

SMART FASHION RECOMMENDER APPLICATION
A PROJECT REPORT

Submitted by

TEAM ID : PNT2022TMID03701

| | |
|------------------------------|--------------------------------|
| SARKAR MURALI GNANESH | (Reg. No. 212219060233) |
| PALAMAKULA JAHNAVI | (Reg. No. 212219060189) |
| GUDLA AKASH | (Reg. No. 212219060094) |
| GUDA TEJANANDA REDDY | (Reg.No.212219060093) |

In partial fulfilment for the award of the degree

Of

BACHELOR OF ENGINEERING

IN

ELECTRONICS AND COMMUN ITATION ENGINEERING

SAVEETHA ENGINEERING COLLEGE

ANNA UNIVERSITY:CHENNAI 600025

NOV 2022

1. INTRODUCTION

- 1.1. Project Overview
- 1.2. Purpose

2. LITERATURE SURVEY

- 2.1. Existing problem
- 2.2. References
- 2.3. Problem Statement Definition

3. IDEATION & PROPOSED SOLUTION

- 3.1. Empathy Map Canvas
- 3.2. Ideation & Brainstorming
- 3.3. Proposed Solution
- 3.4. Problem Solution fit

4. REQUIREMENT ANALYSIS

- 4.1. Functional requirement
- 4.2. Non-Functional requirements

5. PROJECT DESIGN

- 5.1. Data Flow Diagrams
- 5.2. Solution & Technical Architecture
- 5.3. User Stories

6. PROJECT PLANNING & SCHEDULING

- 6.1. Sprint Planning & Estimation
- 6.2. Sprint Delivery Schedule
- 6.3. Reports from JIRA

7. CODING & SOLUTIONING

- 7.1. Feature 1
- 7.2. Feature 2
- 7.3. Database Schema

8. TESTING

8.1. Test Cases

8.2. User Acceptance Testing

9. RESULTS

9.1. Performance Metrics

10.ADVANTAGES & DISADVANTAGES

11.CONCLUSION

12.FUTURE SCOPE

13.APPENDIX

Source Code

GitHub & Project Demo Link

1. INTRODUCTION

1.1 Project Overview

The increasing popularity of online fashion and online retail platforms is having a visible impact on the shopping experience of billions of customers, making millions of products available in online catalogs thus eliminating the need for physical visits to various stores and for waiting in long queues or trying on clothes in dressing rooms by providing personalized and affordable deliveries.

This in turn has created novel challenges for platform providers, within which proper understanding of fashion choices of shoppers plays a crucial role.

Shoppers tend to feel overwhelmed by the sheer choice of the assortment and brands, not being able to receive effective suggestions matching their style preferences as well as not being able to spot the right size and fit during the shopping experience.

As a result, recommender systems are gaining momentum by mining through large and diverse silos of product catalogs as well as customer datasets in order to provide personalized recommendations of outfits, complimenting the shopping session with similar and relevant products, understanding and suggesting the correct size and fit for shoppers, recommending with personalized styles and leveraging the social influence affecting the choice of style and buying behavior of new generations of shoppers. To this end, within this chapter we aim to present a state of the art view of the advancements within the field of recommendation systems in the domain of fashion. In this project We have come up with a new innovative solution through which you can directly do your online shopping based on your choice without any search. It can be done by using the chatbot.

1.2 Purpose

- a) Using chatbot we can manage user's choices and orders.
- b) The chatbot can give recommendations to the users based on their interests.
- c) It can promote the best deals and offers on that day.
- d) It will store the customer's details and orders in the database.
- e) The chatbot will send a notification to customers if the order is confirmed.
- f) Chatbots can also help in collecting customer feedback.

2.LITERATURE SURVEY

2.1 Existing Problem

Customers buy products and order them, If they want to know further details, while they cant find them in websites, They keep step backward to buy their desired product .If the has any queries in the product or how to process to get the product, they wait for the seller to explain it. Although using chatbots may provide faster customer service overall, they aren't perfect. Simple ones may have only limited responses for customers. Therefore, not all customers will get the answers they are searching for.

Only 34% of respondents stated they would be comfortable using chatbots in an online retail situation. Most customers don't want chatbots.

Chatbots were created to respond to simple questions that can be answered with facts. Chatbots have limited responses, so they're not often able to answer multi-part questions or questions that require decisions. This often means your customers are left without a solution, and have to go through more steps to contact your support team.

2.2 References

- [1] E. Turban, D. King, J. Lee, and D. Viehland, Electronic Commerce: A Managerial Perspective. Upper Saddle River, NJ, USA: Prentice-Hall, 2002. [2] R. T. Wigand, ' Electronic commerce: Definition, theory, and context,' Inf. Soc., vol. 13, no. 1, pp. 1–16, Mar. 1997.
- [3] P. Ratnasingham, ' The importance of trust in electronic commerce,' Internet Res., vol. 8, no. 4, pp. 313–321, Oct. 1998.
- [4] Y. Guo, M. Wang, and X. Li, ' Application of an improved Apriori algo?rithm

in a mobile e-commerce recommendation system,' *Ind. Manage. Data Syst.*, vol. 117, no. 2, pp. 287–303, Mar. 2017.

[5] B. Zareie and N. J. Navimipour, 'The effect of electronic learning systems on the employee's commitment,' *Int. J. Manage. Edu.*, vol. 14, no. 2, pp. 167–175, 2016.

[6] N. J. Navimipour and Z. Soltani, 'The impact of cost, technology acceptance and employees' satisfaction on the effectiveness of the electronic customer relationship management systems,' *Comput. Hum. Behav.*, vol. 55, pp. 1052–1066, Feb. 2016.

[7] S. H. Sharif, S. Mahmazi, N. J. Navimipour, and B. F. Aghdam, 'A review on search and discovery mechanisms in social networks,' *Int. J. Inf. Eng. Electron. Bus.*, vol. 5, no. 6, p. 64, 2013.

[8] N. J. Navimipour, A. M. Rahmani, A. H. Navin, and M. Hosseinzadeh, 'Expert cloud: A cloud-based framework to share the knowledge and skills of human resources,' *Comput. Hum. Behav.*, vol. 46, pp. 57–74, May 2015. [9] S. Hazratzadeh and N. J. Navimipour, 'Colleague recommender system in the expert cloud using features matrix,' *Kybernetes*, vol. 45, no. 9, pp. 1342–1357, Oct. 2016.

2.3 Problem Statement Definition

Problem Statement 1:

The User Needs a way to Find Trending Fashion Clothes so that Here find the All Collections.

Problem Statement 2:

The User Needs a way to Find Offers and Discounts so that Here User easy to find Daily Offers.

Problem Statement 3:

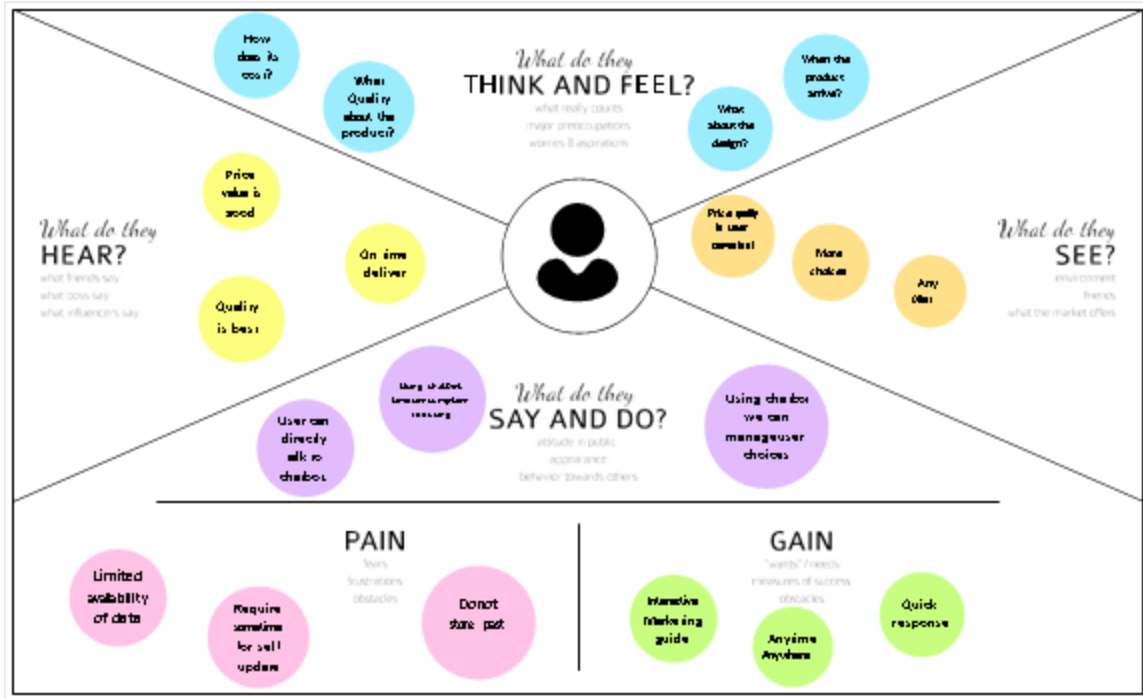
The User Needs a way to Assistant for finding Clothes so that Here User got the Chat Bot assistant.

Problem Statement 4:

The Sellers Needs a way to struggling to sells products offline so that Here Sellers will Sell Products via our application.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 Ideation & Brainstorming



SMART FASHION RECOMMENDER APPLICATION

Fashion applications have seen tremendous growth and are now one of the most used programs in the e-commerce field. The needs of people are continuously evolving, creating a room for innovation among the applications. Having an AI program that understands the algorithm of a specific application can be of great aid. We are implementing such a chatbot, which is fed with the knowledge of the application's algorithm and helps the user completely from finding their needs to processing the payment and initiating delivery.



Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

10 minutes



A Team gathering

Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.



B Set the goal

Think about the problem you'll be focusing on solving in the brainstorming session.



C Learn how to use the facilitation tools

Use the Facilitation Superpowers to run a happy and productive session.

[Open article](#)



Define your problem statement

Unavailability of chatbots that are interactive enough to navigate the user to do whatever they want. The amount of toll a user has to go through to look for a product they desire for. Need for a more User-friendly Interface. The main aim of the project is to develop a smart chatbot that is able to understand the needs of the user and recommend products of desire.



Key rules of brainstorming

To run a smooth and productive session



Stay in topic.



Encourage wild ideas.



Defer judgment.



Listen to others.



Go for volume.



If possible, be visual.



Brainstorm

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

10 minutes

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

RURANA PARVEEN J

1. User should be able to interact with the chatbot.

2. The chatbot should be able to understand the user's needs.

3. The chatbot should be able to recommend products.

4. The chatbot should be able to process payments.

5. The chatbot should be able to initiate delivery.

6. The chatbot should be able to handle user queries.

7. The chatbot should be able to provide a seamless user experience.

8. The chatbot should be able to handle user feedback.

9. The chatbot should be able to handle user complaints.

10. The chatbot should be able to handle user suggestions.

11. The chatbot should be able to handle user inquiries.

12. The chatbot should be able to handle user requests.

13. The chatbot should be able to handle user orders.

14. The chatbot should be able to handle user payments.

15. The chatbot should be able to handle user deliveries.

16. The chatbot should be able to handle user returns.

17. The chatbot should be able to handle user cancellations.

18. The chatbot should be able to handle user refunds.

19. The chatbot should be able to handle user complaints.

20. The chatbot should be able to handle user suggestions.

21. The chatbot should be able to handle user inquiries.

22. The chatbot should be able to handle user requests.

23. The chatbot should be able to handle user orders.

24. The chatbot should be able to handle user payments.

25. The chatbot should be able to handle user deliveries.

26. The chatbot should be able to handle user returns.

27. The chatbot should be able to handle user cancellations.

28. The chatbot should be able to handle user refunds.

29. The chatbot should be able to handle user complaints.

30. The chatbot should be able to handle user suggestions.

31. The chatbot should be able to handle user inquiries.

32. The chatbot should be able to handle user requests.

33. The chatbot should be able to handle user orders.

34. The chatbot should be able to handle user payments.

35. The chatbot should be able to handle user deliveries.

36. The chatbot should be able to handle user returns.

37. The chatbot should be able to handle user cancellations.

38. The chatbot should be able to handle user refunds.

39. The chatbot should be able to handle user complaints.

40. The chatbot should be able to handle user suggestions.

41. The chatbot should be able to handle user inquiries.

42. The chatbot should be able to handle user requests.

43. The chatbot should be able to handle user orders.

44. The chatbot should be able to handle user payments.

45. The chatbot should be able to handle user deliveries.

46. The chatbot should be able to handle user returns.

47. The chatbot should be able to handle user cancellations.

48. The chatbot should be able to handle user refunds.

49. The chatbot should be able to handle user complaints.

50. The chatbot should be able to handle user suggestions.

PARAMESHWARI B

1. User should be able to interact with the chatbot.

2. The chatbot should be able to understand the user's needs.

3. The chatbot should be able to recommend products.

4. The chatbot should be able to process payments.

5. The chatbot should be able to initiate delivery.

6. The chatbot should be able to handle user queries.

7. The chatbot should be able to provide a seamless user experience.

8. The chatbot should be able to handle user feedback.

9. The chatbot should be able to handle user complaints.

10. The chatbot should be able to handle user suggestions.

11. The chatbot should be able to handle user inquiries.

12. The chatbot should be able to handle user requests.

13. The chatbot should be able to handle user orders.

14. The chatbot should be able to handle user payments.

15. The chatbot should be able to handle user deliveries.

16. The chatbot should be able to handle user returns.

17. The chatbot should be able to handle user cancellations.

18. The chatbot should be able to handle user refunds.

19. The chatbot should be able to handle user complaints.

20. The chatbot should be able to handle user suggestions.

21. The chatbot should be able to handle user inquiries.

22. The chatbot should be able to handle user requests.

23. The chatbot should be able to handle user orders.

24. The chatbot should be able to handle user payments.

25. The chatbot should be able to handle user deliveries.

26. The chatbot should be able to handle user returns.

27. The chatbot should be able to handle user cancellations.

28. The chatbot should be able to handle user refunds.

29. The chatbot should be able to handle user complaints.

30. The chatbot should be able to handle user suggestions.

31. The chatbot should be able to handle user inquiries.

32. The chatbot should be able to handle user requests.

33. The chatbot should be able to handle user orders.

34. The chatbot should be able to handle user payments.

35. The chatbot should be able to handle user deliveries.

36. The chatbot should be able to handle user returns.

37. The chatbot should be able to handle user cancellations.

38. The chatbot should be able to handle user refunds.

39. The chatbot should be able to handle user complaints.

40. The chatbot should be able to handle user suggestions.

41. The chatbot should be able to handle user inquiries.

42. The chatbot should be able to handle user requests.

43. The chatbot should be able to handle user orders.

44. The chatbot should be able to handle user payments.

45. The chatbot should be able to handle user deliveries.

46. The chatbot should be able to handle user returns.

47. The chatbot should be able to handle user cancellations.

48. The chatbot should be able to handle user refunds.

49. The chatbot should be able to handle user complaints.

50. The chatbot should be able to handle user suggestions.

51. The chatbot should be able to handle user inquiries.

52. The chatbot should be able to handle user requests.

53. The chatbot should be able to handle user orders.

54. The chatbot should be able to handle user payments.

55. The chatbot should be able to handle user deliveries.

56. The chatbot should be able to handle user returns.

57. The chatbot should be able to handle user cancellations.

58. The chatbot should be able to handle user refunds.

59. The chatbot should be able to handle user complaints.

60. The chatbot should be able to handle user suggestions.

61. The chatbot should be able to handle user inquiries.

62. The chatbot should be able to handle user requests.

63. The chatbot should be able to handle user orders.

64. The chatbot should be able to handle user payments.

65. The chatbot should be able to handle user deliveries.

66. The chatbot should be able to handle user returns.

67. The chatbot should be able to handle user cancellations.

68. The chatbot should be able to handle user refunds.

69. The chatbot should be able to handle user complaints.

70. The chatbot should be able to handle user suggestions.

71. The chatbot should be able to handle user inquiries.

72. The chatbot should be able to handle user requests.

73. The chatbot should be able to handle user orders.

74. The chatbot should be able to handle user payments.

75. The chatbot should be able to handle user deliveries.

76. The chatbot should be able to handle user returns.

77. The chatbot should be able to handle user cancellations.

78. The chatbot should be able to handle user refunds.

79. The chatbot should be able to handle user complaints.

80. The chatbot should be able to handle user suggestions.

81. The chatbot should be able to handle user inquiries.

82. The chatbot should be able to handle user requests.

83. The chatbot should be able to handle user orders.

84. The chatbot should be able to handle user payments.

85. The chatbot should be able to handle user deliveries.

86. The chatbot should be able to handle user returns.

87. The chatbot should be able to handle user cancellations.

88. The chatbot should be able to handle user refunds.

89. The chatbot should be able to handle user complaints.

90. The chatbot should be able to handle user suggestions.

91. The chatbot should be able to handle user inquiries.

92. The chatbot should be able to handle user requests.

93. The chatbot should be able to handle user orders.

94. The chatbot should be able to handle user payments.

95. The chatbot should be able to handle user deliveries.

96. The chatbot should be able to handle user returns.

97. The chatbot should be able to handle user cancellations.

98. The chatbot should be able to handle user refunds.

99. The chatbot should be able to handle user complaints.

100. The chatbot should be able to handle user suggestions.

SANDHYA B

1. User should be able to interact with the chatbot.

2. The chatbot should be able to understand the user's needs.

3. The chatbot should be able to recommend products.

4. The chatbot should be able to process payments.

5. The chatbot should be able to initiate delivery.

6. The chatbot should be able to handle user queries.

7. The chatbot should be able to provide a seamless user experience.

8. The chatbot should be able to handle user feedback.

9. The chatbot should be able to handle user complaints.

10. The chatbot should be able to handle user suggestions.

11. The chatbot should be able to handle user inquiries.

12. The chatbot should be able to handle user requests.

13. The chatbot should be able to handle user orders.

14. The chatbot should be able to handle user payments.

15. The chatbot should be able to handle user deliveries.

16. The chatbot should be able to handle user returns.

17. The chatbot should be able to handle user cancellations.

18. The chatbot should be able to handle user refunds.

19. The chatbot should be able to handle user complaints.

20. The chatbot should be able to handle user suggestions.

21. The chatbot should be able to handle user inquiries.

22. The chatbot should be able to handle user requests.

23. The chatbot should be able to handle user orders.

24. The chatbot should be able to handle user payments.

25. The chatbot should be able to handle user deliveries.

26. The chatbot should be able to handle user returns.

27. The chatbot should be able to handle user cancellations.

28. The chatbot should be able to handle user refunds.

29. The chatbot should be able to handle user complaints.

30. The chatbot should be able to handle user suggestions.

31. The chatbot should be able to handle user inquiries.

32. The chatbot should be able to handle user requests.

33. The chatbot should be able to handle user orders.

34. The chatbot should be able to handle user payments.

35. The chatbot should be able to handle user deliveries.

36. The chatbot should be able to handle user returns.

37. The chatbot should be able to handle user cancellations.

38. The chatbot should be able to handle user refunds.

39. The chatbot should be able to handle user complaints.

40. The chatbot should be able to handle user suggestions.

41. The chatbot should be able to handle user inquiries.

42. The chatbot should be able to handle user requests.

43. The chatbot should be able to handle user orders.

44. The chatbot should be able to handle user payments.

45. The chatbot should be able to handle user deliveries.

46. The chatbot should be able to handle user returns.

47. The chatbot should be able to handle user cancellations.

48. The chatbot should be able to handle user refunds.

49. The chatbot should be able to handle user complaints.

50. The chatbot should be able to handle user suggestions.

51. The chatbot should be able to handle user inquiries.

52. The chatbot should be able to handle user requests.

53. The chatbot should be able to handle user orders.

4 Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes

After you collaborate

You can export the mural as an image or pdf to share with members of your company who might find it helpful.

Quick add-ons

- Share the mural**
Share a view link to the mural with stakeholders to keep them in the loop about the outcomes of the session.
- Export the mural**
Export a copy of the mural as a PNG or PDF to attach to emails, include in slides, or save in your drive.

Keep moving forward

- Strategy blueprint**
Define the components of a new idea or strategy.
[Open the template →](#)
- Customer experience journey map**
Understand customer needs, motivations, and obstacles for an experience.
[Open the template →](#)
- Strengths, weaknesses, opportunities & threats**
Identify strengths, weaknesses, opportunities, and threats (SWOT) to develop a plan.
[Open the template →](#)

[Share template feedback](#)

3.3 Proposed Solution

| S.No. | Parameter | Description |
|-------|--|---|
| 1. | Problem Statement (Problem to be solved) | A new innovative solution through which users can directly do online shopping based on the choice without |

| | | |
|-----------|-----------------------------|--|
| | | <p>any search.</p> <p>It can be done using a chatbot.</p> <p>Modules Used:</p> <p>1. Admin</p> <p>The role of the admin is to check out the database about the stock and have a track of all the things that the user are purchasing.</p> <p>1. User</p> <p>The user will login into the website and go through the products available on the website .Instead of navigating to several screens for booking products online ,The user can directly talk to the chatbot regarding the product</p> |
| 2. | Idea / Solution description | <p>The idea mainly focus on the designing the chatbot using AI algorithms like NLP(Natural Language Processing) which makes the user friendly environment instead of searching the product.The Chatbot gives recommendation of the user's product</p> |

| | | |
|-----------|---------------------------------------|--|
| | | <p>and suggests available offers of the product.</p> <p>The introduction of chatbot makes an interactive website and boosts the productivity of the product.</p> |
| 3. | Novelty / Uniqueness | <p>Many online shopping application is mainly based on the search engine while this project make use of interactive chatbot like alexa,siri etc</p> |
| 4. | Social Impact / Customer Satisfaction | <p>This project uses chatbots that converse with the users and ask questions that encourage them to make a purchase. If a customer is searching for a product, they need not leave the website to know more about it. Chatbots pop up and deliver the information to the users effortlessly.</p> |
| 5. | Business Model (Revenue Model) | <p>There are many systesm available in the recommendation systems.Our Project will focus on improving customer</p> |

| | | |
|----|-----------------------------|--|
| | | experience in online shopping by embedding an interactive chatbot in it which will improve the productivity and boosts the economy |
| 6. | Scalability of the Solution | Our project solution is platform independent. In future various machine learning algorithms can be applied on the chatbot and UI interfaces of webapp can be developed and modified. |

3.4 Problem Solution fit

| | | | | |
|--|--|--|---|--|
| Define CS, fit into CC | 1. CUSTOMER SEGMENT(S) Age, gender, location, browsing habits, interests and even the devices we use, say a lot about our shopping habits. They influence what we buy, why we buy, and how often we spend money. | 6. CUSTOMER CONSTRAINTS <ul style="list-style-type: none"> Quality issues. Delivery and logistics Digital payment failures. Additional charges. Unclear return and guarantee policies Lack of security. | 5. AVAILABLE SOLUTIONS <ul style="list-style-type: none"> Generally, the search bar option is available when a customer or user needs to find the desired product. However, occasionally this can lead to the user or customer finding a product that is unrelated to what they were looking for. | Explore AS, differential |
| | 2. JOBS-TO-BE-DONE / PROBLEMS <ul style="list-style-type: none"> Navigating between various screens to make an online purchase. This one is the grumpy one. Typically, e-commerce features include searching for a user's product may take more time. | 9. PROBLEM ROOT CAUSE <ul style="list-style-type: none"> Making an online purchase while navigating between numerous screens. The grumpy one is the one. When a customer visits your e-commerce website, they expect to easily and quickly locate what they're looking for. | 7. BEHAVIOUR <ul style="list-style-type: none"> A key priority convenience. Simple access on all devices. Availability of omnichannel shopping Uncomplicated payment Quick and dependable shipping. | |
| Focus on J&P, tap into BE, understand RC | 3. TRIGGERS <ul style="list-style-type: none"> Utilize the personal touch. Encourage Loyalty | 10. YOUR SOLUTION <ul style="list-style-type: none"> You can directly do your online shopping based on your choice without any search. It can be done by using a chatbot. User recommendations can be made by the chatbot depending on their interests. It may advertise their day's top specials and promotions. It will keep a database of the customer's information and orders. If the order is accepted, the chatbot will | 8. CHANNELS OF BEHAVIOUR ONLINE <ul style="list-style-type: none"> In order to purchase a specific commodity or service, the customer must first register online at the relevant website. A customer adds a good or service to their "shopping cart" when they like it. OFFLINE <ul style="list-style-type: none"> Warehouseman collects products specified in an order. | Focus on J&P, tap into BE, understand RC |
| | | | | |

| | | |
|--|-----------------------------|--|
| <p>4. EMOTIONS: BEFORE / AFTER</p> <p>BEFORE</p> <p>The search bar option is available when a customer or user needs to find the desired product.</p> <p>AFTER</p> <p>Chatbot is like talking to a human which may make it easier for people to find the product.</p> | <p>notify the customer.</p> | |
|--|-----------------------------|--|

4.

REQUIREMENT ANALYSIS

4.1 Functional requirement

| FR No. | Functional Requirement (Epic) | Sub Requirement (Story / Sub-Task) |
|--------|-------------------------------|--|
| FR-1 | User Registration / Sign up | Registration through Form Registration through Gmail Registration through LinkedIN |
| FR-2 | User Verification | Confirmation via Email Confirmation via OTP |
| FR-3 | Sign In / Login | Login by using Mobile Number or Email |
| FR-4 | Profile Details | Update the Information about Customer Example :- Name Gender Age Mobile number Address |
| FR-5 | Chatbot (Watson Assistant) | Get the Information about Search Products View Offers Discounts |

| | | |
|-------|--|---|
| | | <p>Stock Availability</p> <p>User Personal Information (FR-4)</p> |
| FR-6 | Advance Search Capabilities | Sorting and filtering options |
| FR-7 | Shopping Cart | <p>My Cart Button Add-To-Cart- Button Remove-From-Cart- Button</p> |
| FR-8 | Checking Item Availability | Item Availability in rural and urban Locations |
| FR-9 | Checking The Shipping Status / Tracking The Order Product | Easily Checking Status availability of ordered Items |
| FR-10 | Logout | <p>After the Purchase, user can Logout Or close the application When customer needs</p> |

4.2 Non-Functional requirements

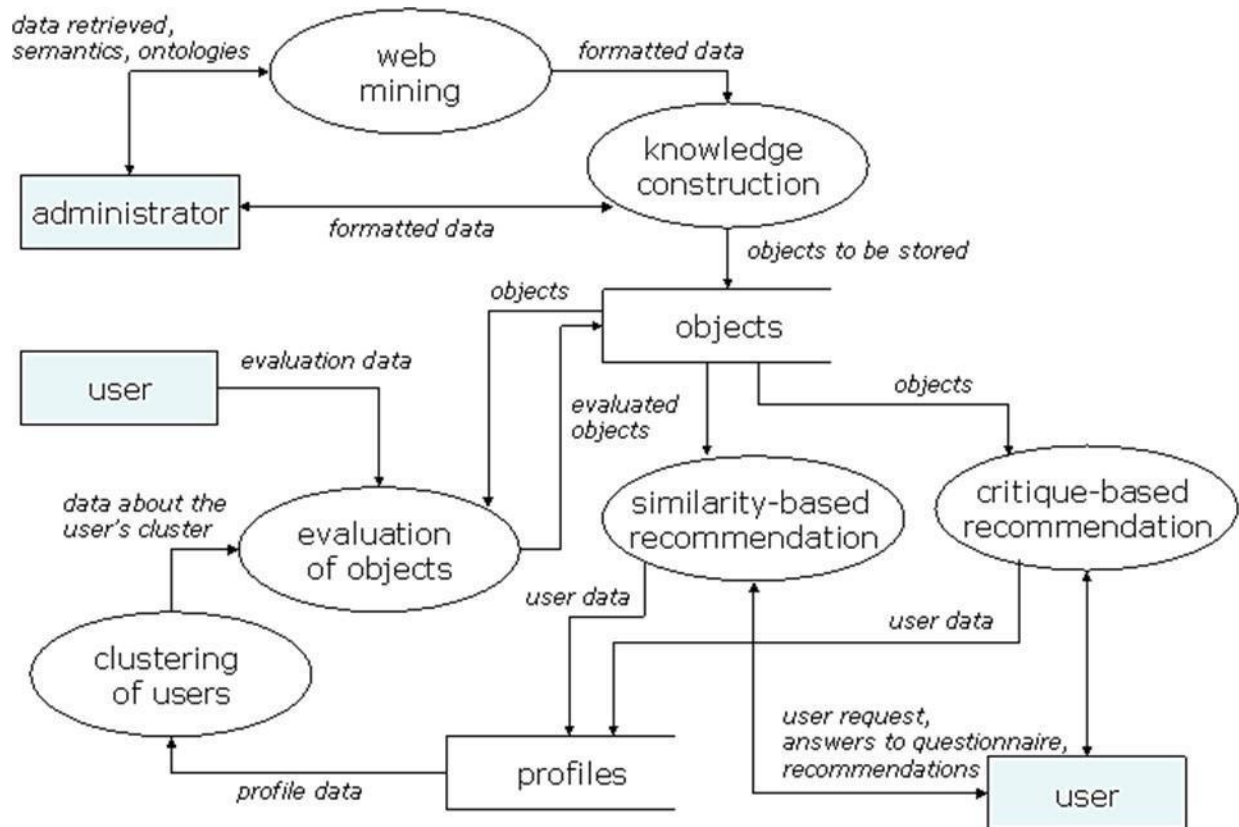
| NFR No. | Non-Functional Requirement (Epic) | Description |
|--------------|-----------------------------------|---|
| NFR-1 | usability | <p>The application will be designed for making Good HUMAN – COMPUTER Interaction in such a way,</p> <p>Any user can easily navigate User Can easily View and understandable</p> <p>Comfort while making Place order</p> <p>Comfort with tracking facilities Easy and Compact design</p> <p>These all are about to achieve a defined goal Effectively, Efficiently and Satisfactorily.</p> |
| NFR-2 | Security | <p>The application will be Using of “Secure Socket Layer” (SSL) Certificate will provide a More security of the Project and This</p> |

| | | |
|--------------|---------------------|--|
| | | process will happen while Python Flask to Cloud Connect. This makes user private Information like Baking, Shipping/Home address, email, Phone number etc., will be kept as more secure. |
| NFR-3 | Reliability | Ability of software to perform critical tasks like Collection and Securing customer Data, Providing Gateway Payment to function correctly in a given Environment, for a Particular amount of time. |
| NFR-4 | Performance | It Focus on the loading application as quickly as possible irrespective of the number of user traffic |
| NFR-5 | Availability | The Application will be Available to all users at any given point of time. User can access the chatbot for raising any queries. |
| NFR-6 | Scalability | Chatbot can be very useful to know about offers and discounts. It will be helpful whenever we make online shopping |

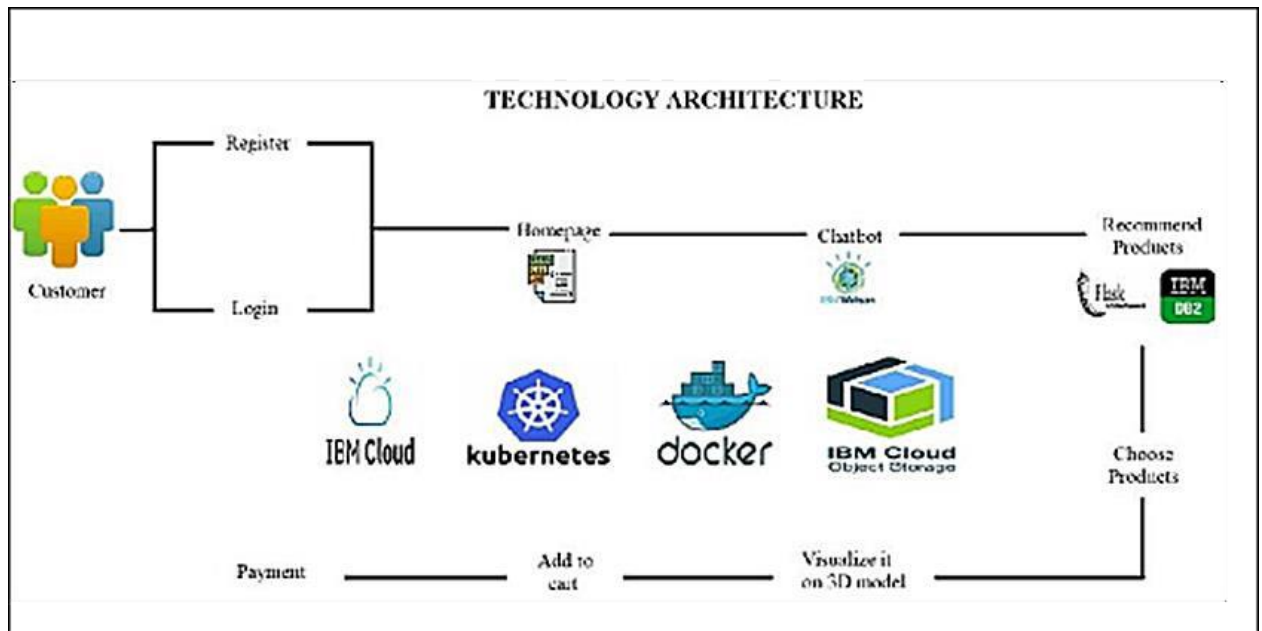
5.

PROJECT DESIGN

5.1 Data Flow Diagrams



5.2 Solution & Technical Architecture



5.3 User Stories

| User Type | Functional Requirement (Epic) | User Story Number | User Story / Task | Acceptance criteria | Priority | Release |
|------------------------|-------------------------------|-------------------|---|-------------------------------------|----------|----------|
| Customer (Mobile user) | Registration | USN-1 | As a user, I can register for the application by entering my email, password, | I can access my account / dashboard | High | Sprint-1 |

| | | | | | | |
|--|--|-------|---|---|------|----------|
| | | | and confirming my password. | | | |
| | | USN-2 | As a user, I will receive confirmati on email once I have registered for the application | I can receive confirm ation email & click confirm | High | Sprint-1 |
| | | USN-3 | As a user, I can register for the | I can register &access | Low | Sprint-2 |

| | | | | | | |
|--|-----------|-------|--|---|--------|----------|
| | | | application through Facebook | the dashboa rd with Gmail login | | |
| | | USN-4 | As a user, I can register for the application through Gmail | I can register &access the dashboa rd with Gmail login | Medium | Sprint-1 |
| | Login | USN-5 | As a user, I can log into the application by entering email & password | I can login into the applicati on with Gmail login | High | Sprint-1 |
| | Dashboard | USN-5 | As a user ,I can log access the | I can access | High | Sprint-1 |

| | | | | | | |
|--|--|--|--|--|--|--|
| | | | dashboard of the application by logging into the application | the dashboa rd by logging into the applicati on | | |
|--|--|--|--|--|--|--|

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

| MILESTONES | ACTIVITY | DESCRIPTION |
|--------------------------------|--|---|
| Project development phase | Delivery of Sprint-1,2, 3, 4 | To develop the code and submit the develop the code after completion of testing |
| Setting-up app environment | Create IBM cloud account | Sign up IBM cloud account |
| | Create flask project | Getting started with the flask to create project |
| | Install IBM cloud CLI | Install IBM command line interface (CLI) |
| | Docker CLI installation | Installing docker CLI |
| | Create an account in sendgrid | Create an account in sendgrid Use service as e-mail integration to the application for sending emails |
| Implementing web application | Create UI to interact with the application | Create UI <ul style="list-style-type: none">✓ registration page✓ login page✓ view products page✓ add products page |
| | Create IBM DB2 and connect with the Python | Create IBM DB2 in IBM cloud and link with the Python |
| Integrating sendgrid service | Sendgrid integration with the Python | To send emails from the application we need to integrate the sendgrid services |
| Developing a chat bot | Building a chat bot and integrate with the application | Build the chat bot and integrate it to the flask application |
| Deployment of app in IBM cloud | Containerise the app | Create a docker image of the application in addition to push it to the IBM container registry |
| | Upload image to IBM container registry | Upload the image to IBM container registry |
| | Deploy in in kubernetes cluster | Once the image is uploaded to IBM container registry deploy the image toward IBM kubernetes cluster |

| MILESTONES | ACTIVITY | DESCRIPTION |
|------------------------|------------------------------|---|
| Ideation phase | Literature survey | Literature survey on the selected project and information gathering |
| | Empathy map | Prepare empathy map to capture the user pains and gains, prepare a list of problem statement |
| | Ideation | Organising the brainstorming session and prioritise the top three ideas based on feasibility and importance |
| Project design phase 1 | Proposed solution | Prepare proposed solution document which includes novelty, feasibility of ideas, business model, social impact, scalability of solution |
| | Problem solution fit | Prepare problem solution fit documents |
| | Solution architecture | Prepare solution architecture document |
| Project design phase 2 | Customer journey map | Prepare customer journey map to understand the user interactions and experience with the application |
| | Functional requirements | Prepare functional and non-functional necessity document |
| | Data flow diagram | Prepare data flow diagram and user stories |
| | Technology architecture | Draw technology architecture diagram |
| Project planning phase | Milestones and activity list | Prepare milestones and activity list of the project |
| | Sprint delivery plan | Prepare spring delivery plan |

6.2 Sprint Delivery Schedule

| Sprint | Functional Requirement (Epic) | User Story Number | User Story / Task | Story points | Priority | Team Members |
|----------|-------------------------------|-------------------|--|--------------|----------|---|
| Sprint-1 | Setting up App environment | USN-1 | As a user, I can register in ICTA Academy and create IBM cloudaccount. | 2 | High | S.Parameshwari B.Sandhiya C.Saranya |
| Sprint-1 | | USN-2 | As a user, I will create a flaskproject | 1 | Low | J.Rilwana Parveen S.Suriya Jothi |
| Sprint-1 | | USN-3 | As a user, I will install IBM CloudCLI | 2 | Medium | S.Parameshwari B.Sandhiya C.Saranya |
| Sprint-2 | Setting up App environment | USN-4 | As a user, I can install Docker CLI | 1 | Low | J.Rilwana Parveen S.Suriya Jothi |
| Sprint-2 | | USN-5 | As a user, I will Create an accountin sendgrid | 2 | Medium | S.Parameshwari B.Sandhiya C.Saranya |

| | | | | | | |
|----------|---------------------------------|--------|---|---|--------|--|
| Sprint-3 | Implementing web application | USN-6 | As a user, I Create UI to interact with the application | 1 | High | J.Rilwana Parveen S.Suriya Jothi |
| Sprint-3 | | USN-7 | As a user, I Create IBM DB2 and connect with Python | 3 | High | S.Parameshwari B.Sandhiya C.Saranya |
| Sprint-3 | Integrating sendgrid service | USN-8 | As a user, I will be integrating sendgridwith python code | 2 | High | J.Rilwana Parveen S.Suriya Jothi |
| Sprint-3 | Developing a chatbot | USN-9 | As a user, I must build a chatbot and integrate to application | 1 | Medium | S.Parameshwari B.Sandhiya C.Saranya |
| Sprint-4 | Development of App in IBM Cloud | USN-10 | As a user, I will Containerize the App | 1 | Low | J.Rilwana Parveen S.Suriya Jothi |
| Sprint-4 | | USN-11 | As a user, I will upload image to IBM Container registry | 2 | Medium | S.Parameshwari B.Sandhiya C.Saranya |
| Sprint-4 | | USN-12 | As a user, I will deploy App in Kubernetes cluster | 3 | High | J.Rilwana Parveen S.Suriya Jothi |
| Sprint-4 | User panel | | As a user <ul style="list-style-type: none"> ● Register, Login, Email, Verification ● Manual Search ● Order placement, Order Details | 3 | High | J.Rilwana Parveen S.Suriya Jothi S.Parameshwari B.Sandhiya C.Saranya |

Project Tracker, Velocity & Burndown Chart

Project Tracker, Velocity & Burndown Chart

| 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 | 18 | 6 Days | 24 Oct 2022 | 29 Oct 2022 | 24 | 29 Oct 2022 |
| Sprint-2 | 18 | 6 Days | 31 Oct 2022 | 05 Nov 2022 | 24 | 05 Nov 2022 |
| Sprint-3 | 18 | 6 Days | 07 Nov 2022 | 12 Nov 2022 | 24 | 12 Nov 2022 |
| Sprint-4 | 18 | 6 Days | 14 Nov 2022 | 19 Nov 2022 | 24 | 19 Nov 2022 |

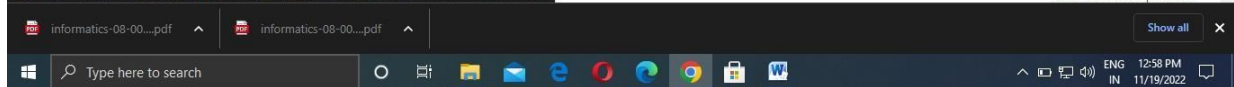
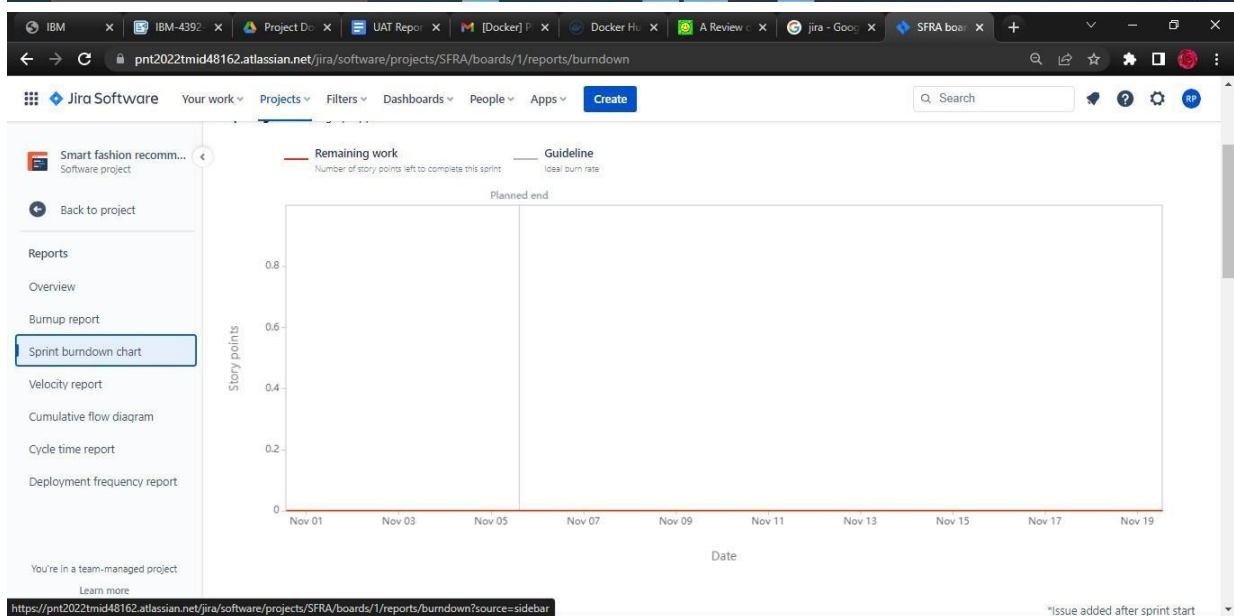
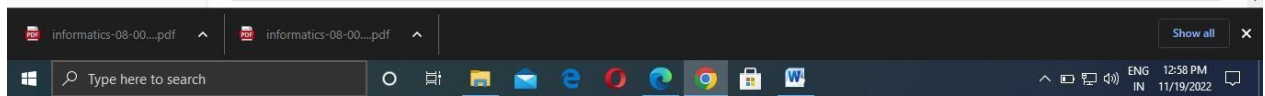
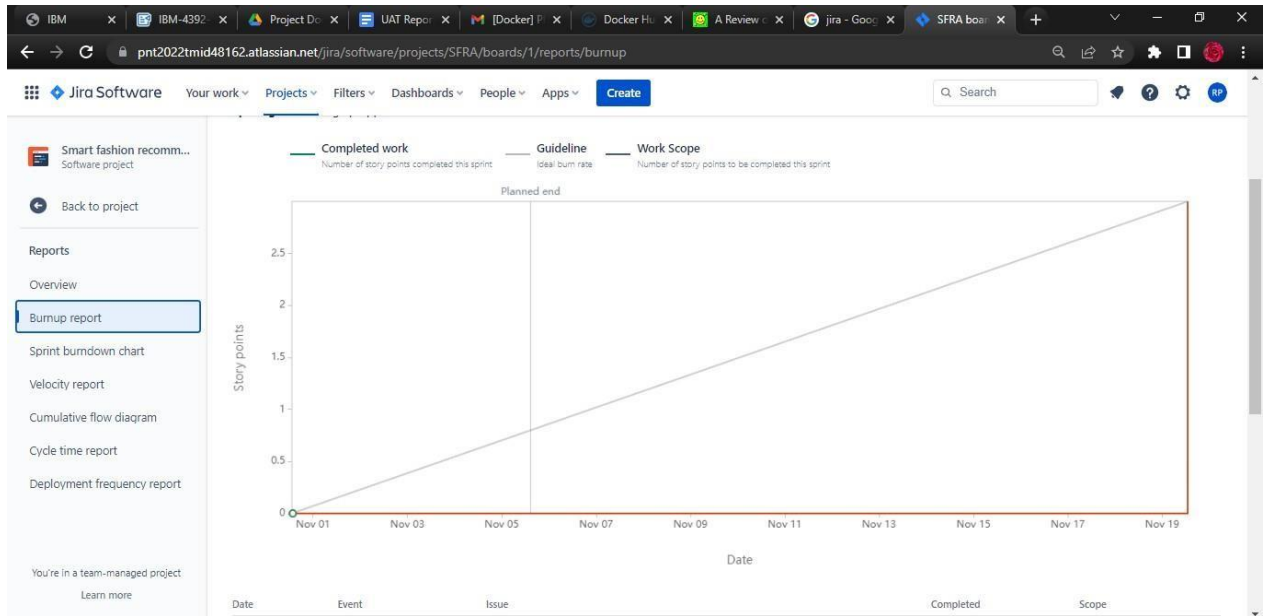
Velocity

Imagine we have a 6-day sprint duration, and the velocity of the team is 18(points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

$$AV = \text{Sprint Duration} / \text{Velocity}$$

$$AV = 24/6 = 4$$

6.3 Reports from JIRA



7. CODING & SOLUTIONING

7.1 Feature 1

User can view the product on home page

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Smart Fashion Recommender Application</title>
  <link rel="stylesheet" href="static/css/home.css">
</head>
<body>

  <nav class="navbar">

</nav>

  <!-- hero section -->
<header class="hero-section">
  <div class="content">
    
```

```
<p class="sub-heading">best fashion collection of all time</p>
</div>
</header>
<section class="product">
  <h2 class="product-category">best selling</h2>
  <button class="pre-btn"></button>
  <button class="nxt-btn"></button>
  <div class="product-container">
    <div class="product-card">
      <div class="product-image">
        <span class="discount-tag">50% offer</span>
        
        <button class="card-btn">add to whislist</button>
      </div>
      <div class="product-info">
        <h2 class="product-brand">brand</h2>
        <p class="product-short-des">a short line about the cloth.</p>
        <span class="price">$20</span><span class="actual-price">$40</span>
      </div>
    </div>
    <div class="product-card">
      <div class="product-image">
        <span class="discount-tag">50% offer</span>
        
        <button class="card-btn">add to whislist</button>
      </div>
      <div class="product-info">
        <h2 class="product-brand">brand</h2>
```

```
<p class="product-short-des">a short line about the cloth.</p>
<span class="price">$20</span><span class="actual-price">$40</span>
</div>
</div>
<div class="product-card">
  <div class="product-image">
    <span class="discount-tag">50% offer</span>
    
    <button class="card-btn">add to whislist</button>
  </div>
  <div class="product-info">
    <h2 class="product-brand">brand</h2>
    <p class="product-short-des">a short line about the cloth.</p>
    <span class="price">$20</span><span class="actual-price">$40</span>
  </div>
</div>
<div class="product-card">
  <div class="product-image">
    <span class="discount-tag">50% offer</span>
    
    <button class="card-btn">add to whislist</button>
  </div>
  <div class="product-info">
```



```
<h2 class="product-brand">brand</h2>
<p class="product-short-des">a short line about the cloth.</p>
<span class="price">$20</span><span class="actual-price">$40</span>
</div>
</div>
<div class="product-card">
  <div class="product-image">
    <span class="discount-tag">50% offer</span>
    
    <button class="card-btn">add to whislist</button>
  </div>
  <div class="product-info">
    <h2 class="product-brand">brand</h2>
    <p class="product-short-des">a short line about the cloth.</p>
    <span class="price">$20</span><span class="actual-price">$40</span>
  </div>
</div>
<div class="product-card">
  <div class="product-image">
    <span class="discount-tag">50% offer</span>
    
    <button class="card-btn">add to whislist</button>
  </div>
  <div class="product-info">
    <h2 class="product-brand">brand</h2>
    <p class="product-short-des">a short line about the cloth.</p>
    <span class="price">$20</span><span class="actual-price">$40</span>
  </div>
</div>
```

</div>

<div class="product-card">

<div class="product-image">

50% offer

<button class="card-btn">add to whislist</button>

</div>

<div class="product-info">

<h2 class="product-brand">brand</h2>

<p class="product-short-des">a short line about the cloth.</p>

\$20\$40

</div>

</div>

<div class="product-card">

<div class="product-image">

50% offer

<button class="card-btn">add to whislist</button>

</div>

<div class="product-info">

<h2 class="product-brand">brand</h2>

<p class="product-short-des">a short line about the cloth.</p>

\$20\$40

```

        </div>
    </div>
</div>
</section>
<!-- collections -->
<section class="collection-container">
    <a href="#" class="collection">
        
        <p class="collection-title">women <br> apparels</p>
    </a>
    <a href="#" class="collection">
        
        <p class="collection-title">men <br> apparels</p>
    </a>
    <a href="#" class="collection">
        
        <p class="collection-title">accessories</p>
    </a>
</section>
<section class="product">
    <h2 class="product-category">shirts</h2>
    <button class="pre-btn"></button>
<button class="nxt-btn"></button>
    <div class="product-container">
        <div class="product-card">
            <div class="product-image">
                <span class="discount-tag">50% offer</span>
                

<button class="card-btn">add to whislist</button>

</div>

<div class="product-info">

<h2 class="product-brand">brand</h2>

<p class="product-short-des">a short line about the cloth.</p>

<span class="price">\$20</span><span class="actual-price">\$40</span>

</div>

</div>

<div class="product-card">

<div class="product-image">

<span class="discount-tag">50% offer</span>



<button class="card-btn">add to whislist</button>

</div>

<div class="product-info">

<h2 class="product-brand">brand</h2>

<p class="product-short-des">a short line about the cloth.</p>

<span class="price">\$20</span><span class="actual-price">\$40</span>

</div>

</div>

```
<div class="product-card">
 <div class="product-image">
 50% offer

 <button class="card-btn">add to whislist</button>
 </div>
 <div class="product-info">
 <h2 class="product-brand">brand</h2>
 <p class="product-short-des">a short line about the cloth.</p>
 $20$40
 </div>
</div>
```

```
<div class="product-card">
 <div class="product-image">
 50% offer

 <button class="card-btn">add to whislist</button>
 </div>
 <div class="product-info">
 <h2 class="product-brand">brand</h2>
 <p class="product-short-des">a short line about the cloth.</p>
 $20$40
 </div>
</div>
<div class="product-card">
 <div class="product-image">
```

```
50% offer

<button class="card-btn">add to whislist</button>
</div>
<div class="product-info">
 <h2 class="product-brand">brand</h2>
 <p class="product-short-des">a short line about the cloth.</p>
 $20$40
</div>
</div>

<div class="product-card">
 <div class="product-image">
 50% offer

 <button class="card-btn">add to whislist</button>
 </div>
 <div class="product-info">
 <h2 class="product-brand">brand</h2>
 <p class="product-short-des">a short line about the cloth.</p>
 $20$40
 </div>
</div>
```

```
<div class="product-card">
 <div class="product-image">
 50% offer

 <button class="card-btn">add to whislist</button>
 </div>
 <div class="product-info">
 <h2 class="product-brand">brand</h2>
 <p class="product-short-des">a short line about the cloth.</p>
 $20$40
 </div>
</div>
```

```
<div class="product-card">
 <div class="product-image">
 50% offer

 <button class="card-btn">add to whislist</button>
 </div>
 <div class="product-info">
 <h2 class="product-brand">brand</h2>
 <p class="product-short-des">a short line about the cloth.</p>
 $20$40
 </div>
</div>
```

```
<div class="product-card">
 <div class="product-image">
 50% offer
```

```

<button class="card-btn">add to whislist</button>
</div>
<div class="product-info">
 <h2 class="product-brand">brand</h2>
 <p class="product-short-des">a short line about the cloth.</p>
 $20$40
</div>
</div>
```

```
</div>
</section>
<section class="product">
 <h2 class="product-category">shoes</h2>
 <button class="pre-btn"></button>
 <button class="nxt-btn"></button>
 <div class="product-container">
 <div class="product-card">
 <div class="product-image">
 50% offer

 <button class="card-btn">add to whislist</button>
```



</div>

<div class="product-info">

<h2 class="product-brand">brand</h2>

<p class="product-short-des">a short line about the cloth.</p>

<span class="price">\$20</span><span class="actual-price">\$40</span>

</div>

</div>

<div class="product-card">

<div class="product-image">

<span class="discount-tag">50% offer</span>



<button class="card-btn">add to whislist</button>

</div>

<div class="product-info">

<h2 class="product-brand">brand</h2>

<p class="product-short-des">a short line about the cloth.</p>

<span class="price">\$20</span><span class="actual-price">\$40</span>

</div>

</div>

<div class="product-card">

<div class="product-image">

<span class="discount-tag">50% offer</span>



<button class="card-btn">add to whislist</button>

</div>

<div class="product-info">

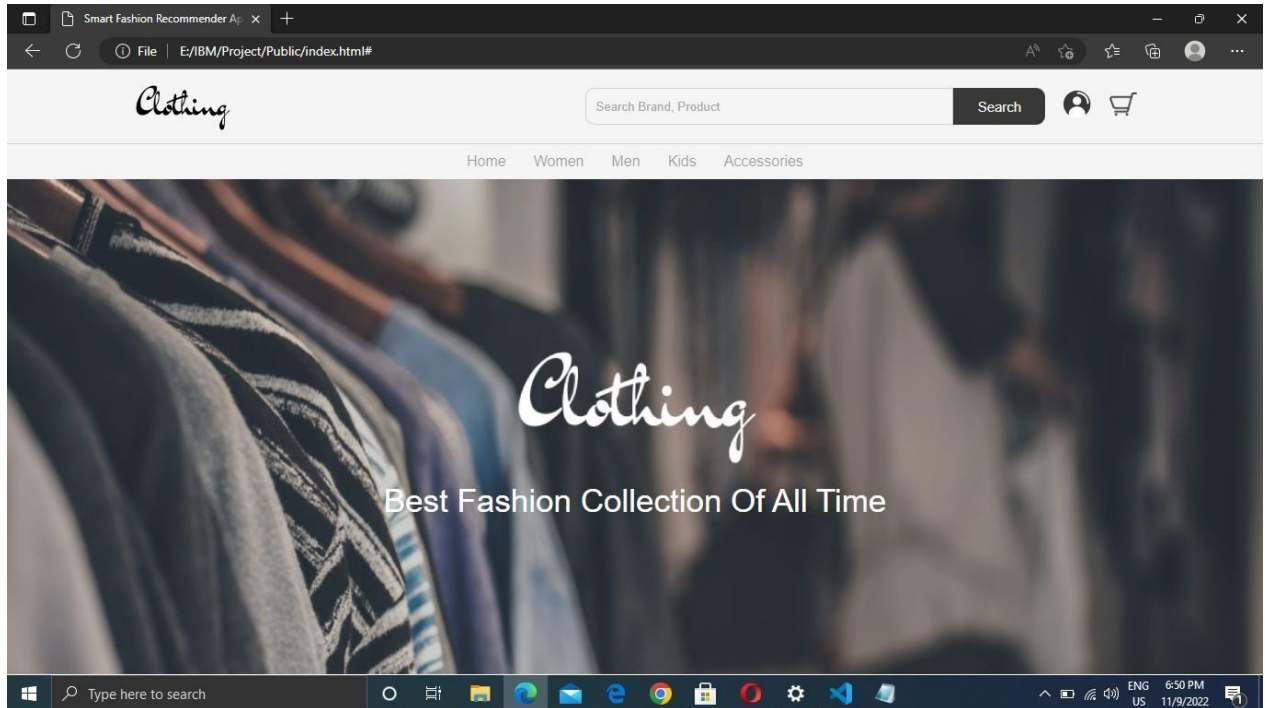
<h2 class="product-brand">brand</h2>

<p class="product-short-des">a short line about the cloth.</p>

```
 $20$40
 </div>
</div>
<div class="product-card">
 <div class="product-image">
 50% offer

 <button class="card-btn">add to whislist</button>
 </div>
 <div class="product-info">
 <h2 class="product-brand">brand</h2>
 <p class="product-short-des">a short line about the cloth.</p>
 $20$40
 </div>
</div>
</div>
</section>
<footer>
</footer>
<script src="static/js/nav.js"></script>
<script src="static/js/home.js"></script>
<script src="static/js/footer.js"></script>
</body>
```

</html>



## 7.1 Feature 2

Using chat bot we can manage user's choices and orders.

```
from flask import Flask, render_template
```

```
import os
```

```
app = Flask(__name__, template_folder='Templates', static_folder='static')
```

```
@app.route('/')
```

```
def index():
```

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

```
@app.route('/product')
```

```
def product():
```

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

```
@app.route('/search')
```

```
def search():
```

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

```
@app.route('/404')
```

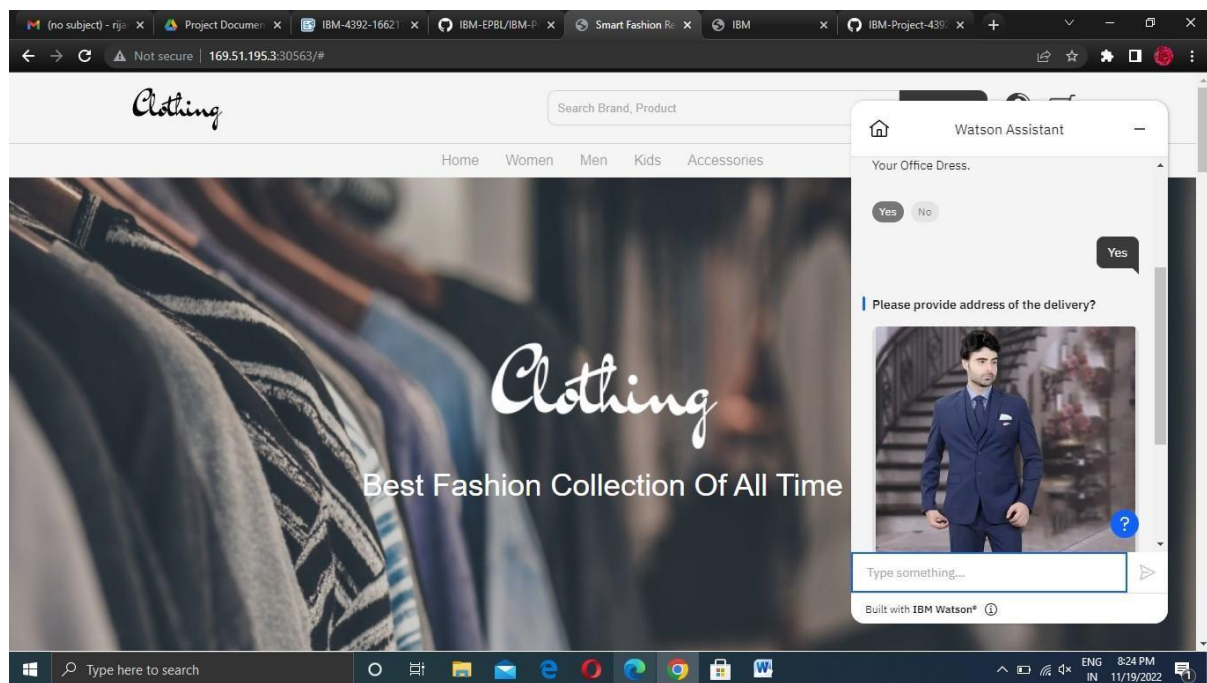
```
def error():
```

```
 return render_template('404.html')
```

```
if __name__ == '__main__':
```

```
 port = int(os.environ.get('PORT', 5000))
```

```
 app.run(debug=True, host='0.0.0.0', port=port)
```



## 8. TESTING

### 8.1 Test Cases

This report shows the number of test cases that have passed, failed, and untested

| Section             | Total Cases | NotTested | Fail | Pass |
|---------------------|-------------|-----------|------|------|
| Login               | 5           | 0         | 0    | 5    |
| Register            | 7           | 0         | 0    | 7    |
| Home Page           | 2           | 0         | 0    | 2    |
| Order page          | 3           | 0         | 0    | 3    |
| Order products      | 9           | 0         | 0    | 9    |
| Final Report Output | 4           | 0         | 0    | 4    |
| Version Control     | 2           | 0         | 0    | 2    |



| Expected Result                                                                                                                                                                                              | Actual Result       | Status | Comments | TC for Automation(Y/N) | BUG ID | Executed By |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------|--------|----------|------------------------|--------|-------------|
| Login/Signup popup should display                                                                                                                                                                            | Working as expected | Pass   |          |                        |        |             |
| Application should show below UI elements:<br>a.email text box<br>b.password text box<br>c.Login button with orange colour<br>d.New customer? Create account link<br>e.Last password? Recovery password link | Working as expected | Pass   |          |                        |        |             |
| User should navigate to user account homepage                                                                                                                                                                | working as expected | Pass   |          |                        |        |             |
| Application should show 'Incorrect email or password ' validation message.                                                                                                                                   | Working as expected | Pass   |          |                        |        |             |
| Application should show 'Incorrect email or password ' validation message.                                                                                                                                   | working as expected | Pass   |          |                        |        |             |
| Application should show 'Incorrect email or password ' validation message.                                                                                                                                   | working as expected | Pass   |          |                        |        |             |
| Application will send replay as dress                                                                                                                                                                        | Working as expected | Pass   |          |                        |        |             |





|                       |    |    |    |    |    |
|-----------------------|----|----|----|----|----|
| By Design             | 5  | 5  | 2  | 3  | 21 |
| Duplicate             | 1  | 0  | 3  | 0  | 4  |
| External              | 2  | 3  | 0  | 1  | 6  |
| Fixed                 | 11 | 2  | 4  | 20 | 37 |
| Not<br>Reproduc<br>ed | 0  | 0  | 1  | 0  | 1  |
| Skipped               | 0  | 0  | 1  | 1  | 2  |
| Won't Fix             | 0  | 5  | 2  | 1  | 8  |
| Totals                | 24 | 14 | 13 | 26 | 77 |

# 9. RESULTS

## 9.1 Performance Metrics

|                                                                                                                              |                                       |                    |                                           |                       |                         |                      |                                           |                             |
|------------------------------------------------------------------------------------------------------------------------------|---------------------------------------|--------------------|-------------------------------------------|-----------------------|-------------------------|----------------------|-------------------------------------------|-----------------------------|
| performance template for Internet of Things & Cloud Application DevelopmentLab - Microsoft Excel (Product Activation Failed) |                                       |                    |                                           |                       |                         |                      |                                           |                             |
| Rihwana parveen                                                                                                              |                                       |                    |                                           |                       |                         |                      |                                           |                             |
| S.No                                                                                                                         | Project Name                          | Functional Changes | Hardware Changes                          | Software Changes      | Impact of Downtime      | Load/Volumen Changes | Risk Score                                | Justification               |
| 1                                                                                                                            | smart fashion recommender application | New                | Low                                       | No Changes            | Moderate                | >5 to 10%            | ORANGE                                    | As we have seen the changes |
| NFT - Risk Assessment                                                                                                        |                                       |                    |                                           |                       |                         |                      |                                           |                             |
| NFT - Detailed Test Plan                                                                                                     |                                       |                    |                                           |                       |                         |                      |                                           |                             |
| S.No                                                                                                                         | Project Overview                      | NFT Test approach  | Assumptions/Dependencies/Risks            | Approval/SignOff      |                         |                      |                                           |                             |
| 1                                                                                                                            | smart fashion recommender application | manual testing     | laptop or mobile with internet connection | Rihwana parveen       |                         |                      |                                           |                             |
| End Of Test Report                                                                                                           |                                       |                    |                                           |                       |                         |                      |                                           |                             |
| S.No                                                                                                                         | Project Overview                      | NFT Test approach  | NFT - Mgr                                 | Test Outcome          | SOUND-GO decision       | Recommendations      | Identified Defects (Detected/Closed/Open) | Approval/SignOff            |
| 1                                                                                                                            | smart fashion recommender application | Manual             |                                           | Worked as we expected | Use laptop/desktop mode | No defects           |                                           | Rihwana parveen             |

## **10 ADVANTAGES & DISADVANTAGES**

### **ADVANTAGES**

Chatbots can gather Customer Insights.

Reduction in Costs of Human Resource

Interactive marketing guide.

Increase in number of customers.

Anytime anywhere.

Customer satisfaction.

Quick response.

### **DISADVANTAGES**

Less contact of customers.

Lagging of customer satisfaction.

No quick response.

Waste of time for customer to to buy product easily.

Can't answer the multipart question.

**10.**

## **CONCLUSION**

This project smart fashion recommender application has been developed using python flask. This bot has the ability to have conversation with the consumers knowing , understanding their needs and problem and gives answer to all what They need. Thus, site of proposed marketing guide gives us better bot performance in marketing field . Thus a higher system performance is achieved.The website provides a computerized version of marketing which will benefit in the product of the company. It makes entire process easy .

## 11.

## FUTURE SCOPE

Future chatbots can communicate at multiple levels with automation at the system level. They have the ability to maintain the system, task, and people contexts. There is a possibility of introduction of master bots and eventually chatbot OS. The future of chatbots is that businesses will automate simple payments

and allow users to pay directly over live chat or Facebook Messenger apps. The instant process makes the customer happy and improves customer satisfaction.

MasterCard has also launched a chatbot, especially for customer payments. Voice bots are becoming mainstream. Voice is the next big thing!. Users are already used to starting their days with “Ok Google, what’s in my calendar today?”.

be voice-driven”. It is a greatly emerging conversational banking trend. It’s all about facilitating your users with a seamless experience with your business, and you can achieve that by voice-driven chatbots. Accenture research says “Digital consumers prefer messaging platforms that have a voice and text-based interfaces .

## 13. APPENDIX

### Source Code

#### Index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="UTF-8">
 <meta http-equiv="X-UA-Compatible" content="IE=edge">
 <meta name="viewport" content="width=device-width, initial-
scale=1.0">
 <title>Smart Fashion Recommender Application</title>
 <link rel="stylesheet" href="static/css/home.css">
</head>
<body>

 <nav class="navbar">

</nav>
```

```
<!-- hero section -->
<header class="hero-section">
 <div class="content">

 <p class="sub-heading">best fashion collection of all time</p>
 </div>
</header>
<section class="product">
 <h2 class="product-category">best selling</h2>
 <button class="pre-btn"></button>
<button class="nxt-btn"></button>
 <div class="product-container">
 <div class="product-card">
 <div class="product-image">
 50% offer

 <button class="card-btn">add to whislist</button>
```

```
</div>

<div class="product-info">
 <h2 class="product-brand">brand</h2>
 <p class="product-short-des">a short line about the cloth.</p>
 $20<span class="actual-
price">$40
</div>

</div>

<div class="product-card">
 <div class="product-image">
 50% offer

 <button class="card-btn">add to whislist</button>
 </div>
 <div class="product-info">
 <h2 class="product-brand">brand</h2>
 <p class="product-short-des">a short line about the cloth.</p>
 $20<span class="actual-
price">$40
 </div>
</div>

<div class="product-card">
 <div class="product-image">
```



```
50% offer

<button class="card-btn">add to whislist</button>
</div>
<div class="product-info">
 <h2 class="product-brand">brand</h2>
 <p class="product-short-des">a short line about the cloth.</p>
 $20<span class="actual-
price">$40
</div>
</div>
<div class="product-card">
 <div class="product-image">
 50% offer

 <button class="card-btn">add to whislist</button>
 </div>
 <div class="product-info">
```

```
<h2 class="product-brand">brand</h2>
<p class="product-short-des">a short line about the cloth.</p>
$20<span class="actual-
price">$40
</div>
</div>
<div class="product-card">
 <div class="product-image">
 50% offer

 <button class="card-btn">add to whislist</button>
 </div>
 <div class="product-info">
 <h2 class="product-brand">brand</h2>
 <p class="product-short-des">a short line about the cloth.</p>
 $20<span class="actual-
price">$40
 </div>
</div>
<div class="product-card">
 <div class="product-image">
 50% offer

 <button class="card-btn">add to whislist</button>
</div>
<div class="product-info">
 <h2 class="product-brand">brand</h2>
 <p class="product-short-des">a short line about the cloth.</p>
 $20<span class="actual-
price">$40
 </div>
</div>
<div class="product-card">
 <div class="product-image">
 50% offer

 <button class="card-btn">add to whislist</button>
 </div>
 <div class="product-info">
 <h2 class="product-brand">brand</h2>
 <p class="product-short-des">a short line about the cloth.</p>
```

```
 $20<span class="actual-
price">$40
 </div>
</div>
<div class="product-card">
 <div class="product-image">
 50% offer

 <button class="card-btn">add to whislist</button>
 </div>
 <div class="product-info">
 <h2 class="product-brand">brand</h2>
 <p class="product-short-des">a short line about the cloth.</p>
 $20<span class="actual-
price">$40
 </div>
</div>
</div>
</section>
<!-- collections -->
<section class="collection-container">


```

```
<p class="collection-title">women
 apparels</p>

 <p class="collection-title">men
 apparels</p>

 <p class="collection-title">accessories</p>

</section>
<section class="product">
 <h2 class="product-category">shirts</h2>
 <button class="pre-btn"></button>
<button class="nxt-btn"></button>
 <div class="product-container">
 <div class="product-card">
 <div class="product-image">
```

```
50% offer

<button class="card-btn">add to whislist</button>
</div>
<div class="product-info">
 <h2 class="product-brand">brand</h2>
 <p class="product-short-des">a short line about the cloth.</p>
 $20<span class="actual-
price">$40
</div>
</div>
```

```
<div class="product-card">
 <div class="product-image">
 50% offer

 <button class="card-btn">add to whislist</button>
 </div>
 <div class="product-info">
 <h2 class="product-brand">brand</h2>
 <p class="product-short-des">a short line about the cloth.</p>
 $20<span class="actual-
```

price">\$40</span>

</div>

</div>

<div class="product-card">

<div class="product-image">

<span class="discount-tag">50% offer</span>



<button class="card-btn">add to whislist</button>

</div>

<div class="product-info">

<h2 class="product-brand">brand</h2>

<p class="product-short-des">a short line about the cloth.</p>

<span class="price">\$20</span><span class="actual-price">\$40</span>

</div>

</div>

<div class="product-card">

<div class="product-image">

```
50% offer

<button class="card-btn">add to whislist</button>
</div>
<div class="product-info">
 <h2 class="product-brand">brand</h2>
 <p class="product-short-des">a short line about the cloth.</p>
 $20<span class="actual-
price">$40
</div>
</div>
<div class="product-card">
 <div class="product-image">
 50% offer

 <button class="card-btn">add to whislist</button>
 </div>
 <div class="product-info">
 <h2 class="product-brand">brand</h2>
 <p class="product-short-des">a short line about the cloth.</p>
 $20<span class="actual-
price">$40
```



</div>

</div>

<div class="product-card">

<div class="product-image">

<span class="discount-tag">50% offer</span>



<button class="card-btn">add to whislist</button>

</div>

<div class="product-info">

<h2 class="product-brand">brand</h2>

<p class="product-short-des">a short line about the cloth.</p>

<span class="price">\$20</span><span class="actual-  
price">\$40</span>

</div>

</div>

<div class="product-card">

<div class="product-image">

<span class="discount-tag">50% offer</span>

```

<button class="card-btn">add to whislist</button>
</div>
<div class="product-info">
 <h2 class="product-brand">brand</h2>
 <p class="product-short-des">a short line about the cloth.</p>
 $20<span class="actual-
price">$40
</div>
</div>
```

```
<div class="product-card">
 <div class="product-image">
 50% offer

 <button class="card-btn">add to whislist</button>
 </div>
 <div class="product-info">
 <h2 class="product-brand">brand</h2>
 <p class="product-short-des">a short line about the cloth.</p>
 $20<span class="actual-
price">$40
```

</div>

</div>

<div class="product-card">

<div class="product-image">

<span class="discount-tag">50% offer</span>



<button class="card-btn">add to whislist</button>

</div>

<div class="product-info">

<h2 class="product-brand">brand</h2>

<p class="product-short-des">a short line about the cloth.</p>

<span class="price">\$20</span><span class="actual-  
price">\$40</span>

</div>

</div>

</div>

</section>

```
<section class="product">
 <h2 class="product-category">shoes</h2>
 <button class="pre-btn"></button>
<button class="nxt-btn"></button>
 <div class="product-container">
 <div class="product-card">
 <div class="product-image">
 50% offer

 <button class="card-btn">add to whislist</button>
 </div>
 <div class="product-info">
 <h2 class="product-brand">brand</h2>
 <p class="product-short-des">a short line about the cloth.</p>
 $20<span class="actual-
price">$40
 </div>
 </div>
 </div>
 <div class="product-card">
 <div class="product-image">
 50% offer
```

```

 <button class="card-btn">add to whislist</button>
 </div>
 <div class="product-info">
 <h2 class="product-brand">brand</h2>
 <p class="product-short-des">a short line about the cloth.</p>
 $20<span class="actual-
price">$40
 </div>
</div>
<div class="product-card">
 <div class="product-image">
 50% offer

 <button class="card-btn">add to whislist</button>
 </div>
 <div class="product-info">
 <h2 class="product-brand">brand</h2>
```

```
<p class="product-short-des">a short line about the cloth.</p>
 $20<span class="actual-
price">$40
</div>
</div>
<div class="product-card">
 <div class="product-image">
 50% offer

 <button class="card-btn">add to whislist</button>
 </div>
 <div class="product-info">
 <h2 class="product-brand">brand</h2>
 <p class="product-short-des">a short line about the cloth.</p>
 $20<span class="actual-
price">$40
 </div>
</div>
</div>
```

</section>

<footer>

</footer>

<script src="static/js/nav.js"></script>

<script src="static/js/home.js"></script>

<script src="static/js/footer.js"></script>

</body>

</html>

## **Product.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Product</title>

```
<link rel="stylesheet" href="static/css/home.css">
```

```
<link rel="stylesheet" href="static/css/product.css">
```

```
</head>
```

```
<body>
```

```
 <nav class="navbar">
```

```
 </nav>
```

```
 <section class="product-details">
```

```
 <div class="image-slider">
```

```
 <div class="product-images">
```

```

```

```

```

```

```

```

```

```
 </div>
```

```
 </div>
```

```
 <div class="details">
```

```
 <h2 class="product-brand">calvin klein</h2>
```

```
 <p class="product-short-des">Lorem ipsum dolor sit, amet
consectetur adipisicing elit.</p>
```

```
 $99
```



<span class="product-actual-price">\$200</span>

<span class="product-discount">( 50% off )</span>

<p class="product-sub-heading">select size</p>

<input type="radio" name="size" value="s" checked hidden  
id="s-size">

<label for="s-size" class="size-radio-btn check">s</label>

<input type="radio" name="size" value="m" hidden id="m-  
size">

<label for="m-size" class="size-radio-btn">m</label>

<input type="radio" name="size" value="l" hidden id="l-size">

<label for="l-size" class="size-radio-btn">l</label>

<input type="radio" name="size" value="xl" hidden id="xl-  
size">

<label for="xl-size" class="size-radio-btn">xl</label>

<input type="radio" name="size" value="xxl" hidden id="xxl-  
size">

<label for="xxl-size" class="size-radio-btn">xxl</label>

```
<button class="btn cart-btn">add to cart</button>
<button class="btn">add to wishlist</button>
</div>

</section>
<section class="detail-des">
 <h2 class="heading">description</h2>
 <p class="des">Lorem ipsum dolor sit amet consectetur adipisicing
elit. Veniam,.....</p>
</section>
<section class="product">
 <h2 class="product-category">best selling</h2>
 <button class="pre-btn"></button>
 <button class="nxt-btn"></button>
 <div class="product-container">
 <div class="product-card">
 <div class="product-image">
 50% offer

 <button class="card-btn">add to whislist</button>
 </div>
```

```
<div class="product-info">
 <h2 class="product-brand">brand</h2>
 <p class="product-short-des">a short line about the
cloth.</p>
 $20<span class="actual-
price">$40
</div>
</div>
<div class="product-card">
 <div class="product-image">
 50% offer

 <button class="card-btn">add to whislist</button>
 </div>
 <div class="product-info">
 <h2 class="product-brand">brand</h2>
 <p class="product-short-des">a short line about the
cloth.</p>
 $20<span class="actual-
```

```
price">$40
 </div>
</div>
<div class="product-card">
 <div class="product-image">
 50% offer

 <button class="card-btn">add to whislist</button>
 </div>
 <div class="product-info">
 <h2 class="product-brand">brand</h2>
 <p class="product-short-des">a short line about the
cloth.</p>
 $20<span class="actual-
price">$40
 </div>
</div>
<div class="product-card">
 <div class="product-image">
 50% offer

 <button class="card-btn">add to whislist</button>
```

</div>

<div class="product-info">

<h2 class="product-brand">brand</h2>

<p class="product-short-des">a short line about the  
cloth.</p>

<span class="price">\$20</span><span class="actual-  
price">\$40</span>

</div>

</div>

<div class="product-card">

<div class="product-image">

<span class="discount-tag">50% offer</span>



<button class="card-btn">add to whislist</button>

</div>

<div class="product-info">

<h2 class="product-brand">brand</h2>

<p class="product-short-des">a short line about the  
cloth.</p>

```
 $20<span class="actual-
price">$40
```

```
 </div>
```

```
</div>
```

```
<div class="product-card">
```

```
 <div class="product-image">
```

```
 50% offer
```

```

```

```
 <button class="card-btn">add to whislist</button>
```

```
 </div>
```

```
 <div class="product-info">
```

```
 <h2 class="product-brand">brand</h2>
```

```
 <p class="product-short-des">a short line about the
cloth.</p>
```

```
 $20<span class="actual-
price">$40
```

```
 </div>
```

```
</div>
```

```
<div class="product-card">
```

```
 <div class="product-image">
```

```
 50% offer
```

```

```

```
<button class="card-btn">add to whislist</button>
</div>
<div class="product-info">
 <h2 class="product-brand">brand</h2>
 <p class="product-short-des">a short line about the
cloth.</p>
 $20<span class="actual-
price">$40
</div>
</div>
<div class="product-card">
 <div class="product-image">
 50% offer

 <button class="card-btn">add to whislist</button>
 </div>
 <div class="product-info">
 <h2 class="product-brand">brand</h2>
 <p class="product-short-des">a short line about the
```

cloth.</p>

<span class="price">\$20</span><span class="actual-price">\$40</span>

</div>

</div>

</div>

</section>

<section class="product">

<h2 class="product-category">shirts</h2>

<button class="pre-btn"></button>

<button class="nxt-btn"></button>

<div class="product-container">

<div class="product-card">

<div class="product-image">

<span class="discount-tag">50% offer</span>



<button class="card-btn">add to whislist</button>

</div>

<div class="product-info">

<h2 class="product-brand">brand</h2>

<p class="product-short-des">a short line about the



cloth.</p>

<span class="price">\$20</span><span class="actual-price">\$40</span>

</div>

</div>

<div class="product-card">

<div class="product-image">

<span class="discount-tag">50% offer</span>



<button class="card-btn">add to whislist</button>

</div>

<div class="product-info">

<h2 class="product-brand">brand</h2>

<p class="product-short-des">a short line about the cloth.</p>

<span class="price">\$20</span><span class="actual-price">\$40</span>

</div>

</div>

```
<div class="product-card">
 <div class="product-image">
 50% offer

 <button class="card-btn">add to whislist</button>
 </div>
 <div class="product-info">
 <h2 class="product-brand">brand</h2>
 <p class="product-short-des">a short line about the
cloth.</p>
 $20<span class="actual-
price">$40
 </div>
</div>
<div class="product-card">
 <div class="product-image">
 50% offer

 <button class="card-btn">add to whislist</button>
 </div>
 <div class="product-info">
 <h2 class="product-brand">brand</h2>
```

```
<p class="product-short-des">a short line about the
cloth.</p>
 $20<span class="actual-
price">$40
 </div>
</div>
```

```
</div>
</section>
<section class="product">
 <h2 class="product-category">shoes</h2>
 <button class="pre-btn"></button>
 <button class="nxt-btn"></button>
 <div class="product-container">
 <div class="product-card">
```

```
<div class="product-image">
 50% offer

 <button class="card-btn">add to whislist</button>
</div>

<div class="product-info">
 <h2 class="product-brand">brand</h2>
 <p class="product-short-des">a short line about the
cloth.</p>
 $20<span class="actual-
price">$40
 </div>
</div>

<div class="product-card">
 <div class="product-image">
 50% offer

 <button class="card-btn">add to whislist</button>
 </div>
 <div class="product-info">
 <h2 class="product-brand">brand</h2>
 <p class="product-short-des">a short line about the
```

cloth.</p>

<span class="price">\$20</span><span class="actual-price">\$40</span>

</div>

</div>

<div class="product-card">

<div class="product-image">

<span class="discount-tag">50% offer</span>



<button class="card-btn">add to whislist</button>

</div>

<div class="product-info">

<h2 class="product-brand">brand</h2>

<p class="product-short-des">a short line about the cloth.</p>

<span class="price">\$20</span><span class="actual-price">\$40</span>

</div>

</div>

```
<div class="product-card">
 <div class="product-image">
 50% offer

 <button class="card-btn">add to whislist</button>
 </div>
 <div class="product-info">
 <h2 class="product-brand">brand</h2>
 <p class="product-short-des">a short line about the
cloth.</p>
 $20<span class="actual-
price">$40
 </div>
</div>

</section>

<footer></footer>
```

```
<script src="static/js/nav.js"></script>
<script src="static/js/footer.js"></script>
<script src="static/js/home.js"></script>
<script src="static/js/product.js"></script>
</body>
</html>
```

## **Search.html**

```
<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="UTF-8">
 <meta http-equiv="X-UA-Compatible" content="IE=edge">
 <meta name="viewport" content="width=device-width, initial-
scale=1.0">
 <title>Search Results for</title>
 <link rel="stylesheet" href="static/css/home.css">
 <link rel="stylesheet" href="static/css/search.css">
</head>
<body>
 <nav class="navbar"></nav>
```

```
<section class="search-results">
 <h2 class="heading">search results for
product</h2>
</section>
<div class="product-container">
 <div class="product-card">
 <div class="product-image">
 50% offer

 <button class="card-btn">add to whislist</button>
 </div>
 <div class="product-info">
 <h2 class="product-brand">brand</h2>
 <p class="product-short-des">a short line about the cloth.</p>
 $20<span class="actual-
price">$40
 </div>
 </div>
 <div class="product-card">
 <div class="product-image">
 50% offer

```



```
 <button class="card-btn">add to whislist</button>
 </div>
 <div class="product-info">
 <h2 class="product-brand">brand</h2>
 <p class="product-short-des">a short line about the
cloth.</p>
 $20<span class="actual-
price">$40
 </div>
</div>
<div class="product-card">
 <div class="product-image">
 50% offer

 <button class="card-btn">add to whislist</button>
 </div>
 <div class="product-info">
 <h2 class="product-brand">brand</h2>
 <p class="product-short-des">a short line about the
```

```

cloth.</p>
 $20<span class="actual-
price">$40
 </div>
</div>
<div class="product-card">
 <div class="product-image">
 50% offer

 <button class="card-btn">add to whislist</button>
 </div>
 <div class="product-info">
 <h2 class="product-brand">brand</h2>
 <p class="product-short-des">a short line about the
cloth.</p>
 $20<span class="actual-
price">$40
 </div>
 </div>
<div class="product-card">
 <div class="product-image">
 50% offer

<button class="card-btn">add to whislist</button>

</div>

<div class="product-info">

<h2 class="product-brand">brand</h2>

<p class="product-short-des">a short line about the
cloth.</p>

\$20<span class="actual-
price">\$40

</div>

</div>

<div class="product-card">

<div class="product-image">

50% offer

<button class="card-btn">add to whislist</button>

</div>

<div class="product-info">

<h2 class="product-brand">brand</h2>

```
<p class="product-short-des">a short line about the
cloth.</p>
    <span class="price">$20</span><span class="actual-
price">$40</span>
    </div>
</div>
<div class="product-card">
    <div class="product-image">
        <span class="discount-tag">50% offer</span>
        
        <button class="card-btn">add to whislist</button>
    </div>
    <div class="product-info">
        <h2 class="product-brand">brand</h2>
        <p class="product-short-des">a short line about the
cloth.</p>
        <span class="price">$20</span><span class="actual-
price">$40</span>
    </div>
</div>
<div class="product-card">
    <div class="product-image">
        <span class="discount-tag">50% offer</span>
```

```
        
        <button class="card-btn">add to whislist</button>
    </div>
    <div class="product-info">
        <h2 class="product-brand">brand</h2>
        <p class="product-short-des">a short line about the
cloth.</p>
        <span class="price">$20</span><span class="actual-
price">$40</span>
    </div>
</div>
```

</div>

<footer></footer>

<script src="static/js/nav.js"></script>

<script src="static/js/footer.js"></script>

</body>

</html>

404.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Page not found-User lost</title>

<link rel="stylesheet" href="static/css/home.css">

<link rel="stylesheet" href="static/css/404.css">

```
</head>

<body>
    <nav class="navbar"></nav>
    
    <p class="four-0-four-msg">look like you are lost. Head to back to our
    <a href="#">homepage</a></p>

    <script src="static/js/nav.js"></script>

</body>
</html>
```

app.py

```
@app.route('/')
def index():
    return render_template('index.html')


@app.route('/product')
def product():
    return render_template('product.html')
```

```
@app.route('/search')
```

```
def search():
```

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

```
@app.route('/404')
```

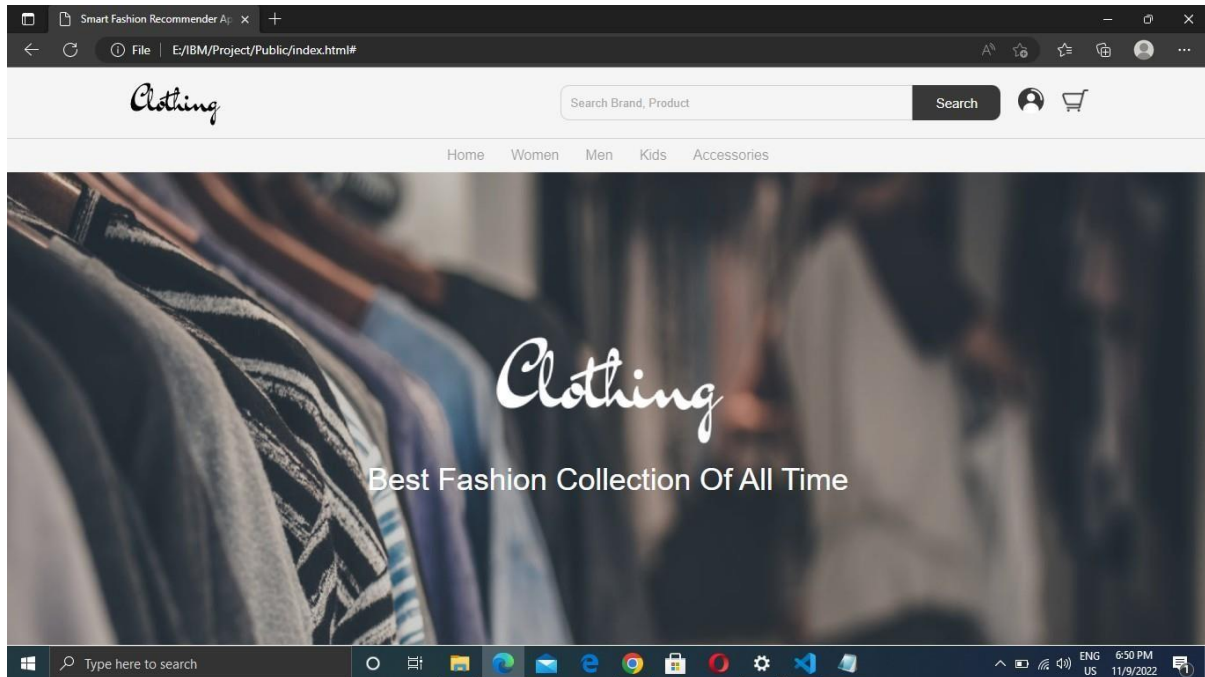
```
def error():
```

```
    return render_template('404.html')
```

```
if __name__ == '__main__':
```

```
    port = int(os.environ.get('PORT', 5000))
```

```
    app.run(debug=True, host='0.0.0.0', port=port)
```



Smart Fashion Recommender A1

File | E:/IBM/Project/Public/index.html#

Search Brand, Product

Search

Home


Women

Men

Kids

Accessories

Best Selling




50% Offer

BRAND

A Short Line About The Cloth.

\$20 \$40




50% Offer

BRAND

A Short Line About The Cloth.

\$20 \$40




50% Offer

BRAND

A Short Line About The Cloth.

\$20 \$40




50% Offer

BRAND

A Short Line About The Cloth.

\$20 \$40




50% Offer

BRA

A Short

\$20

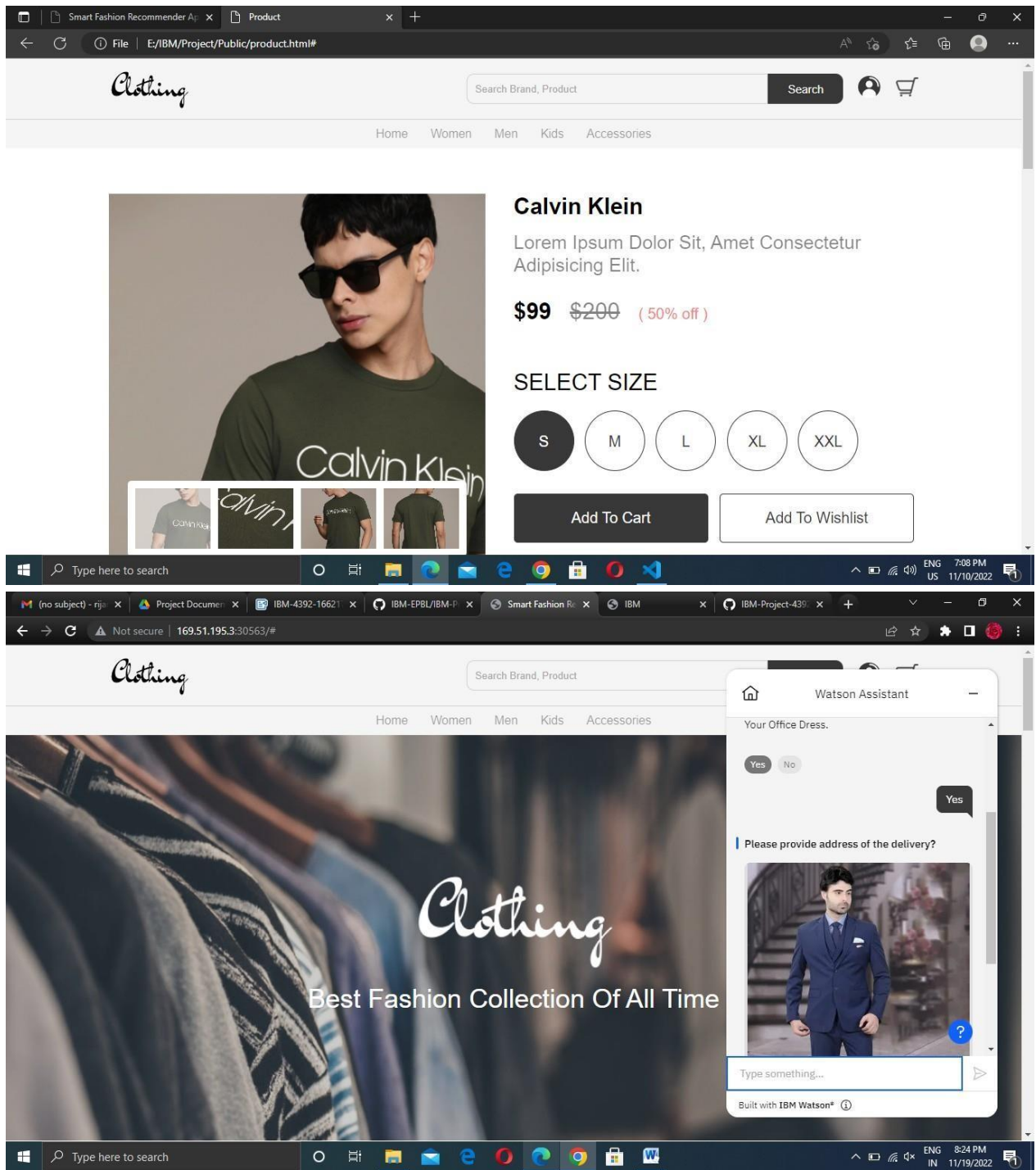
Type here to search



ENG US

7:34 PM

11/9/2022



GitHub & Project Demo Link

<https://github.com/IBM-EPBL/IBM-Project-35563-1660286062>

