6">

```



Mission</h4>

cattle in. Gathering thing made fly you're no

Gathering thing us land years living.

Links</h4>

href="#">Product</a></li>

href="#">Contact</a></li>

wrap">

src="static/img/gallery/r1.jpg" alt=""></li>

src="static/img/gallery/r2.jpg" alt=""></li>

src="static/img/gallery/r3.jpg" alt=""></li>

src="static/img/gallery/r5.jpg" alt=""></li>

src="static/img/gallery/r7.jpg" alt=""></li>

src="static/mg/gallery/r8.jpg" alt=""></li>

```
<div class="single-footer-widget tp_widgets">
  <h4 class="footer_title large_title">Our
```

```
<p>
```

So seed seed green that winged

divided deep moved us lan

```
</p>
```

```
<p>
```

So seed seed green that winged

cattle in. Gathering thing made fly you're no divided deep moved

```
</p>
```

```
</div>
```

```
</div>
```

```
<div class="offset-lg-1 col-lg-2 col-md-6 col-sm-6">
```

```
<div class="single-footer-widget tp_widgets">
```

```
<h4 class="footer_title">Quick
```

```
<ul class="list">
```

```
<li><a href="#">Home</a></li>
```

```
<li><a href="#">Shop</a></li>
```

```
<li><a href="#">Blog</a></li>
```

```
<li><a
```

```
<li><a href="#">Brand</a></li>
```

```
<li><a
```

```
</ul>
```

```
</div>
```

```
</div>
```

```
<div class="col-lg-2 col-md-6 col-sm-6">
```

```
<div class="single-footer-widget instafeed">
```

```
<h4 class="footer_title">Gallery</h4>
```

```
<ul class="list instafeed d-flex flex-
```

```
<li><img
```

```
<li><img
```

```
<li><img
```

```
<li><img
```

```
<li><img
```

```
<li><img
```

```
</ul>
```

```
</div>
```

```

</div>
<div class="offset-lg-1 col-lg-3 col-md-6 col-sm-6">
  <div class="single-footer-widget tp_widgets">
    <h4 class="footer_title">Contact

    <div class="ml-40">
      <p class="sm-head">
        <span class="fa fa-
location-arrow"></span>

        Head Office
      </p>
      <p>123, Main Street, Your

    City</p>

    <p class="sm-head">
      <span class="fa fa-
phone"></span>

      Phone Number
    </p>
    <p>
      +123 456 7890 <br>
      +123 456 7890
    </p>

    <p class="sm-head">
      <span class="fa fa-
envelope"></span>

      Email
    </p>
    <p>
      free@infoexample.com

      www.infoexample.com
    </p>
  </div>
</div>
</div>
</div>
</div>
</div>
<div class="footer-bottom">
  <div class="container">
    <div class="row d-flex">
      <p class="col-lg-12 footer-text text-center">
        <!-- Link back to Colorlib can't be removed.

```

Template is licensed under CC BY 3.0. -->

Copyright &copy;<script>document.write(new Date().getFullYear());</script> All rights reserved | This template is made with <i class="fa fa-heart" aria-hidden="true"></i> by <a href="https://colorlib.com" target="\_blank">Colorlib</a>

```
<!-- Link back to Colorlib can't be removed. Template is licensed under CC BY 3.0. --></p>
</div>
</div>
</div>
</footer>
<!--===== End footer Area =====-->
```

```
<script src="static/vendors/jquery/jquery-3.2.1.min.js"></script>
<script src="static/vendors/bootstrap/bootstrap.bundle.min.js"></script>
<script src="static/vendors/scrollr.min.js"></script>
<script src="static/vendors/owl-carousel/owl.carousel.min.js"></script>
<script src="static/vendors/nice-select/jquery.nice-select.min.js"></script>
<script src="static/vendors/jquery.ajaxchimp.min.js"></script>
<script src="static/vendors/mail-script.js"></script>
<script src="static/js/main.js"></script>
</body>
</html>
```

## GitHub & Project Demo Link

### ❖ Our GitHub Repository Direct Link

[IBM-EPBL/IBM-Project-39580-1664791696](https://github.com/IBM-EPBL/IBM-Project-39580-1664791696):Smart Fashion Recommender Application Link(gitHub.com)

### ❖ Project Demonstration Direct Link

Smart Fashion Recommender Application -YouTube  
<https://youtu.be/haiJI1euW8A>