



islington college

(इस्लिंग्टन कलेज)

Module Code & Module Title
CS6P05NI Final Year Project

Assessment Weightage & Type

40% FYP Final Report

Semester

2020 Spring

Student Name: Pujan Raj Rai

London Met ID: 18030046

College ID: NP01CP4A180205

Internal Supervisor: Roshan Tandukar

External Supervisor: Amulya Lohani

Assignment Due Date: 26th April 2021

Assignment Submission Date: 26th April 2021

Word Count (Where Required):

I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a mark of zero will be awarded.

Acknowledgement

I would like to express my special thank of gratitude to my teacher Mr. Amulya Lohani and Mr. Roshan Tandukar for their excellent supervision, monitoring, valuable feedbacks and suggestions on my final year project. This project helped me to grow my depth knowledge on how development process is done.

I would also would like to thank college for giving me this wonderful opportunity to do Final year project on multi-vendor ecommerce site. I would also like to thank my friends, family and colleague for encouraging to do this project.

Abstract

"No bargain, No time waste, Authentic Product with reasonable price. Now available at Samancha".

Every day we buy product to full fill our daily needs. But bargain for every item to get it in reasonable price is always troublesome and headache to everyone. So, we bring you the solution by building an ecommerce site where people can browse all the product from different shop just from your bed, living room, from anywhere around the world.

You don't have to worry about anything just go to the search for the product and add product to the cart and check out when u is doing shopping. The future is here old man. People of Nepal really love shopping. So, we bring new way of shopping in the market. Seat home, save time and do more shopping. We not only provide u to buy product online but we also let gives you platform to sell your product online.

Table of Contents

<u>CHAPTER 1: INTRODUCTION.....</u>	<u>1</u>
<u>1.1 PROJECT DESCRIPTION</u>	<u>1</u>
<u>1.2 CURRENT SCENARIO.....</u>	<u>2</u>
<u>1.3 PROBLEM DOMAIN AND PROJECT AS A SOLUTION.....</u>	<u>3</u>
<u>1.4 AIM AND OBJECTIVES.....</u>	<u>4</u>
<u>1.5 STRUCTURE OF THE REPORT.....</u>	<u>5</u>
<u>CHAPTER 2: BACKGROUND</u>	<u>6</u>
<u>2.1 ABOUT THE END USERS</u>	<u>6</u>
<u>2.2 UNDERSTANDING THE SOLUTION.....</u>	<u>7</u>
<u>2.3 SIMILAR PROJECTS</u>	<u>8</u>
2.3.1 AMAZON	8
2.3.2 ALIBABA	9
2.3.3 DARAZ	10
2.3.4 SASTO DEAL.....	11
<u>2.4 COMPARISONS.....</u>	<u>12</u>
<u>CHAPTER 3: DEVELOPMENT</u>	<u>13</u>
<u>3.1 CONSIDERED METHODOLOGIES</u>	<u>13</u>
• WATERFALL MODEL	13
• PROTOTYPE METHODOLOGY	15
• AGILE (SCRUM) METHODOLOGY.....	16
<u>3.2 SELECTED METHODOLOGY</u>	<u>18</u>
• RUP	18
<u>3.3 PHASES OF METHODOLGY.....</u>	<u>19</u>
<u>3.4 SURVEY RESULTS.....</u>	<u>20</u>
<u>3.4.1 PRE-SURVEY RESULTS.....</u>	<u>20</u>

3.5 REQUIREMENT ANALYSIS	21
3.5.1 INTRODUCTION	21
3.6 DESIGN	22
3.6.1 USE CASE DIAGRAM OF THE SYSTEM.....	22
3.6.2 ERD	23
3.6.3 COLLABORATION DIAGRAM	24
3.6.3.1 USER REGISTRATION.....	24
3.6.3.2 USER LOGIN.....	24
3.6.3.3 ADD PRODUCT	25
3.6.3.4 VIEW PRODUCT	25
3.6.3.5 EDIT PRODUCT	26
3.6.3.6 DELETE PRODUCT.....	26
3.6.3.7 MAKE ORDER	27
3.6.3.8 VIEW ORDER.....	27
3.6.4 SEQUENCE FLOW DIAGRAM.....	28
3.6.4.1 USER REGISTRATION.....	28
3.6.4.2 USER LOGIN.....	29
3.6.4.3 ADD PRODUCT	30
3.7 IMPLEMENTATION	33
3.7.1 FRONTEND	33
3.7.1.1 CUSTOMER FRONTEND	33
3.7.1.2 SELLER FRONTEND.....	36
3.7.2 BACKEND.....	39
CHAPTER 4: TESTING AND ANALYSIS	51
4.1 TEST PLAN	51
4.1.1 UNIT TESTING, TEST PLAN.....	51
4.1.2 SYSTEM TESTING, TEST PLAN	53
4.2 UNIT TESTING	54
4.2.1 CHECKING IF USER CAN LOGIN WITH EMPTY EMAIL OR PASSWORD	54
4.2.2 CHECK IF USER CAN LOGIN WITH INVALID EMAIL FORMAT.....	55
4.2.3 CHECK IF USER CAN LOGIN WITH INVALID CREDENTIAL.....	56
4.2.4 CHECK IF USER CAN REGISTER WITH EMPTY FIELD	57
4.2.5 CHECK IF USER CAN REGISTER WITH PASSWORD LESS THAT 7 DIGITS	58
4.2.6 CHECK IF USER CAN REGISTER WITH EMAIL THAT ALREADY EXIST IN DATABASE	59

4.2.7	CHECK IF USER CAN PUT MOBILE NUMBER LENGTH MORE THAN 10 DIGIT	60
4.2.8	CHECK IF SELLER CAN UPLOAD .TXT FILE INSTEAD OF IMAGE IN CITIZENSHIP CARD	61
4.2.9	CHECK IF SELLER CREATE PRODUCT BY PUTTING CHARACTER IN PRICE	62
4.2.10	CHECK IF SELLER CAN PUT PRODUCT PRICE NEGATIVE.....	63
4.2.11	CHECK IF ONE SELLER CAN ACCESS THE SHOP INFORMATION OF ANOTHER SELLER THROUGH URL	64
4.2.12	CHECK IF CUSTOMER CAN ACCESS TO SELLER DASHBOARD THROUGH URL.....	65
4.2.13	CHECK IF USER CAN ADD ITEMS TO CART WITHOUT LOGGED IN	66
4.2.14	CHECK IF USER CAN REGISTER AS SELLER	67
4.2.15	CHECK IF USER PASSWORD IS ENCRYPTED AFTER SUCCESSFUL REGISTRATION OF USER.....	69
4.2.16	CHECK IF LOGGED IN SELLER CAN ADD SHOP	70
4.2.17	CHECK IF LOGGED IN SELLER CAN ADD PRODUCT TO SHOP.....	72
4.2.18	CHECK IF LOGGED IN SELLER CAN VIEW ALL THE PRODUCT OF THEIR SHOP.....	73
4.2.19	CHECK IF THE ADDED PRODUCT IS DISPLAYED IN THEIR PARTICULAR SHOP LANDING PAGE	74
4.2.20	CHECK IF LOGGED IN USER CAN REMOVE PRODUCT FROM THE SHOP	75
4.2.21	CHECK IF LOGGED IN USER CAN ADD PRODUCT TO CART	76
4.2.22	CHECK IF LOGGED IN USER CAN REMOVE THE PRODUCT FROM THE CART	77
4.2.23	CHECK IF LOGGED IN USER CAN CHECKOUT THEMSELVES AFTER ADDING ITEMS TO CART.	79
4.2.24	CHECK IF USER CAN SEARCH PRODUCT BY THEIR NAME	81
4.2.25	CHECK IF LOGGED IN CUSTOMER CAN SEND MESSAGE TO SELLER	82
4.2.26	CHECK IF LOGGED IN CUSTOMER CAN SEE MESSAGE SEND BY SELLER	83
4.3	<u>SYSTEM TESTING.....</u>	84
4.3.1	CHECK IF USER CAN CREATE SELLER PROFILE	84
4.3.2	CHECK IF USER CAN BUY PLACE AN ORDER	86
4.3.3	CHECK SELLER CAN PERFORM CRUD OPERATION ON SHOP PRODUCT	89
4.4	<u>CRITICAL ANALYSIS.....</u>	94
	<u>CHAPTER 5: CONCLUSION</u>	95
5.1	<u>LEGAL, SOCIAL AND ETHICAL ISSUES.....</u>	95
5.1.1	LEGAL ISSUES	95
5.1.2	SOCIAL ISSUES	97
5.1.3	ETHICAL ISSUES	98
5.2	<u>5.2 ADVANTAGES</u>	100
5.3	<u>LIMITATION</u>	101
5.4	<u>FUTURE WORK</u>	102
	<u>CHAPTER 6: REFERENCES.....</u>	103

CHAPTER 7: APPENDIX	107
7.1.1 PRE-SURVEY FORM	107
7.1.2 SAMPLE OF FILLED PRE-SURVEY FORMS	109
7.1.3 PRE-SURVEY RESULT	111
7.2 APPENDIX B: POST-SURVEY	113
7.2.1 POST-SURVEY FORM	113
7.2.2 SAMPLE OF FILLED POST-SURVEY FORMS	117
7.2.3 POST-SURVEY RESULT	120
7.3 APPENDIX C: SAMPLE CODES.....	123
7.3.1 SAMPLE CODE OF THE UI.....	123
7.3.2 SAMPLE CODE FOR THE AUTOMATION SCRIPT	189
7.4 APPENDIX D: DESIGNS	200
7.4.1 GANTT CHART.....	200
7.4.2 WORK BREAKDOWN STRUCTURE	201
7.4.5 HARDWARE ARCHITECTURE	202
7.4.6 DATA FLOW DIAGRAMS	203
7.4.7 USE CASE	204
7.4.8 HIGH LEVEL USE CASE OF THE SYSTEM.....	205
7.4.8 WIREFRAME	209
7.5 APPENDIX E: SCREENSHOTS OF THE SYSTEM	225
7.6: APPENDIX F: USER FEEDBACK.....	231
7.6.1 USER FEEDBACK FORM	231
7.6.2 SAMPLE OF FILLED USER FEEDBACK FORMS.....	232
7.7 APPENDIX G: FUTURE WORK	234

7.7.1 READINGS FOR FUTURE WORK.....	234
--	------------

Table of Figures

Figure 1 similar project amazon	8
Figure 2 Similar Project alibaba	9
Figure 3 similar project daraz	10
Figure 4 similar project sasto deal	11
Figure 5 waterfall methodology.....	13
Figure 6 prototype methodology.....	15
Figure 7 scrum methodology	16
Figure 8 RUP methodology	18
Figure 9 Use case Diagram.....	22
Figure 10 ER Diagram	23
Figure 11 collaboration digram 1.....	24
Figure 12 collaboration digram 2.....	24
Figure 13 collaboration diagram 3	25
Figure 14 collaboration diagram 4	25
Figure 15 collaboration diagram 5	26
Figure 16 collaboration diagram 6	26
Figure 17 collaboration diagram 7	27
Figure 18 collaboration diagram 8	27
Figure 19 Sequence flow diagram 1	28
Figure 20Sequence flow diagram 2	29
Figure 213 Sequence flow diagram	30
Figure 22 Sequence flow diagram 4	31
Figure 23 Sequence flow diagram 5	32
Figure 24 frontend(i).....	33
Figure 25 frontend(iii).....	34
Figure 27 frontend(ii).....	34
Figure 26 frontend(iv).....	34
Figure 28 frontend(vi).....	35
Figure 29 frontend(vii).....	35
Figure 30 frontend(viii).....	36
Figure 31 frontend(ix).....	37
Figure 32 frontend(x).....	37
Figure 33 frontend(xi).....	38
Figure 34 frontend(xii).....	38
Figure 35 backend(i).....	39
Figure 36 backend(ii).....	40
Figure 37 backend(iii).....	41
Figure 38 backend(iv).....	42
Figure 39 backend(v)	43
Figure 40 backend(vi)	43
Figure 41 backend(vii)	44
Figure 42 backend(viii).....	45
Figure 43 backend(ix)	46
Figure 44 backend(ix)	47
Figure 45 backend(x)	47

Figure 46 backend(xi)	48
Figure 47 backend(xii)	49
Figure 48 backend(xiii)	50
Figure 49 unit testing 1	54
Figure 50 unit testing 2	55
Figure 51 unit testing 3	56
Figure 52 unit testing 4	57
Figure 53 unit testign 5	58
Figure 54 unit testing 6	59
Figure 55 unit testing7	60
Figure 56 unit testing 9	62
Figure 57 unit testing 10	63
Figure 58 unit testing 11	64
Figure 59 unit testing 12	65
Figure 60 unit testing 13	66
Figure 61 unit testing 14	67
Figure 62 unit testing 14(ii)	68
Figure 63 unit testing 14(iii)	68
Figure 64 unit testing 15	69
Figure 65 unit testing 15(ii)	69
Figure 66 unit testing 16	70
Figure 67 unit testing 16(ii)	71
Figure 68 unit testing 17(ii)	72
Figure 69 unit testing 17	72
Figure 70 unit testing 18	73
Figure 71 unit testing 19	74
Figure 72 unit testing 20(i)	75
Figure 73 unit testing 20(ii)	75
Figure 74 unit testing 21	76
Figure 75 unit testing 22	77
Figure 76 unit testing 22(ii)	78
Figure 77 unit testing 23	79
Figure 78 unit testing 23(ii)	80
Figure 79 unit testing 23(iii)	80
Figure 80 unit testing 24	81
Figure 81 unit testing 25	82
Figure 82 unit testing 26	83
Figure 83 system tesing 1(i).....	84
Figure 84 system tesing 1(ii).....	85
Figure 85 system tesing 1(iii)	85
Figure 86 system tesing 2(i).....	86
Figure 87 system testing 2(ii).....	87
Figure 88 system tesing 2(iii)	87
Figure 89 system tesing 2(iv).....	88
Figure 90 system testing 2(v).....	88
Figure 91 system tesing 3(i).....	90

Figure 92 system tesing 3(ii).....	90
Figure 93 system tesing 3(iii)	91
Figure 94 system tesing 3(iv).....	91
Figure 95 system tesing 3(v).....	92
Figure 96 system tesing 3(vi).....	92
Figure 97 system tesing 3(vii).....	93
Figure 98 system tesing 3(viii)	93
Figure 99 pre-servery form 1	107
Figure 100 pre-servery form 2	107
Figure 101PRE-SURVEY FORM3	108
Figure 102 PRE-SURVEY FORM 4	108
Figure 103 SAMPLE OF FILLED PRE-SURVEY FORMS 1	109
Figure 104SAMPLE OF FILLED PRE-SURVEY FORMS 2	109
Figure 105AMPLE OF FILLED PRE-SURVEY FORMS 3	110
Figure 106 AMPLE OF FILLED PRE-SURVEY FORMS 4	110
Figure 107 PRE-SURVEY RESULT 1	111
Figure 108 PRE-SURVEY RESULT 2	111
Figure 109 PRE-SURVEY RESULT 3	112
Figure 110 PRE-SURVEY RESULT 4	112
Figure 111 POST-SURVEY FORM 1.....	113
Figure 112 POST-SURVEY FORM 2.....	114
Figure 113 POST-SURVEY FORM 3.....	115
Figure 114 POST-SURVEY FORM 4.....	116
Figure 115 SAMPLE OF FILLED POST-SURVEY FORMS 1.....	117
Figure 116 SAMPLE OF FILLED POST-SURVEY FORMS 2.....	118
Figure 117 SAMPLE OF FILLED POST-SURVEY FORMS 3.....	119
Figure 118 POST-SURVEY RESULT 1	120
Figure 119 POST-SURVEY RESULT 2	120
Figure 120 POST-SURVEY RESULT 3	121
Figure 121 POST-SURVEY RESULT 4	121
Figure 122 POST-SURVEY RESULT 5	121
Figure 123 POST-SURVEY RESULT 6	122
Figure 124 POST-SURVEY RESULT 7	122
Figure 125 gannt chart	200
Figure 126 work breakdown	201
Figure 127 Hardware Architecture	202
Figure 128 DFD	203
Figure 129 Use case	204
Figure 130 Wrieframe i.....	209
Figure 131Wrieframe	211
Figure 132 Wireframe login	211
Figure 133 Wireframe register	212
Figure 134 Wireframe user details	213
Figure 135 wireframe shop register	214
Figure 136 Wireframe shop details	215
Figure 137 email verification	216

Figure 138 Wireframe change password	217
Figure 139 Wireframe password reset link	218
Figure 140 Wireframe reset.....	219
Figure 141 Wireframe dashboard 1	220
Figure 142 Wireframe dashboard2.....	221
Figure 143 dashboard 3.....	222
Figure 144 Wireframe dashboard 4	223
Figure 145 Wireframe dashboard 5	224
Figure 146 screenshot of the system 1	225
Figure 147 screenshot of the system 2.....	225
Figure 148 screenshot of the system 3.....	226
Figure 149 4	226
Figure 150 screenshot of the system 5	227
Figure 151 screenshot of the system 6.....	227
Figure 152 screenshot of the system 8.....	228
Figure 153 screenshot of the system 7	228
Figure 154 screenshot of the system 9	229
Figure 155 screenshot of the system 10.....	229
Figure 156 screenshot of the system 11	230
Figure 157 screenshot of the system 12	230
Figure 158 USER FEEDBACK FORM 1	231
Figure 159 USER FEEDBACK FORM 2	232
Figure 160 SAMPLE OF FILLED USER FEEDBACK FORMS 1	232
Figure 161 SAMPLE OF FILLED USER FEEDBACK FORMS 2	233

Table of Tables

Table 1 Understanding the solution	7
Table 2 Unit Testing plan	52
Table 3 System testing plan	53
Table 4 unit test 1.....	54
Table 5 unit test 2.....	55
Table 6 unit test 3.....	56
Table 7 unit test 4.....	57
Table 8 unit test 5.....	58
Table 9 unit test 6.....	59
Table 10 unit test 7.....	60
Table 11unit test 8.....	61
Table 12 unit test 9.....	62
Table 13 unit test 10.....	63
Table 14 unit test 11.....	64
Table 15 unit test 12.....	65
Table 16 unit test 13.....	66
Table 17 unit test 14.....	67
Table 18unit test 15.....	69
Table 19 unit test 16.....	70
Table 20 unit test 17.....	72
Table 21 unit test 18.....	73
Table 22 unit test 19.....	74
Table 23 unit test 20.....	75
Table 24 unit test 21.....	76
Table 25 unit test 22.....	77
Table 26 unit test 23.....	79
Table 27 unit test 24.....	81
Table 28 unit test 25.....	82
Table 29 unit test 26.....	83
Table 30 System testing 1	84
Table 31 system tesing 2.....	86
Table 32 system testing 3.....	89
Table 33 High level use case 1	205
Table 34 High level use case 2	205
Table 35 High level use case 3	205
Table 36 High level use case 4	205
Table 37 High level use case 5	205
Table 38 High level use case 6	206
Table 39 High level use case 7	206
Table 40 High level use case 8	206
Table 41 High level use case 9	206
Table 42 High level use case 10	207
Table 43 High level use case 11	207
Table 44 High level use case 12	207
Table 45 High level use case 13	207

Table 46 High level use case 14	207
Table 47 High level use case 15	208
Table 48 High level use case 16	208
Table 49 High level use case 17	208

CHAPTER 1: INTRODUCTION

1.1 PROJECT DESCRIPTION

In today's world, the internet has made a huge impact on human life. The introduction of the internet has changed society, driving it forward from the industrial age to the modern computer era. The internet has changed the business, education, government, healthcare, and even ways in which we interact with our loved one, becoming one of the key drivers of social evolution. The speed of the information transfer increases with new technologies.

Ecommerce, also known as electronic commerce, is the online purchase and selling of goods and services, as well as the financial and data transfers required to complete these transactions. Ecommerce is most often associated with the online sale of physical goods, but it may also apply to any form of commercial transaction made possible by the internet. Ecommerce focuses on the exchange of goods and services, while e-business covers all aspects of operating an online business.

On August 11, 1994, a man sold a CD by the band Sting to a friend via his website Net Market, an American retail portal, which marked the beginning of ecommerce. This is the first instance of a customer buying a product from a company through the Internet, or "ecommerce," as it is now known. Since then, ecommerce has advanced to make finding and purchasing items through online retailers and marketplaces easier. Independent freelancers, small companies, and large corporations have all benefited from ecommerce, which allows them to sell their products and services on a scale that would be impossible to achieve through traditional offline retail. (shopify, 2020)

So, I am also building a multi-vendor ecommerce site. The main objective of this site is to provide shop owner to host their shop as well as their product online where people from all around the country can see their shop and product and buy it.

1.2 CURRENT SCENARIO

In the today's modern world people have very busy schedule, and managing time has been very difficult. So regular shopping has been time consuming, boring and outdated.

Ecommerce is one of the trending business models in developed and developing country. Multi-national company like amazon, eBay, Alibaba are doing great on the field of ecommerce. While there are many options where people can do online shopping in other country but it is not the same case if we when it comes to Nepal. The big international company are not coming in Nepal due to our complex process to register international company and also due its close market as well as people of the country were not interested. But now due to the reach of internet in every household people are very interested in online shopping and because they don't have to bargain and compare price of product in different online site and buy from the desire and affordable ones.

While the customer who are interested on online shopping are increasing but shop owners are finding difficult to host their own ecommerce site online due its high development and maintenance cost. Even though there are multivendor site like sastodeal, daraz etc. they only provide shop owner to find to upload their product not their shop. So, I am developing multi-vendor ecommerce site where people can lunched their shop ecommerce site just with few clicks.

1.3 PROBLEM DOMAIN AND PROJECT AS A SOLUTION

1.3.1 Problem Domain

While the customer who are interested on online shopping are increasing but the shop owners are having difficulties to host their own ecommerce site online due its high development and maintenance cost. Even though there are multivendor site like Sastodeal, Gyapu etc. they only allow shop owner to upload and sell their product. If a customer bought a product and like it a lot and if they want to buy a new product from the same shop it's very difficult to order from the same shop. So, this kind of website only allows product to go online but not the shop itself. Some of the businessman also have multiple shop in different location but there is no option for a single user to have different shop in a same website.

1.3.2 Project As a solution

So, for the problem mentioned in problem domain I have developed multi-vendor ecommerce site where people can launch their shop ecommerce site just with few clicks. This Web app allows user to add multiple shop and inside shop they can add product for their respective shop. And all the shop has their own landing page. And for each shop they can track their order details. This web app also allows user to interact with customer through chat. In case of customer, customer can order anything from any shop after paying. Customer also can search both product and shop as they like.

1.4 AIM AND OBJECTIVES

The main goal of this project is to eradicate the old style of shopping and helping people, startups, to launch their shop online who are willing to go online or cannot afford their site to go online with just some click in affordable price.

The objectives of this project are listed below:

- To learn about the Django and develop user-friendly web app.
- To launch any category shop online with just a matter of click
- To create affordable place where people can launch their shop online
- To let a single seller to add multiple shop and add multiple products in each of their shop
- To help people to find the best product and shop according to the needs
- To make user friendly, responsive website.
- To allow customer and seller to interact with each other through chat.

1.5 STRUCTURE OF THE REPORT

1.3.3 BACKGROUND

A summary of the project's activities and priorities is included in the project context. This segment also explains how the challenges and needs can be solved after it has been successfully introduced. The project's end-user is also defined in this section. In this part, the project's related features will be compared to those of other projects.

1.3.4 DEVELOPMENT

The software methodology used to develop this project is described in a development section, as well as the phases of development. The results of the survey taken before and after creation are also included in this section. The functionality of both web and mobile apps are also discussed in this section.

1.3.5 TESTING AND ANALYSIS

The main goal of this chapter is to list all of the tests that were performed during the development of the application, including unit testing, system testing, and critical analysis.

1.3.6 CONCLUSION

The conclusion is the last portion of the study, and it briefly outlines the project's key topic and goal. This segment discusses the system's social, ethical, and legal problems, as well as how the system's social, ethical, and legal system functions. This segment also discusses the project's shortcomings as well as its potential possibilities.

CHAPTER 2: BACKGROUND

2.1 ABOUT THE END USERS

This end user of this multi-vendor ecommerce web application can be both customer and seller. Customer in this web application can search both product and shop according to their requirements. They can add product to the cart and checkout themselves after paying the total amount. They also can interact with seller through chat they can give review to the item they have bought. They can track their order details like the condition of the product if they are delivered or they are sent from the shop or they are still in review.

In case of seller, they can add multiple shop and for each shop they can add multiple products. They can also check the order details of each shop they also can update shop product. They can also remove the product which is not currently available. They can also interact with reply to customer if they have inbox you.

2.2 UNDERSTANDING THE SOLUTION

The project aims are to provide the platform to customer to buy products and seller lunch their shop online.

All the technology used while developing this system are listed below:

Hardware:

Computer	A modern computer with good internet connection is required to execute and developed web app.
----------	---

Software:

IDLE: PyCharm	PyCharm is powerful modern IDLE which will make our work easier and faster.
Programing Language: Python, JavaScript, jQuery	High-level programing language will be used to develop this system. Python will be used in backend while JavaScript and jQuery will be used in frontend to display and validate data.
Framework: Django.	Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. So, by using Django my project will be completed on time with all requirements.
Database: MySQL	MySQL is a relational database management system based on SQL Structured Query Language. MySQL will be used to store data of my system
Version Control: Git and GitHub	Git and GitHub will be used to control the version which will allow me to go back to different version of my system when required
Draw.io	Draw.io will be used for the construction for different diagrams.
Web browser	Any modern we browser can be used to execute my program.
HTML, BOOTSTRAP	CSS, Technology like HTML CSS and BOOTSTRAP will also be used to make our website look beautiful.

Table 1 Understanding the solution

2.3 SIMILAR PROJECTS

2.3.1 Amazon

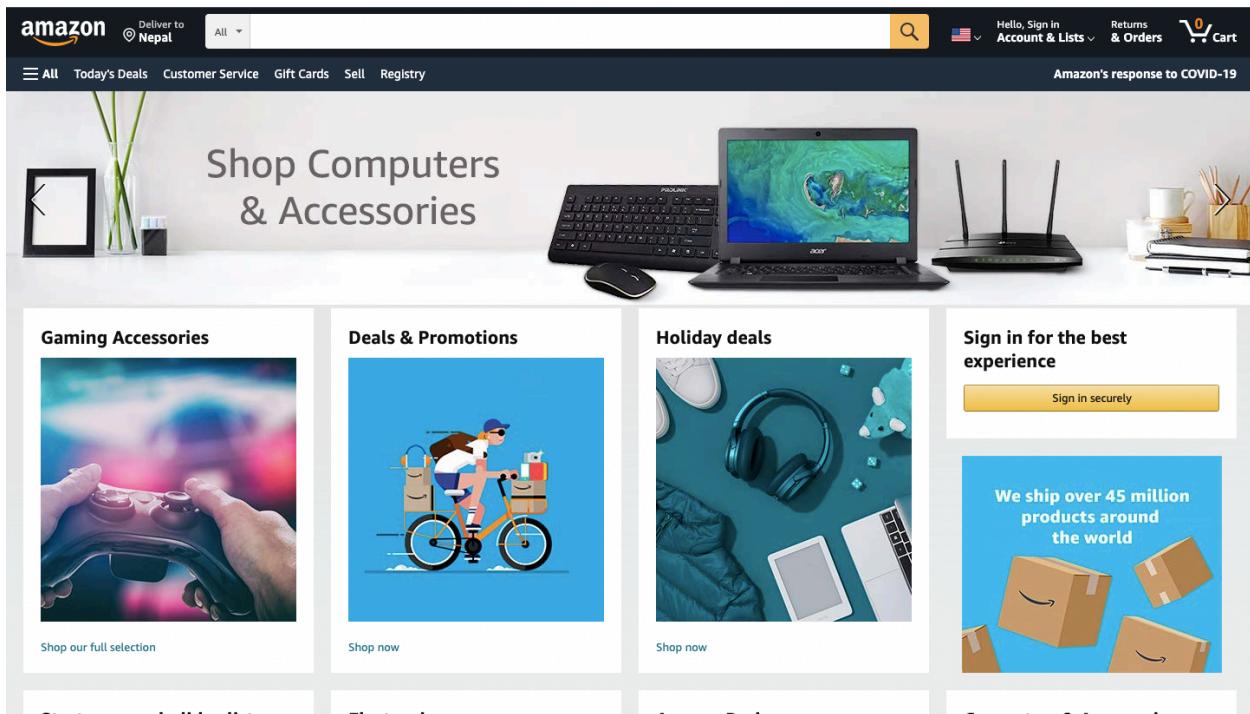


Figure 1 similar project amazon

Amazon is a multi-billionaire online retailer which allows individuals as well as business owners to sell and display their product on their site. It is one of the biggest company online retailers in the whole world. On this site you can sell your item online by paying them a certain percentage of your sale. It's a multinational retail and E-commerce company that deals with various ranges of clothing, foods, gifts, and many more products. It is a platform that gives lots of variety to the customers for the same kind of product line, and also allows comparing various products and the price of those products. Amazon accepts that advancement is that the power to bring change inside the world so it continually advances to attract clients, to strengthen their product, to animate the designers following Amazon's foundation for formation of organizations and boosting the innovative capacity of their item makers. Amazon made it simple for the buyers to benefit internet shopping in an instinctively friendly environment that results in healthier and proficient retailing as compared to regular shop. (amazon, 2020)

2.3.2 Alibaba

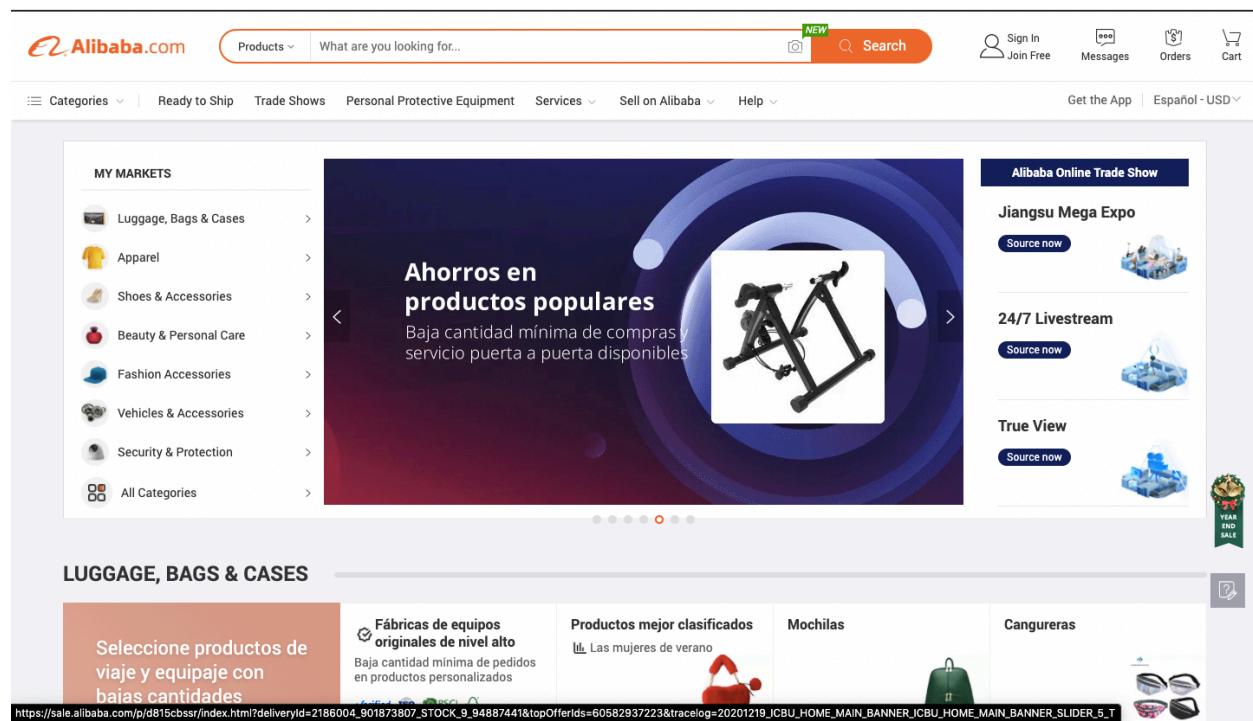


Figure 2 Similar Project alibaba

Alibaba has become the most important online and mobile commerce company within the world in only a couple of years and barely anyone expected it to be so successful. It has provided numerous business opportunities for little and medium enterprises to leverage innovative technology to compete more efficiently domestically and globally. It discusses the marketing strategies, promotion, distribution channels, and a few important lessons. Alibaba's success like its specific marketing strategies, various challenges, its strong branding image, superior customer value proposition, better shopping experience, huge sales volume, and economies of scale. (Rashad Yazdanifard, 2014)

2.3.3 Daraz

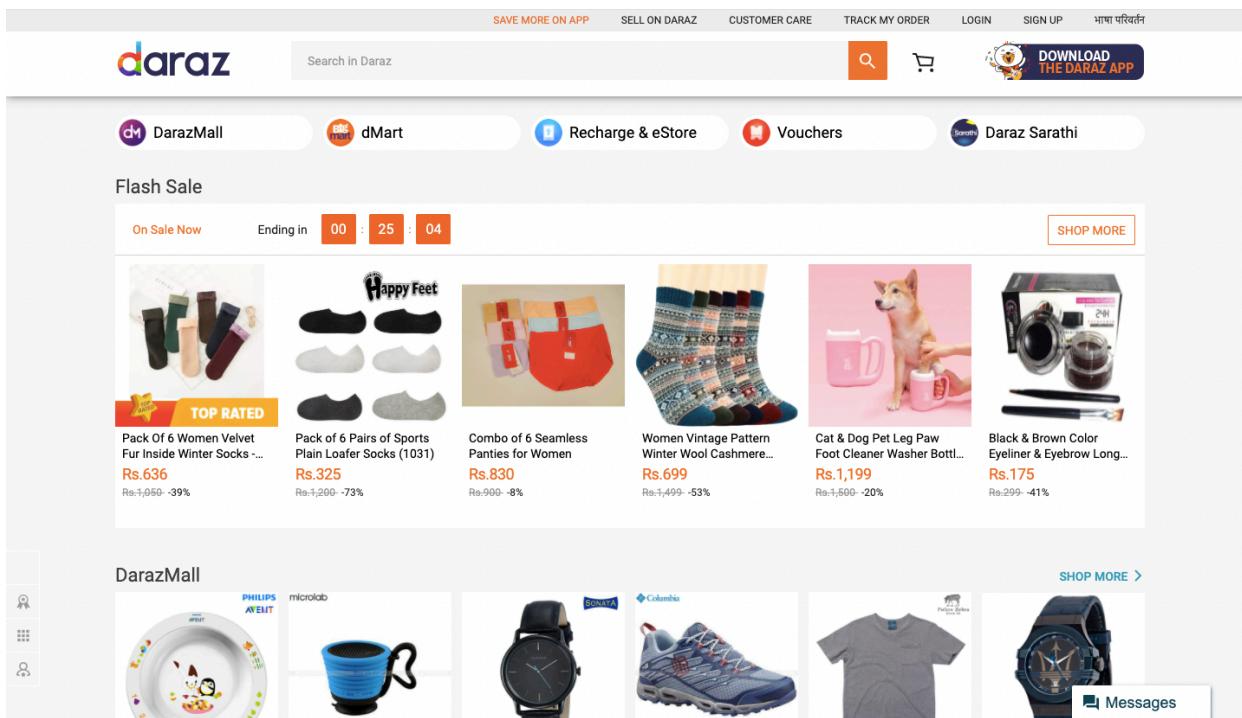


Figure 3 similar project daraz

It is the fastest growing E-Commerce platform in Nepal offering an unparalleled shopping experience at the comfort of your home. You can enjoy convenient and easy shopping without stepping out of the house. Daraz promises to require your style statement to an entire new level offering a good sort of products and makes. You can also renovate your house by buying home appliances. You get to avail delivery right at the doorstep with easy payment methods. Whether you reside in small towns of Nepal or major cities, you'll easily get your required products delivered at the doorstep. (crunchbase, 2020)

2.3.4 Sasto Deal

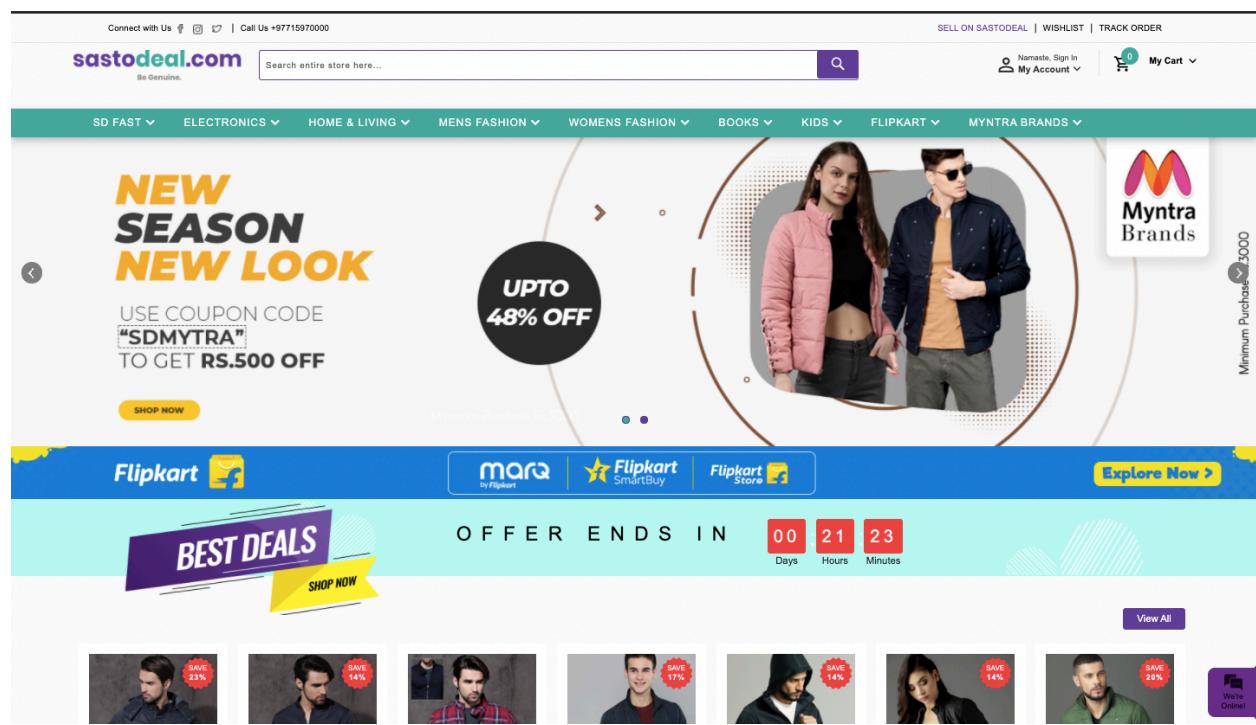


Figure 4 similar project sasto deal

Sasto Deal has been ready to provide goods during a relatively cheaper rate compared to the market value by compressing retailer's margin by tapping higher levels of wholesalers and traders. On the opposite hand, the corporate benefits retailers by featuring their products on their online buy free and taking their marketing budget off their shoulders. Though there are many similar online retailers in Nepal, Sasto Deal's founders credit their unique business culture and customer hospitality as the key to their success. It treats its customers, who are mostly working age people, as friends rather than clients and strives hard to set itself apart in every section of the sales-funnel services. (Sharma, 2020)

2.4 COMPARISONS

While developing this project I have visited many ecommerce sites from where I have a lot of knowledge about the requirements about the ecommerce site with their advantage and disadvantage.

While visiting ecommerce site I have found four i.e., Alibaba, Daraz, Sasto deal and Amazon website which are quite similar to my project. When I visited this website, I found out that there is no any web page portal for any individual for shop owner. Even though Alibaba provides web page for shop owner they are only used by big suppliers but not by the local shop owner and you have to order many at once. So, shop owner is not getting opportunity to have their own portal because of which there is decrease in sale. There are a lot of time when people like really like some product of the shop and they want to buy more from the same shop because they believe that their product is authentic and they are good in reasonable price. But in online shopping they cannot contact or visit their site. So, they have don't have any option rather than buying it from other store next time they visit the website. So, shop owner has to lose customer and customer have to customer. Similarly, people like to browser to the shop they like but in context of the multi-vendor ecommerce site they cannot browse their shop and cannot look the offer provided by the shop owner which is very painful for the business.

So, I analyzed that there is a necessary of the multi-vendor ecommerce site where shop owner have their own shop portal and their customer can browser their shop. Which will boost the shop sale and help customer to get authentic product from authentic shop.

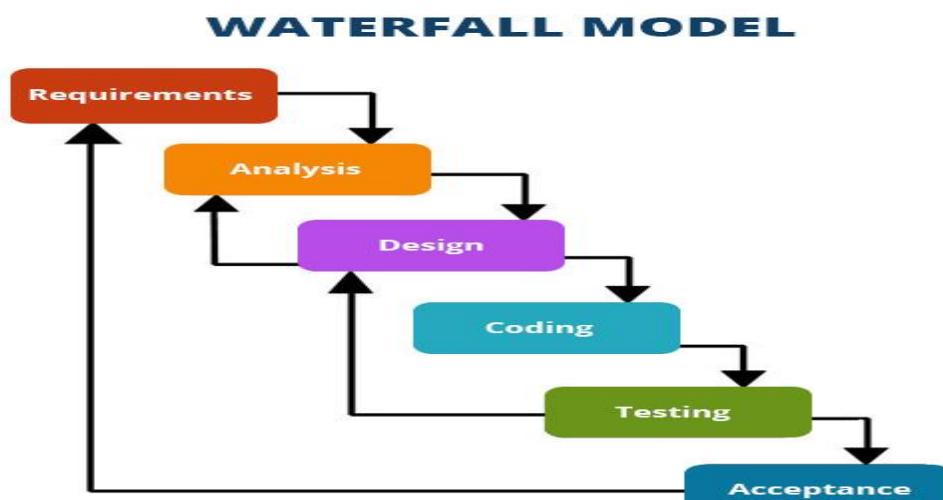
CHAPTER 3: DEVELOPMENT

3.1 CONSIDERED METHODOLOGIES

Software Methodology is a set of related activates that leads to the production of the software product. These activities that leads to the production of a software product. These activities may involve the development of software from scratch in a standard programming language like java, python, c, php etc. However, business application is not necessarily developed in this way but if they follow certain software development methodology, they are more likely to get better result in less time so it is recommended to do and most of the software development company follow certain methodology according to their needs.

- **Waterfall Model**

It is a traditional method of software development process where each process is clarified into a linear flow which basically means that any phase in this development process begins only of the earlier phase is completed. There is no such definition on going back to previous phase to handle changes in requirements in this development approach.



Copyright©2016, www.dreamztech solution. All rights reserved

Figure 5 waterfall methodology (CodeLabII Ben , 2016)

This methodology was considered due to following reason:

- Easy to understand as after each stage is finished the next starts
- Each phase is planned in detail and requires elaborated documentation
- Very effective for small projects where requirements are unchangeable and well understood
- Significant amount of time is saved as each phase are processed and completed at once in a time

Reason for not using this methodology

There is no such definition on going back to previous phase to handle changes in requirements in this development approach. My project is multi-vendor site and there is no guarantee of a fixed requirement. There are initial requirements gathered by conducting meetings, interviews and surveys. There can arise a condition where I would want to add or remove some features so if I had opted to develop project following this methodology then it would arise many problems.

- **Prototype Methodology**

This software development process initiates developers to make only the sample of the resolution to validate the customer's functional essence and make essential changes before the authentic final solution is created. This methodology was considered to be used as the methodology for following reasons:

- When a prototype is shown to the clients, they get a clear understanding of the functionality of the software and a complete sense of it.
- This methodology gives clear idea about the software's functional process.
- This method reduces the risk of failure significantly, because potential risks are identified early on and moderation steps can be taken quickly.

- **Reason for not using this methodology**

This methodology requires significantly more involvement of client and which may affect the flow of development process of my project. The requirement given by client can be changed and may result in many modifications, which ultimately disturbs the workflow of entire project.

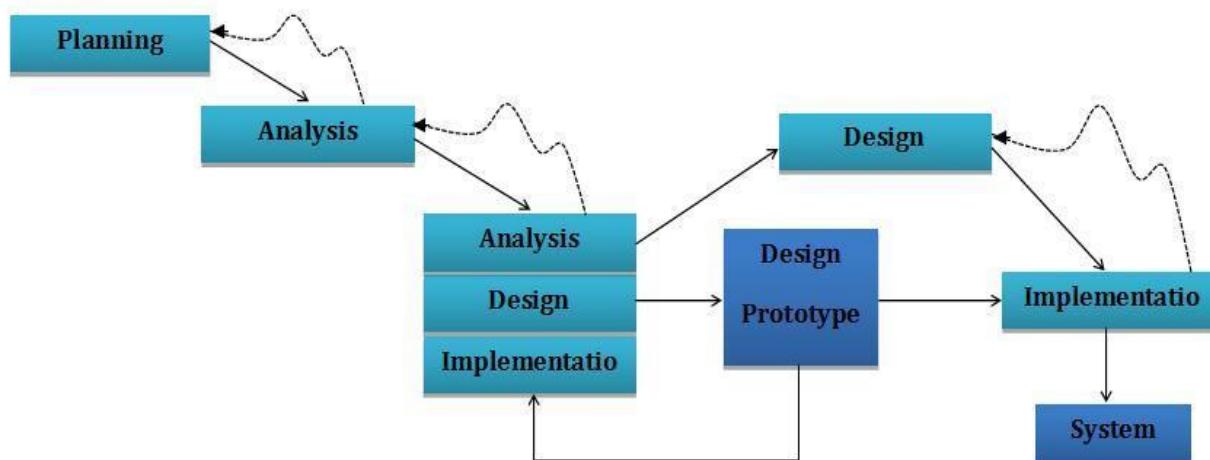


Figure 6 prototype methodology (writepass, 2012)

- **Agile (Scrum) Methodology**

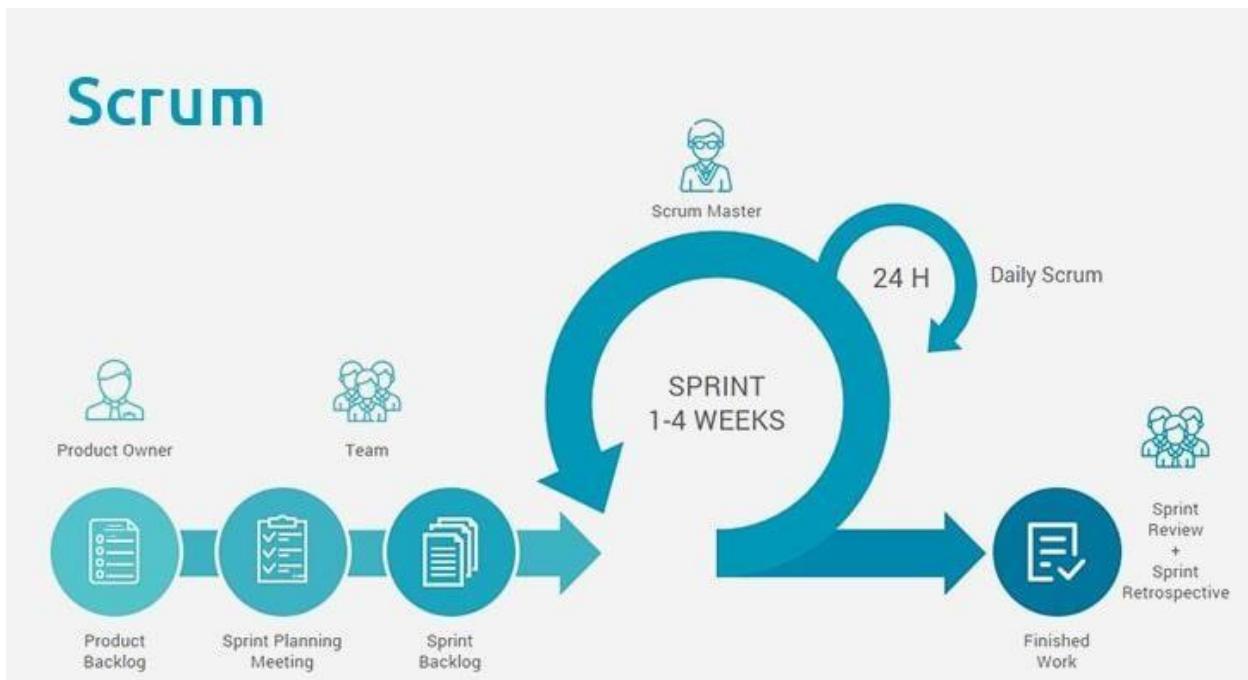


Figure 7 scrum methodology (rahmania, 2012)

Agile Methodology was the first methodology considered for the project. According to a study conducted by Version One, 94 percent of businesses are now using agile in some form. Scrum is one of the most widely used agile project management methodologies. The approach focuses on defining key features and aids in the goal-setting process at the start of each sprint.

The reason for considering this methodology are as follows:

- Adaptability and flexibility:

Defining the requirements and specification for various project modules is a challenging task. Agile technique is thought to be better suited for unpredictability.

It gives you the freedom and adaptability to identify and refine requirements and designs as the project progresses.

- Lower costs:

Scrum guarantees efficient time management. It embraces feedback and allows improvements in requirements and designs based on the feedback because it is an agile technique. This eliminates the need for redundant paperwork and controls.

Reasons for not choosing Agile (Scrum) Methodology:

- It necessitates a collaborative environment
- Agile methodology emphasizes teamwork. Scrum technique will be impossible to apply in a one-man project.
- Due to the lack of a fixed completion date, the chances of the project failing are high.

3.2 SELECTED METHODOLOGY

- **RUP**

To develop this project Rational Unified Process (RUP) is best methodology. It is a software development process from Rational, a division of IBM. In this methodology development process is divided into four different parts which are inception, elaboration, project constructions and project transition. Rup is very powerful and modern methodology which is used by big tech company which doesn't let resource to get wasted and also reduces the development cost.

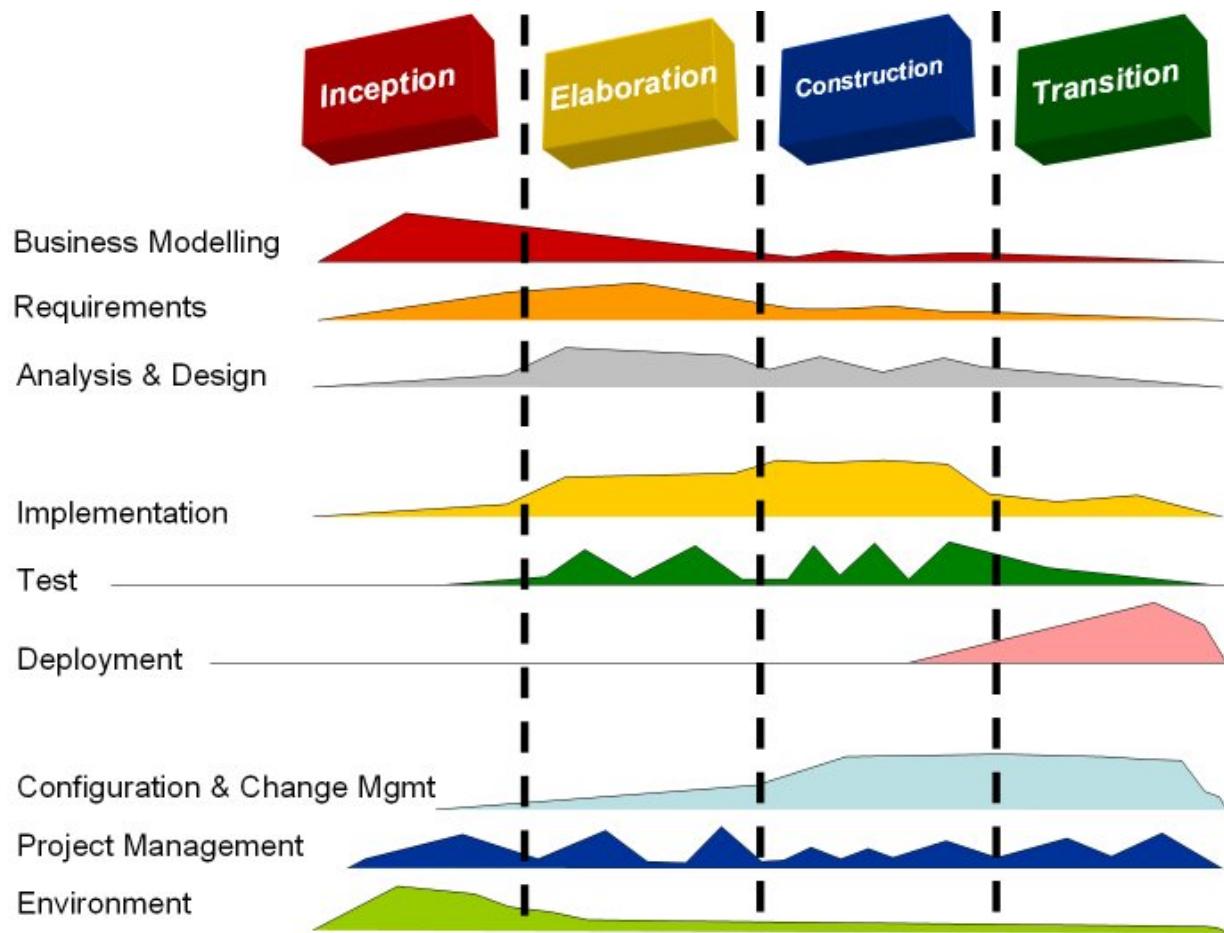


Figure 8 RUP methodology

3.3 PHASES OF METHODOLOGY

There are four phases in this methodology which are to be implemented as follows:

Inception:

- Idea of the project were finalized.
- Topic was discussed with supervisor and feedback was taken.
- Further research was done about the scope and possibilities of the project
- All the requirement and resources were collected and cost of the project was collect
- Resources required for the project are calculated and cost is estimated.

Elaboration

- The resources and cost are reevaluated and finalized here.
- Further research was done and resources, cost were reevaluated.
- The proposal is documented. Proposal documentation was submitted
- Wireframes, UML diagrams were made.

Project Construction

- The frontend and backend of the project will be developed in this phase.
- Testing will be done
- Finally documentation will be done
- Company policy with its privacy policy, terms and condition and user manual will be written Project Transition
- The project will be deployed in server.
- Feedback will be collected from users and analyzed
- Application will be update with according to the feedback will be well documented

3.4 SURVEY RESULTS

3.4.1 PRE-SURVEY RESULTS

Prior to system implementation, a pre-survey is conducted to gain an initial understanding of how the system's end-users will think about and view the project's proposed concept. It assists us in finalizing and ensuring all aspects of the system's functionality and design. Furthermore, the responses offered some significant benefits and improvements to the system that was being built.

The responses of the participants are used to decide if the proposed method is effective, accurate, or successful. Appendix contains a summary of responses to the whole questionnaire:

POST-SURVEY RESULTS

The end-user then completes a post-survey using the application. This survey aids in determining the efficacy of the system's various features and functions. In the appendix, you'll find the questionnaires and answers that were collected:

3.5 REQUIREMENT ANALYSIS

3.5.1 Introduction

This sub-primary chapter's purpose is to include an overview of the project's overall features and functionalities. It also enables you to comprehend how the system functions.

3.5.2 intended Audience and Reading Suggestion

The document is primarily aimed at developers, device testers, and end users (seller or customers) who may be interested in the application's use.

The working features, necessary components, and operating environment are all broken down in this document. As a result, it can be seen as a foundation for the creation process, assisting us in visualizing the overall system.

3.5.3 Project Scope

The major scopes of the project are given below:

- Digitally record the information of the customer who had order the
- Facility for user be able to search preferable product according to quantity and color.
- Monitor the services by incoming feedbacks given by the customers.
- Data analytics for admin panel to extract information through prevailing data.
- To allow customer to buy the product
- To make portal where seller can add, edit and delete their product
- To let seller to sell their added product

3.6 DESIGN

3.6.1 Use Case Diagram of the System

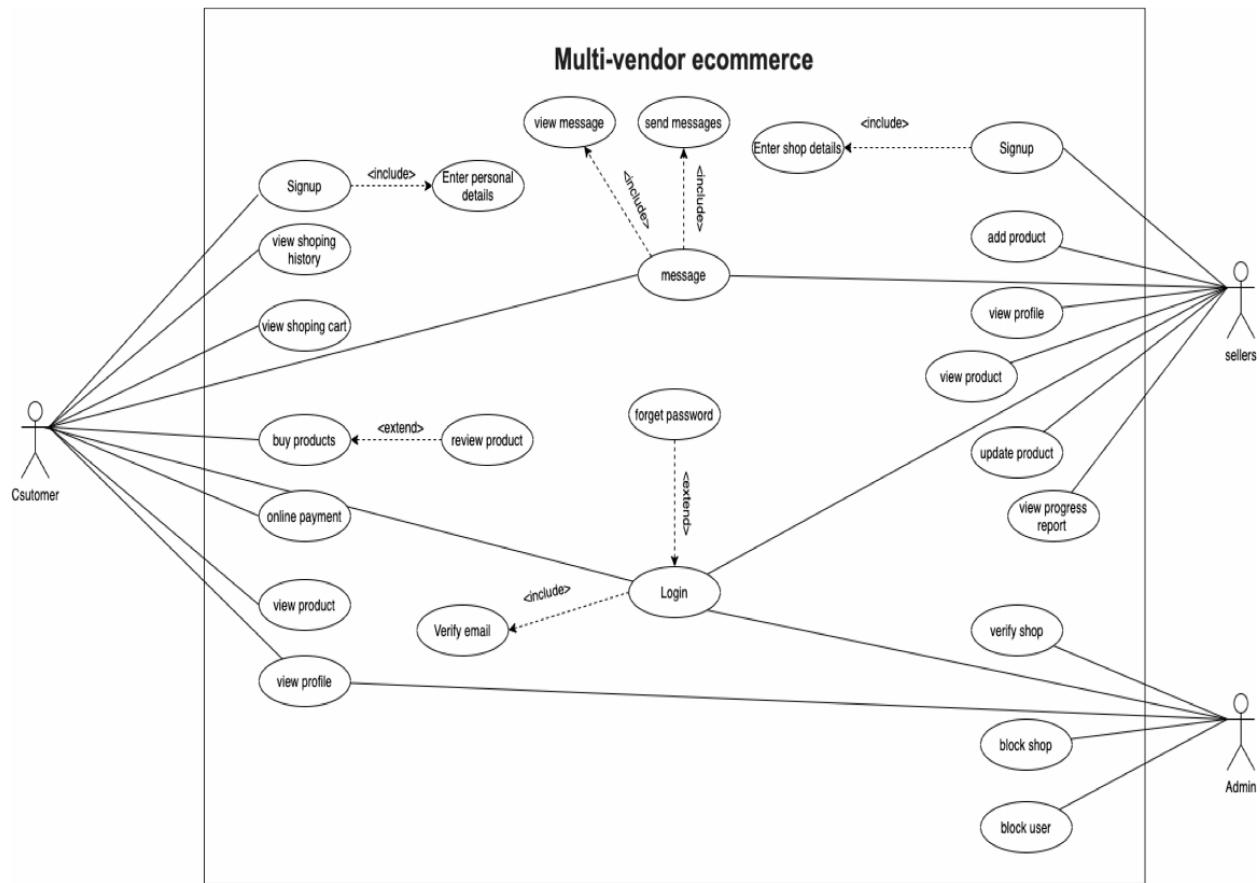


Figure 9 Use case Diagram

3.6.2 ERD

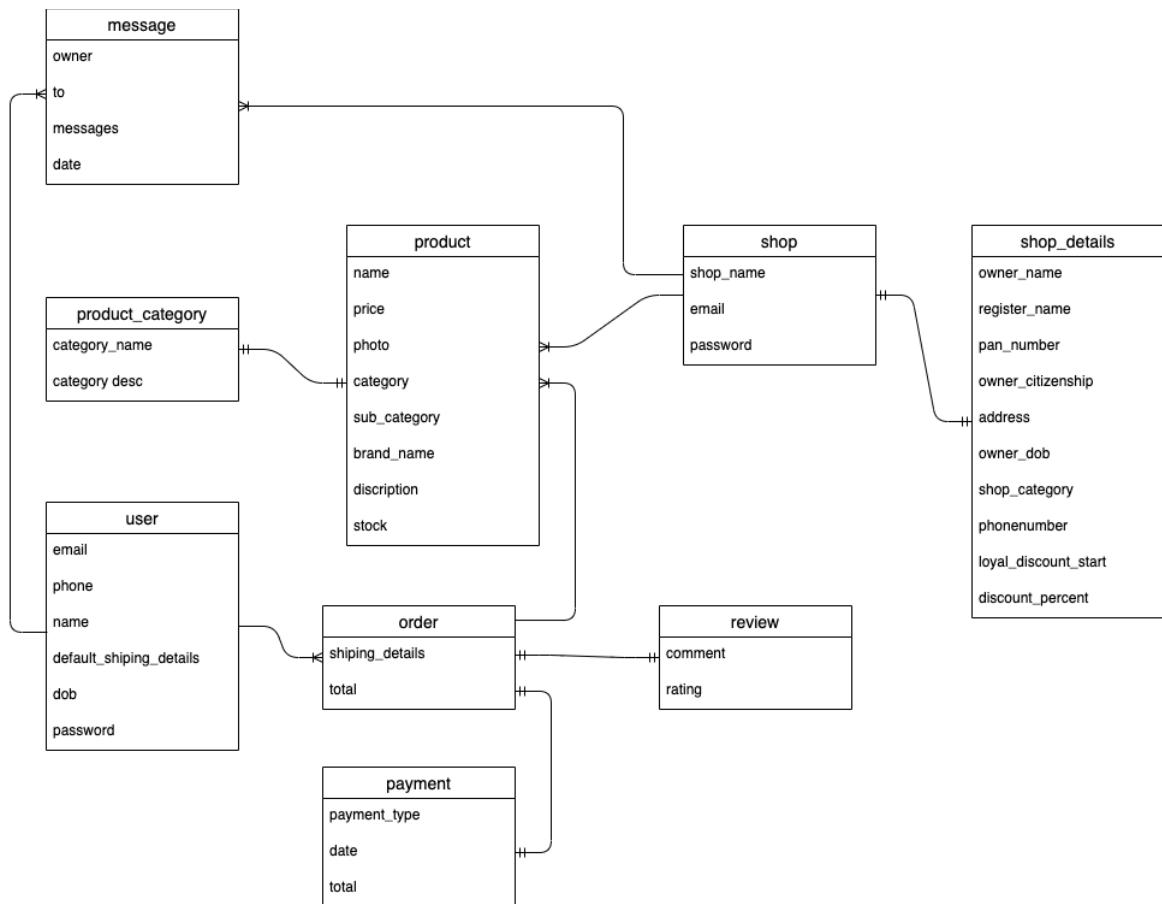


Figure 10 ER Diagram

3.6.3 Collaboration Diagram

3.6.3.1 User Registration

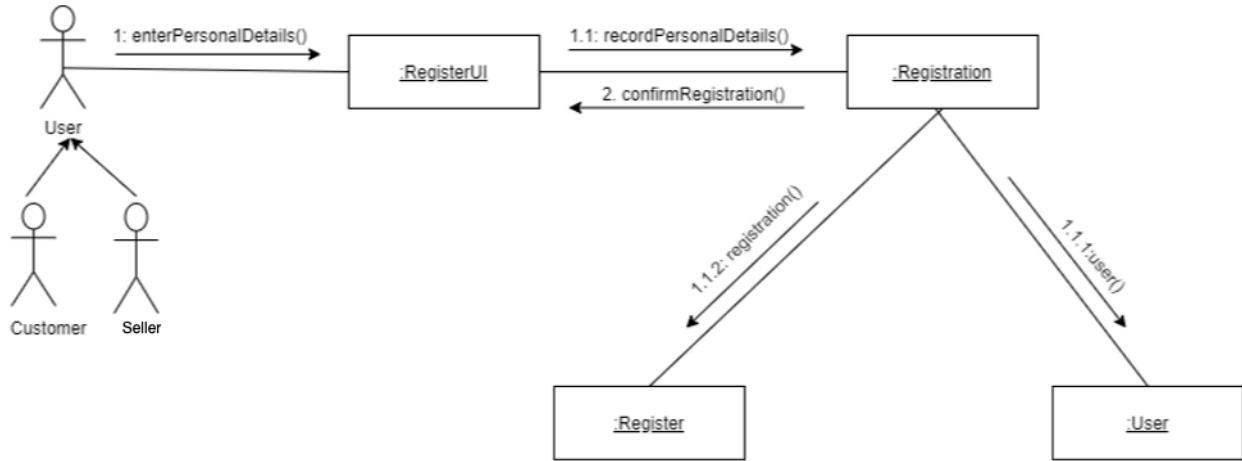


Figure 11 collaboration digram 1

3.6.3.2 User Login

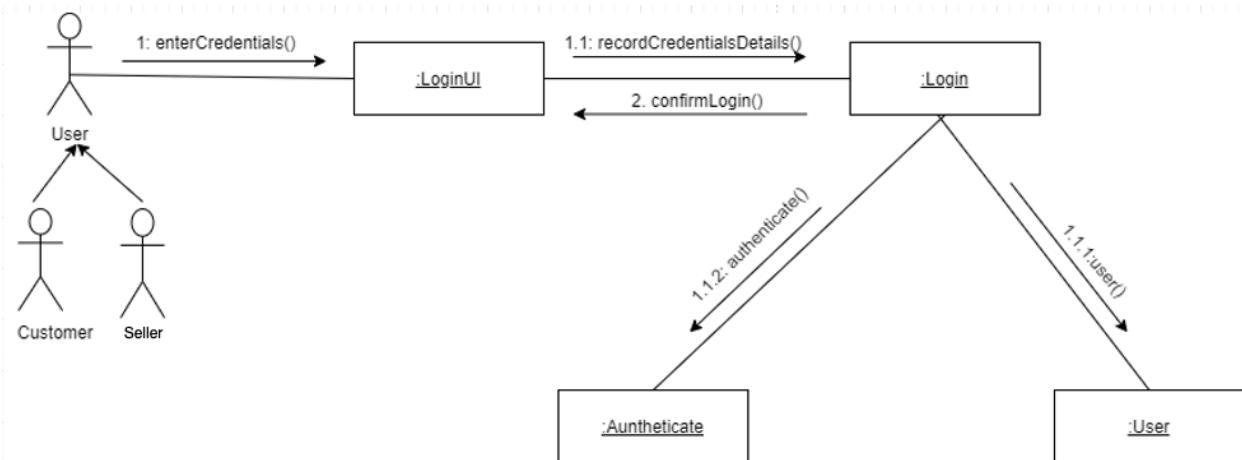


Figure 12 collaboration digram 2

3.6.3.3 Add Product

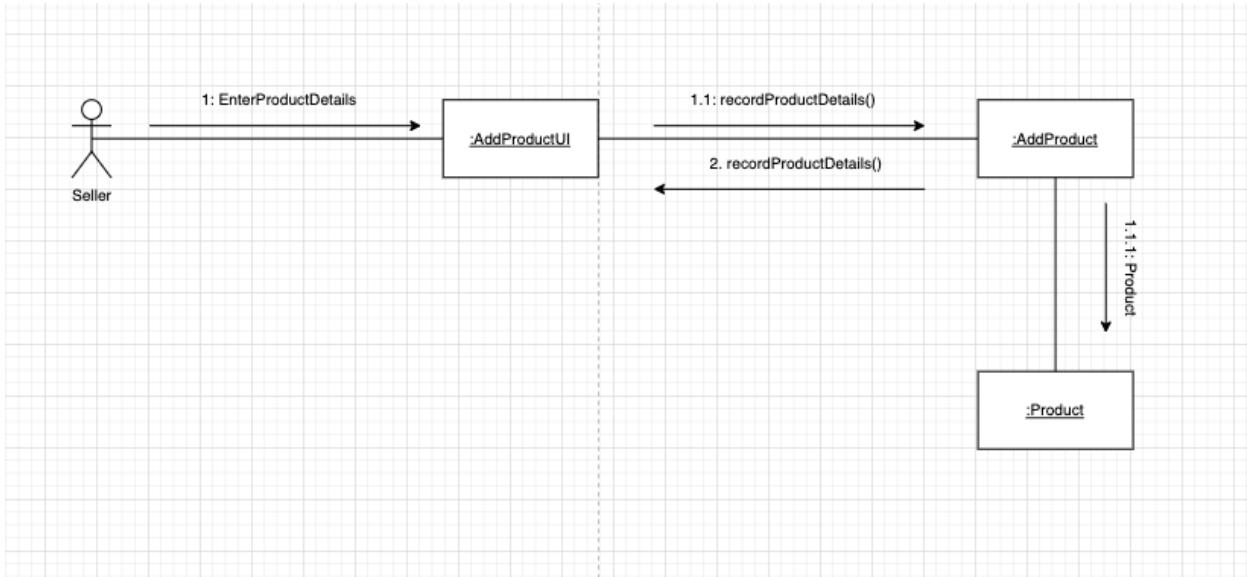


Figure 13 collaboration diagram 3

3.6.3.4 View Product

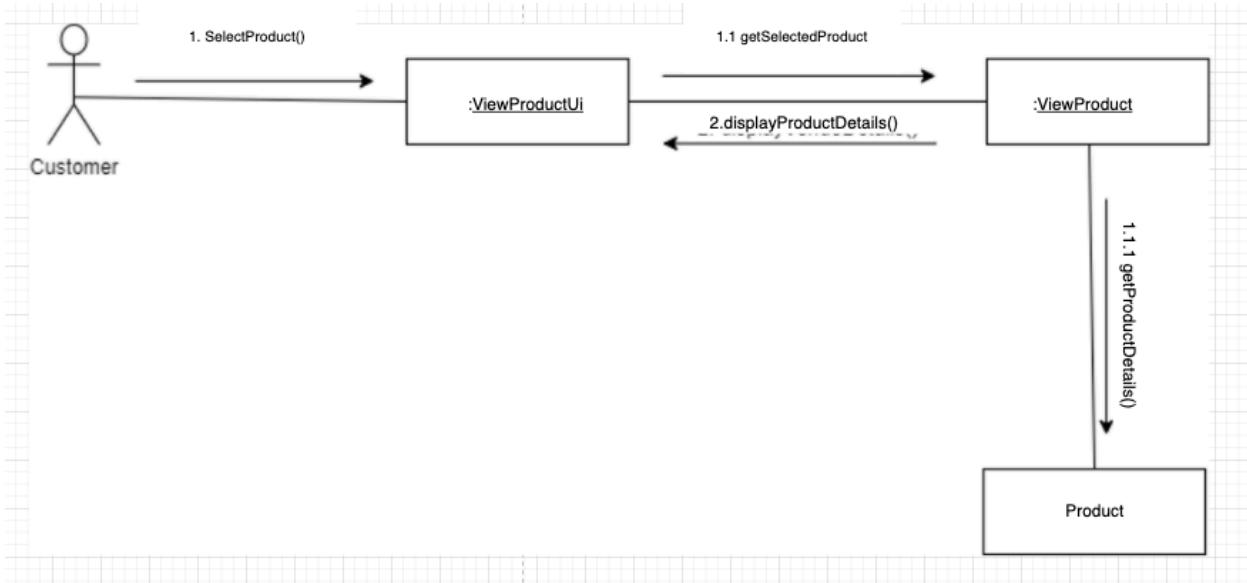


Figure 14 collaboration diagram 4

3.6.3.5 Edit Product

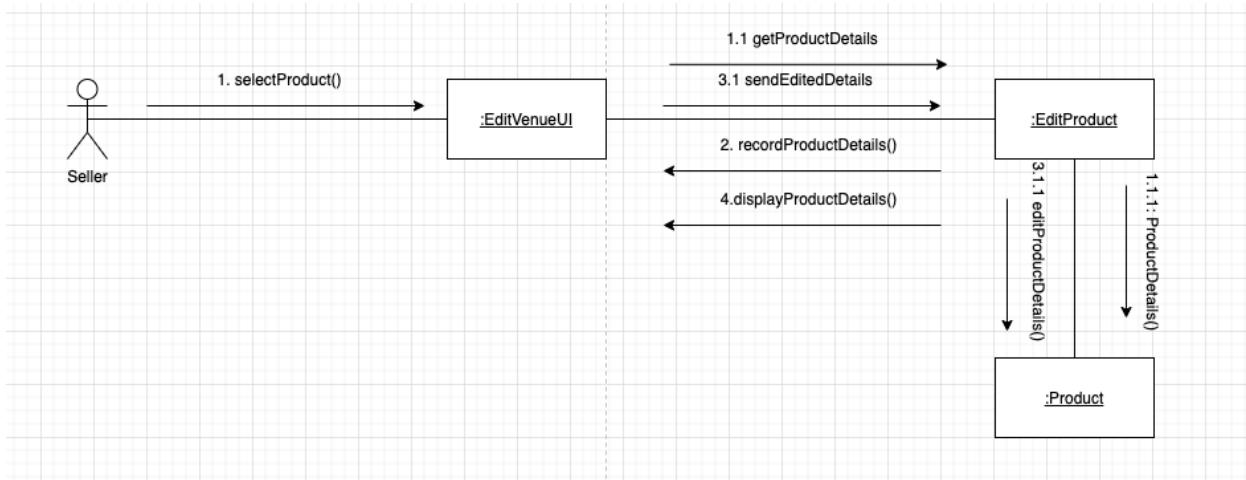


Figure 15 collaboration diagram 5

3.6.3.6 Delete Product

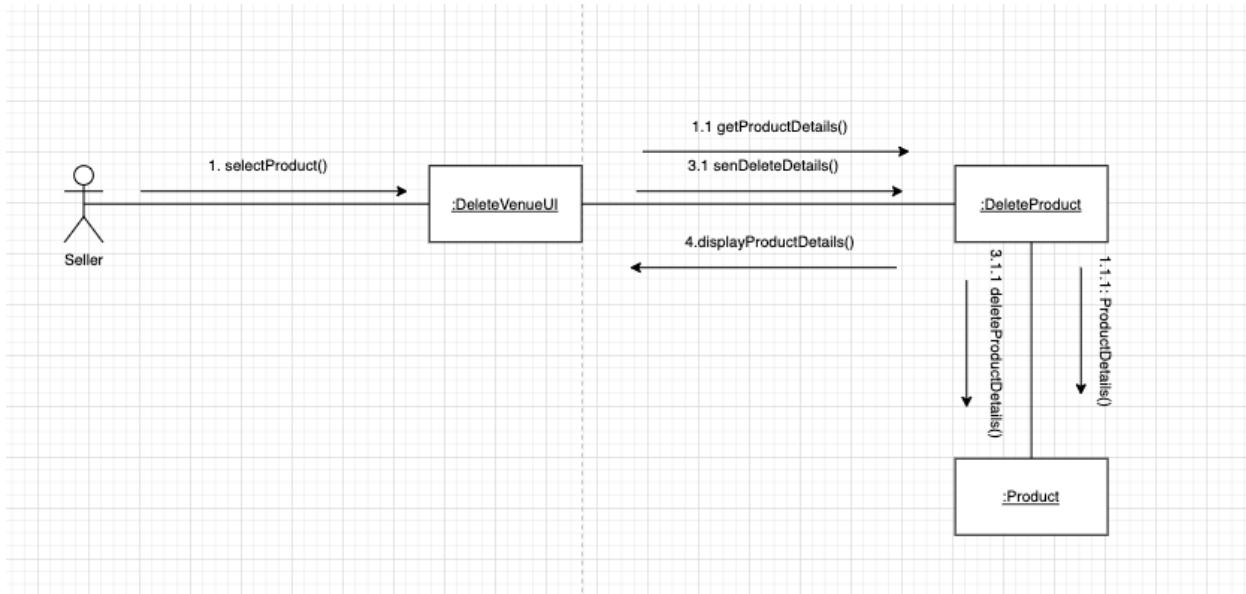


Figure 16 collaboration diagram 6

3.6.3.7 Make order

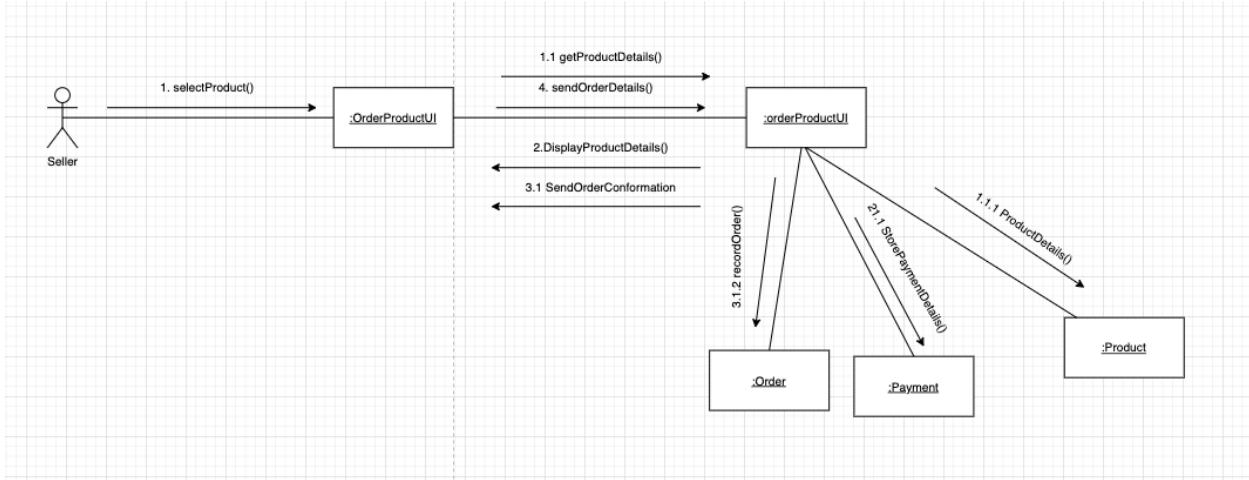


Figure 17 collaboration diagram 7

3.6.3.8 View Order

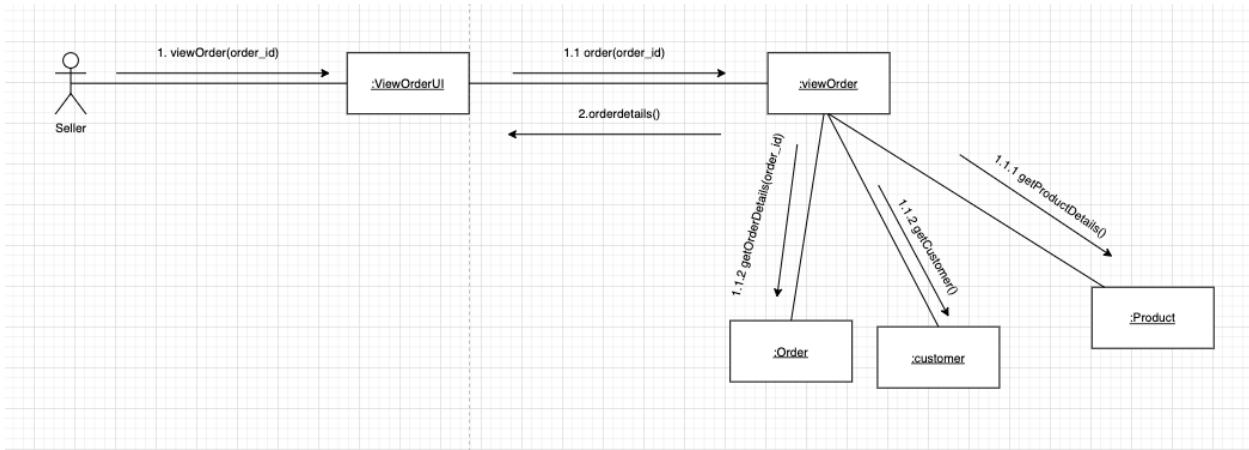


Figure 18 collaboration diagram 8

3.6.4 Sequence flow diagram

3.6.4.1 User Registration

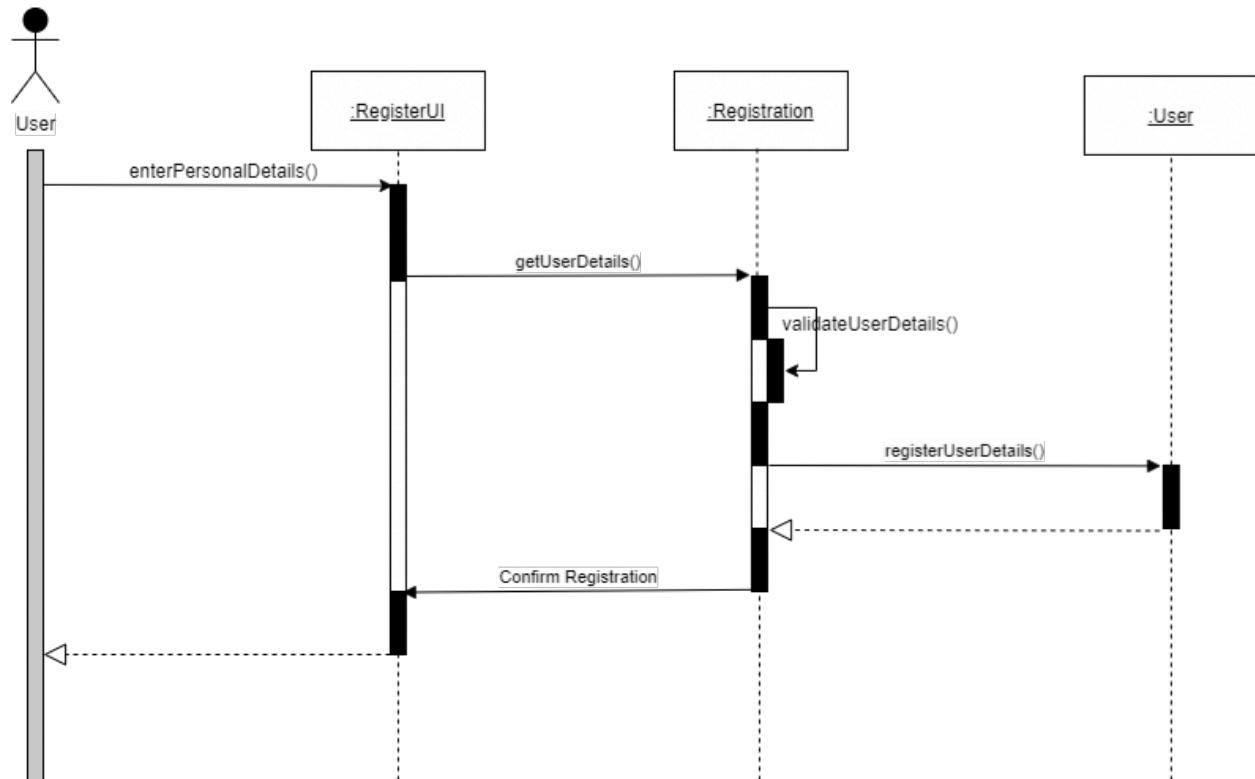


Figure 19 Sequence flow diagram 1

3.6.4.2 User Login

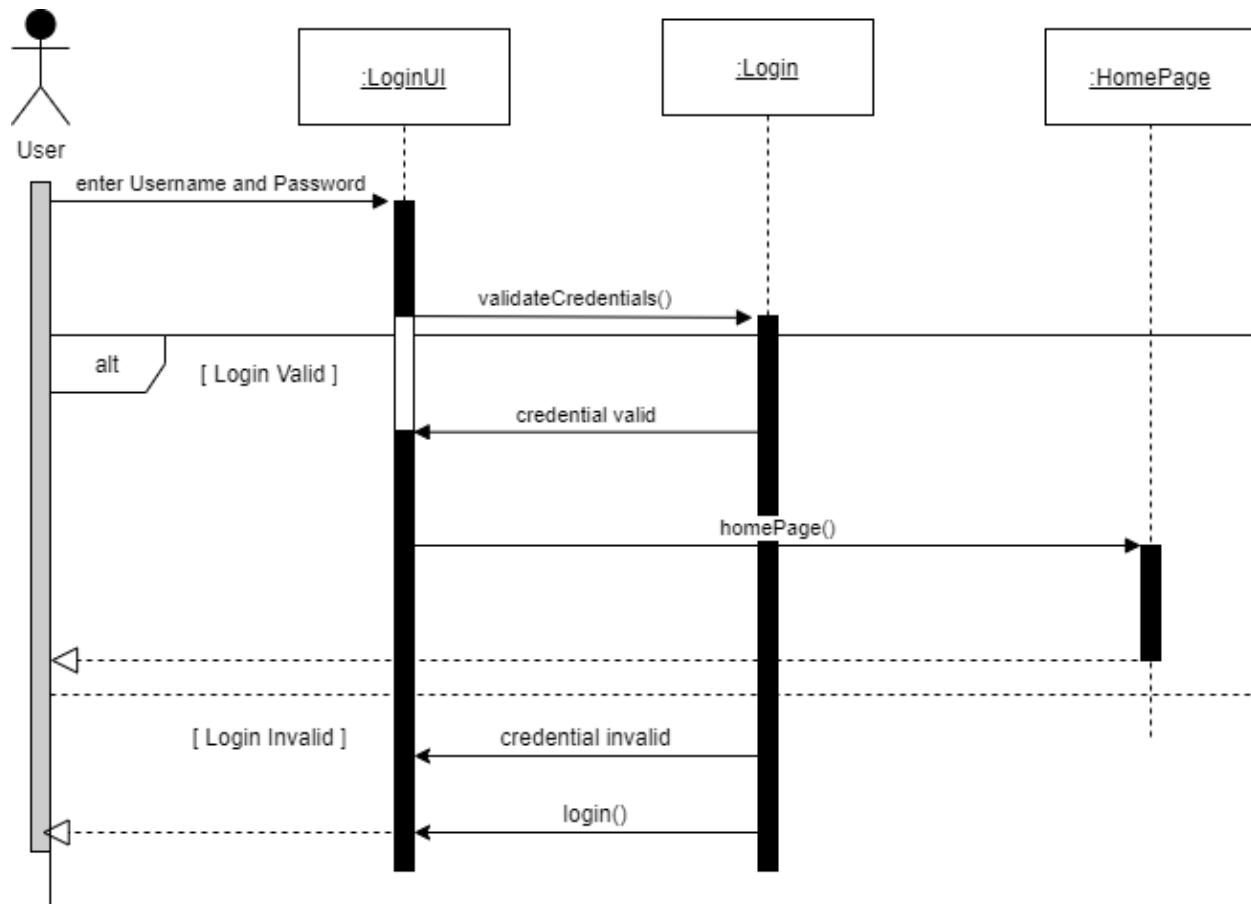


Figure 20 Sequence flow diagram 2

3.6.4.3 Add Product

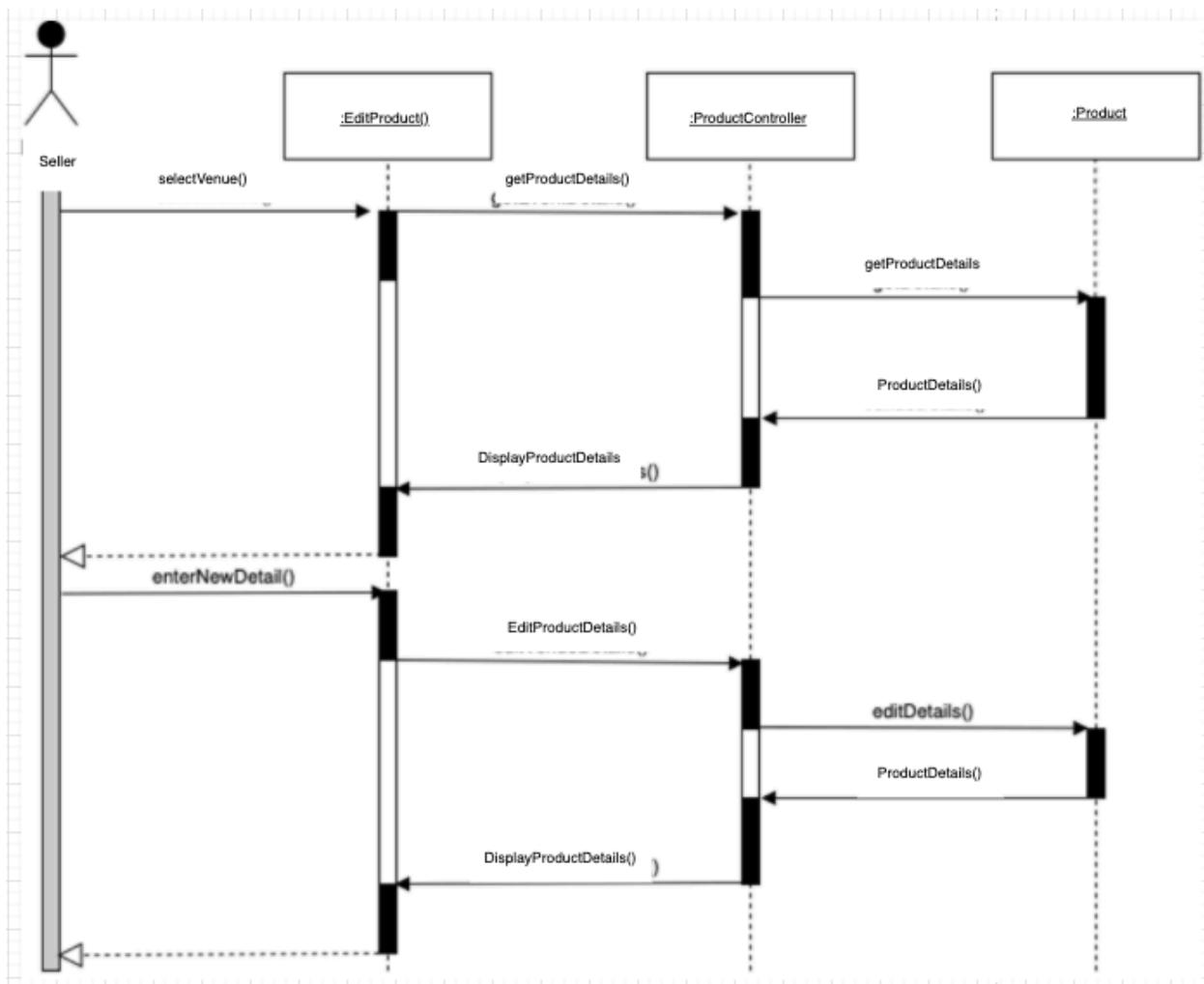


Figure 213 Sequence flow diagram

3.6.4.4 Edit Venue

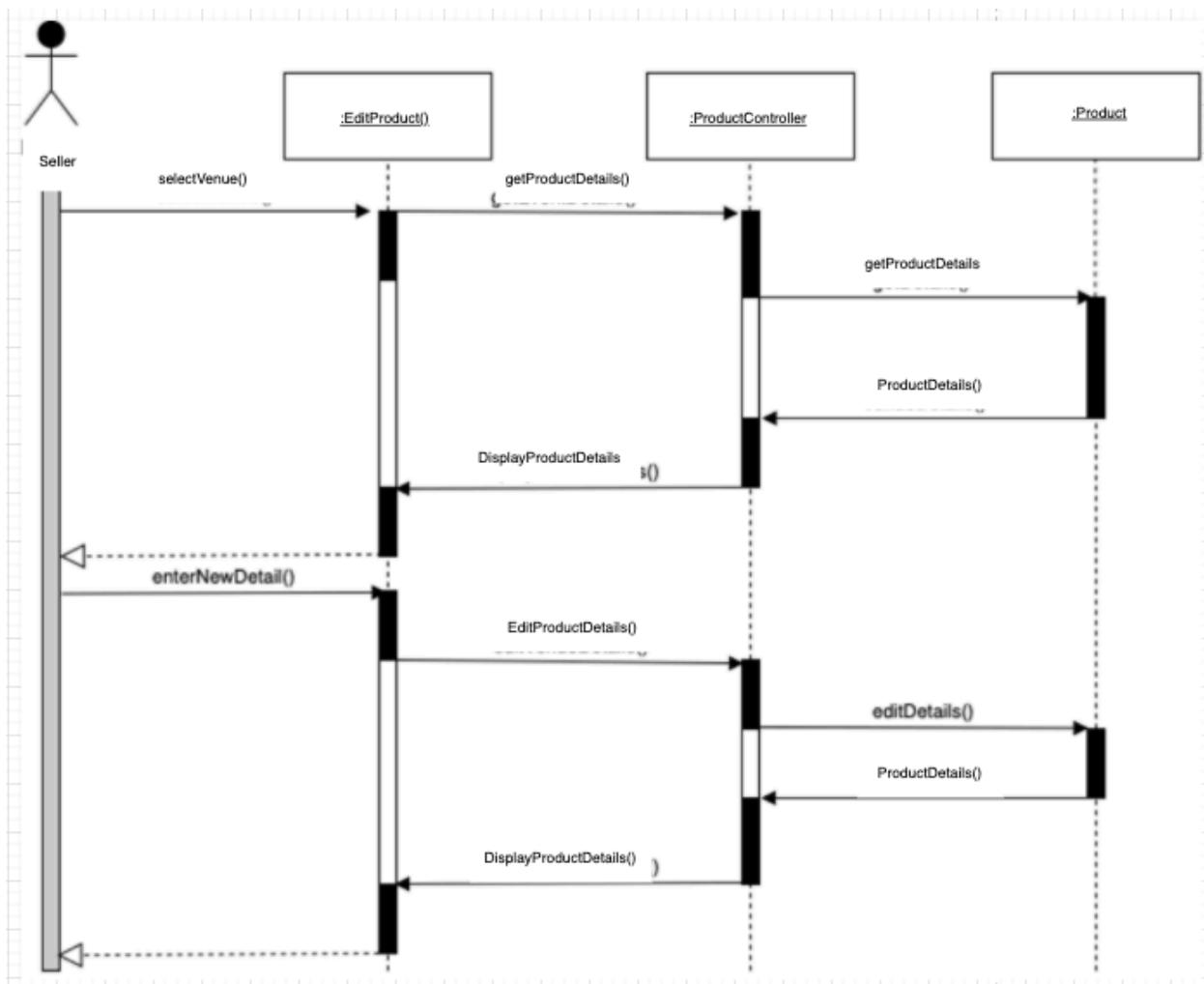


Figure 22 Sequence flow diagram 4

3.6.4.5 Delete Product

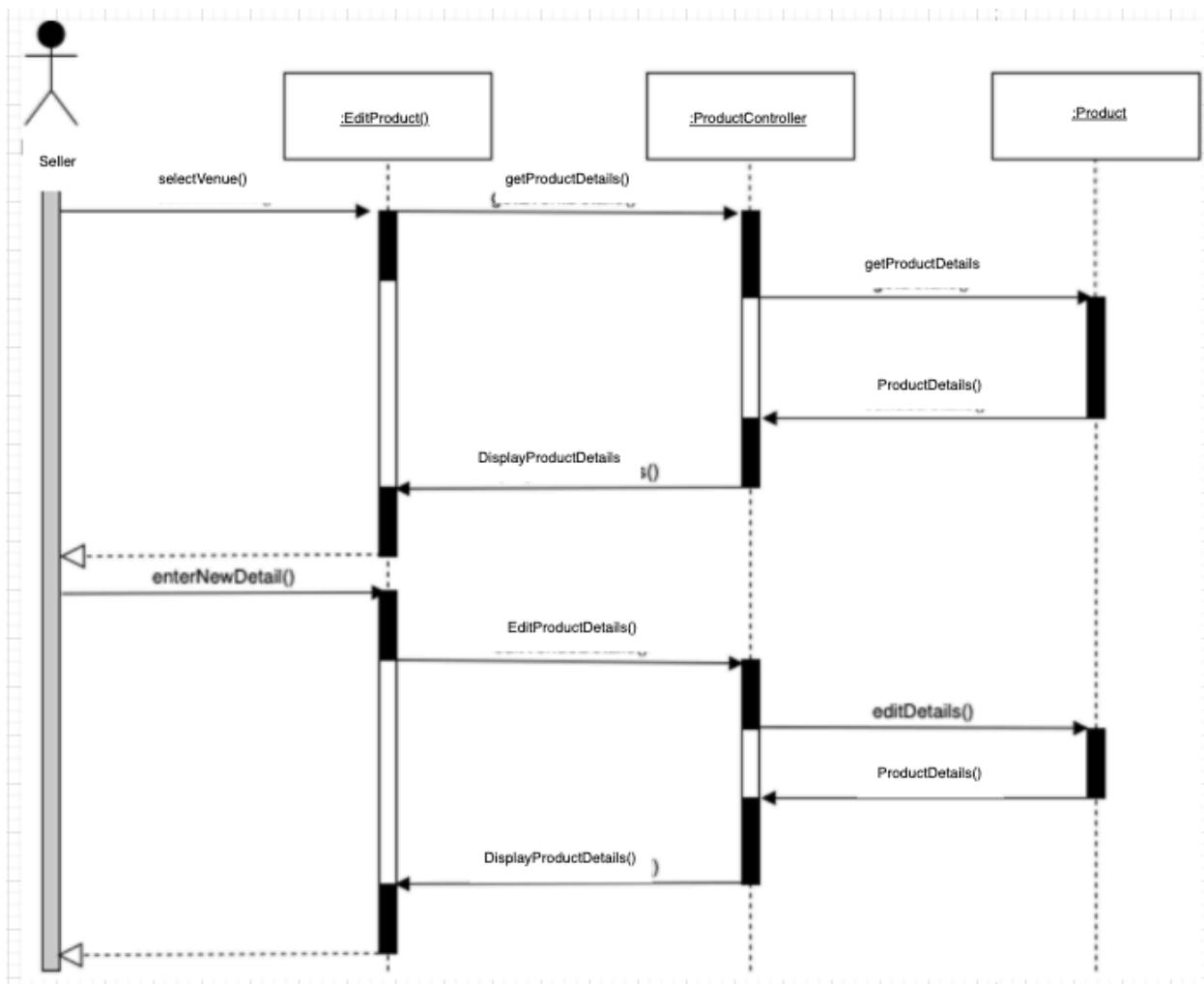


Figure 23 Sequence flow diagram 5

3.7 IMPLEMENTATION

To implement my idea to make multi-vendor ecommerce site. Python web framework Django is chosen due its easy and fast development process. After selecting the framework, I choose PyCharm IDLE as my IDE which is very famous python development. So, after doing a lot of research I came to find out that MySQL is perfect match database for my project as it very easy to use and was also taught in my college. To connect MySQL with Django I had to install pymysql using pip and then for hosting my database I choose mamp as my database hosting engine. For frontend I had used html, CSS, bootstrap, JavaScript, jQuery and some of the frontend opensource library. I had also installed pillow to support dynamic image in my project. Other library like Django-ckeditor was used as editor where seller can write the description of the product. After coding, using the different open source library and fixing bug with the help of my friend, colleague and teacher I was finally able to complete my system.

3.7.1 Frontend

3.7.1.1 Customer Frontend

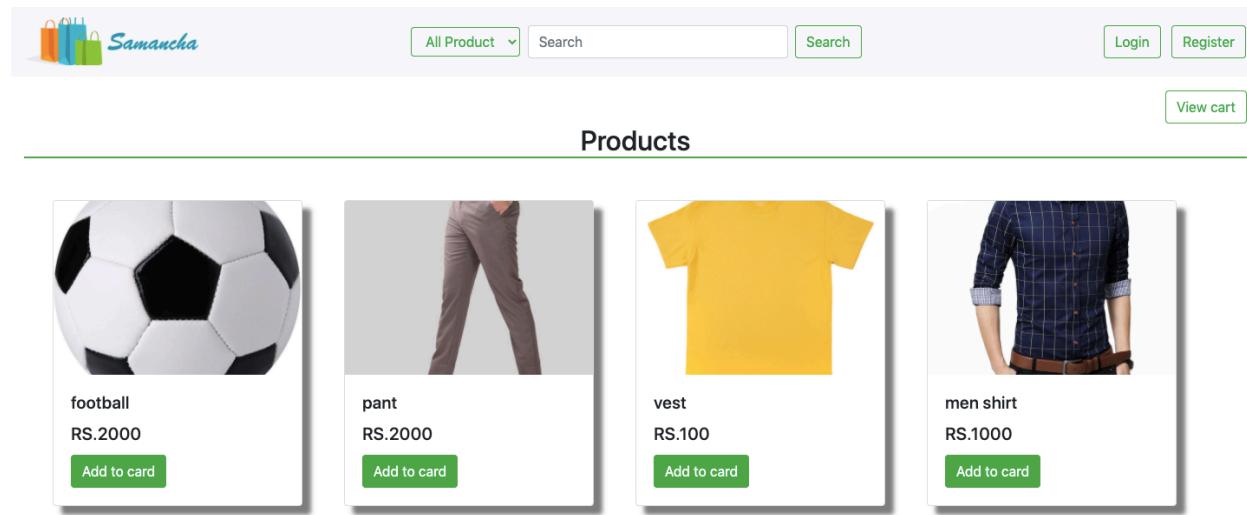


Figure 24 frontend(i)

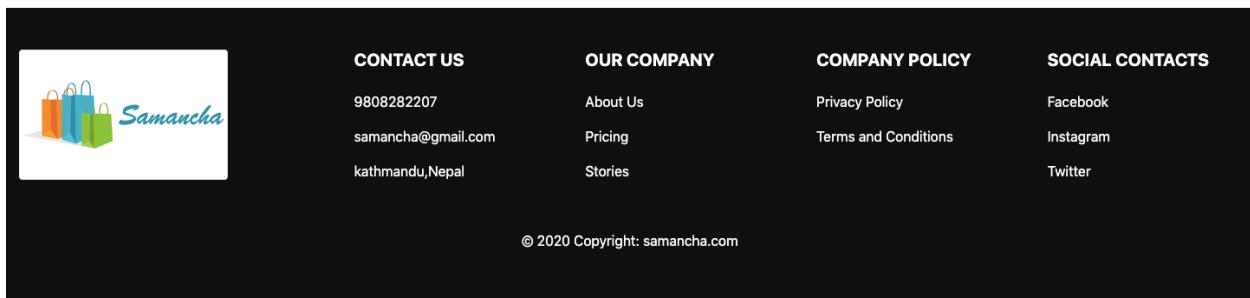


Figure 27 frontend(ii)

Product Name	Product Photo	Product Price	Action
football		2000	<button>Remove From Cart</button>
pant		2000	<button>Remove From Cart</button>

Showing 1 to 2 of 2 entries

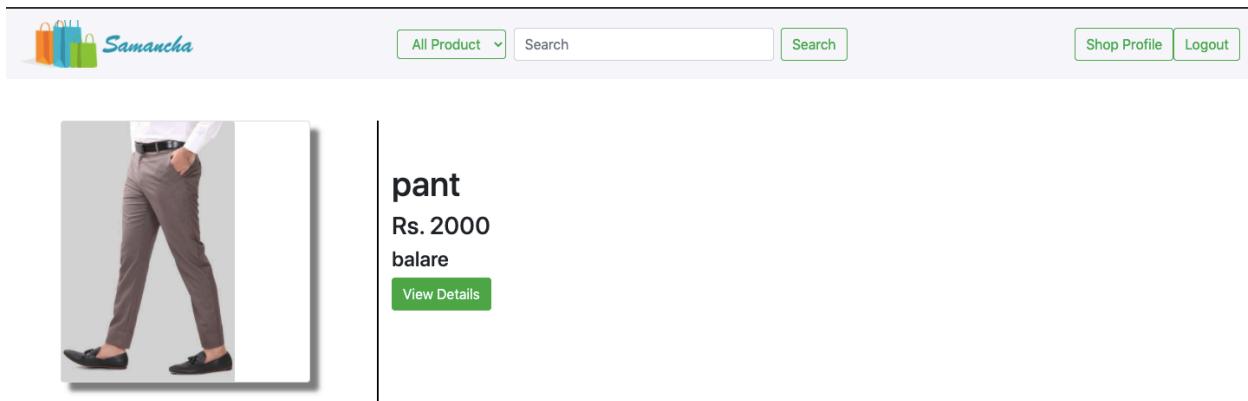
Total:4000
Check Out

Figure 25 frontend(iii)

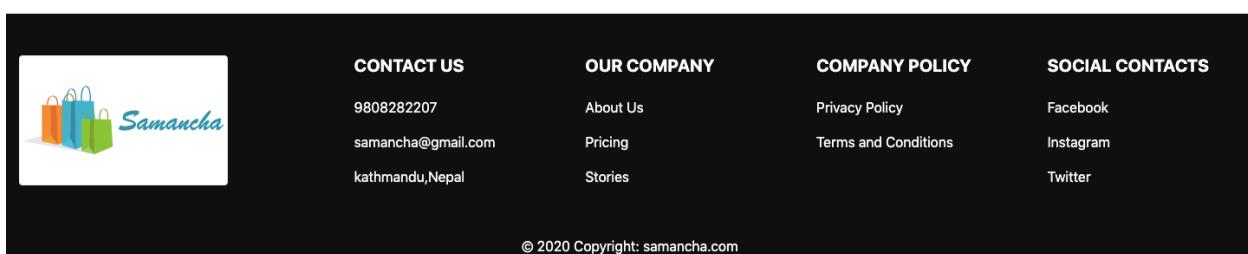
Product Name	Product Photo	Product Price	Status
football		2000	On the Way
pant		2000	In review

Showing 1 to 2 of 2 entries

Figure 26 frontend(iv)

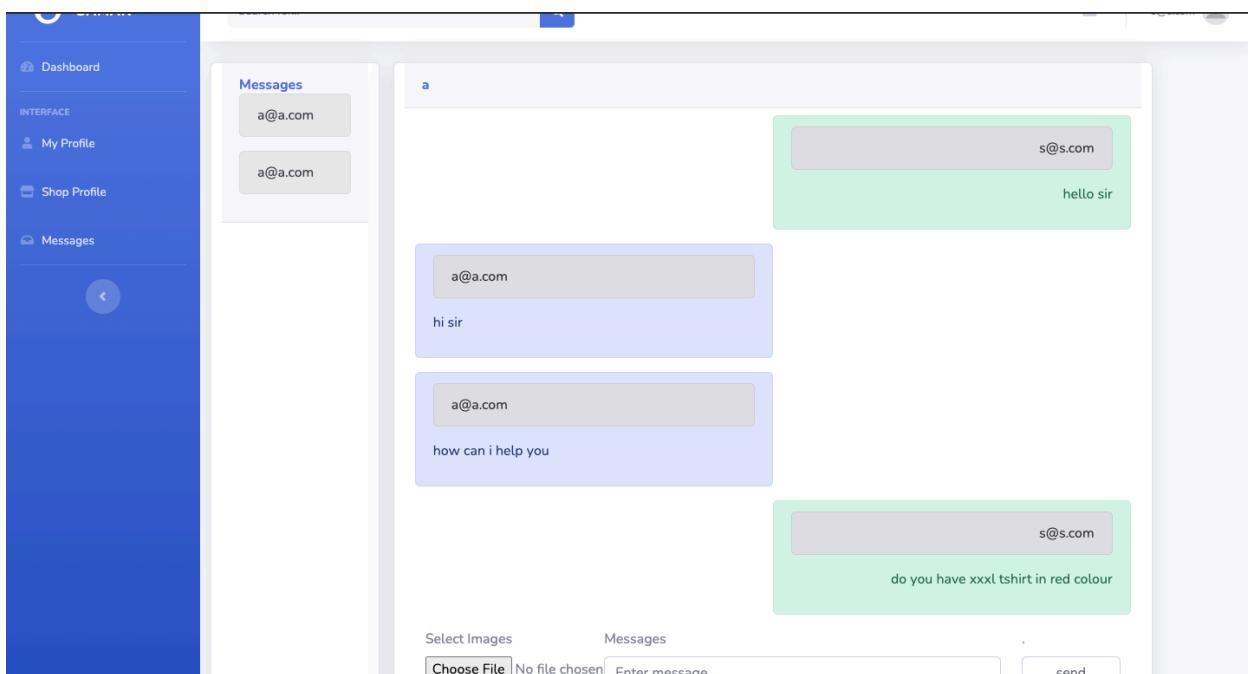


The screenshot shows a product listing for a pair of brown trousers. On the left is a thumbnail image of a person wearing the pants. To the right, the product name is listed as "pant" with a price of "Rs. 2000". Below the name is the brand name "balare". A green "View Details" button is visible at the bottom.



The footer section contains the company logo, contact information, company details, company policy, and social media links. The footer also includes a copyright notice: "© 2020 Copyright: samancha.com".

Figure 28 frontend(vi)



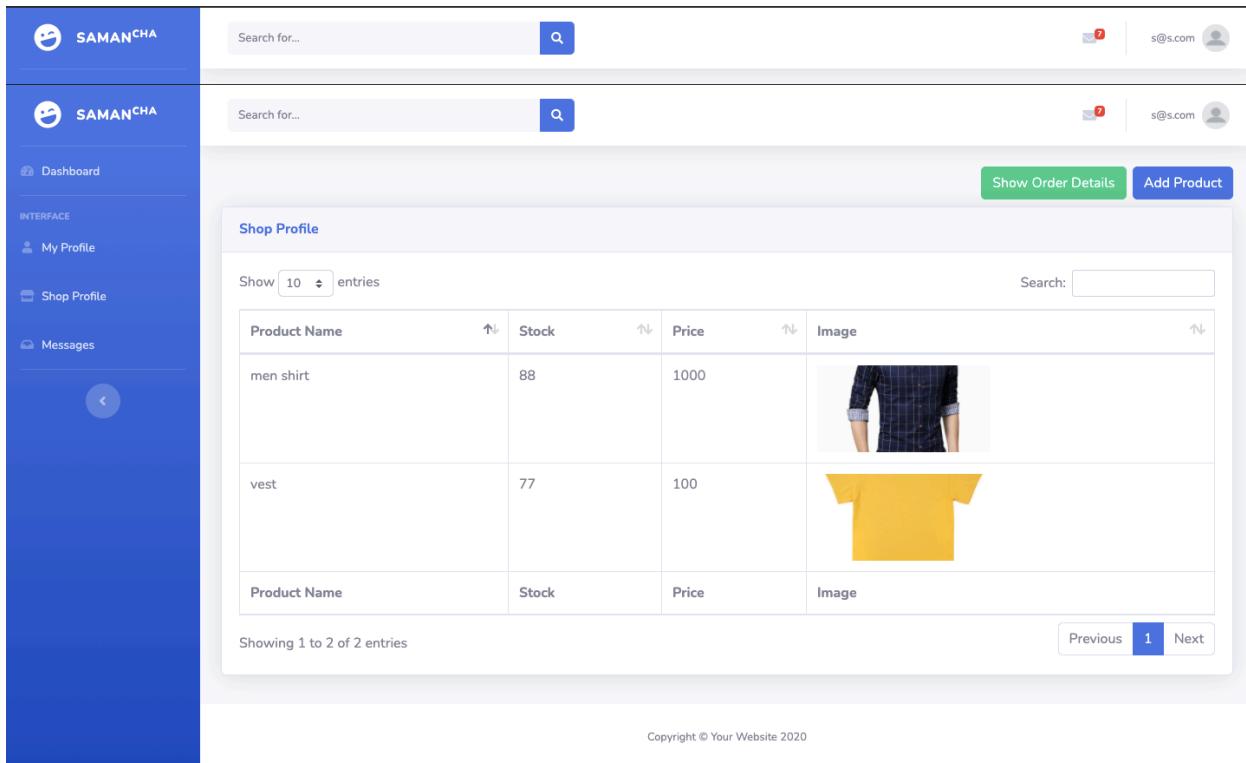
The screenshot shows a messaging interface between two users. The messages are:

- User 1 (a@a.com): "hi sir"
- User 1 (a@a.com): "how can i help you"
- User 2 (s@s.com): "hello sir"
- User 2 (s@s.com): "do you have xxl tshirt in red colour"

At the bottom, there are input fields for "Select Images" (with a "Choose File" button), "Messages" (with an "Enter message" input field), and a "send" button.

Figure 29 frontend(vii)

3.7.1.2 seller Frontend



The screenshot shows the SAMANCHA seller frontend interface. On the left, there is a sidebar with a blue background containing the SAMANCHA logo, a search bar, and a user profile section with a message icon (7 notifications) and an email address (s@s.com). The main content area has a white background. At the top, there is a navigation bar with a search bar, a magnifying glass icon, and a user profile icon. Below the navigation bar, there are two buttons: "Show Order Details" and "Add Product". The main area is titled "Shop Profile" and contains a table with two entries. The table has columns for "Product Name", "Stock", "Price", and "Image". The first entry is "men shirt" with stock 88 and price 1000, accompanied by an image of a blue plaid shirt. The second entry is "vest" with stock 77 and price 100, accompanied by an image of a yellow vest. Below the table, it says "Showing 1 to 2 of 2 entries". At the bottom right, there are navigation buttons for "Previous", "1", and "Next".

Product Name	Stock	Price	Image
men shirt	88	1000	
vest	77	100	

Figure 30 frontend(viii)

The screenshot shows the 'My Order Details' section of the SAMANCHA application. The left sidebar includes links for Dashboard, My Profile, Shop Profile, and Messages. The main area has a search bar and a table with columns: Product Name, Product Photo, Product Price, Customer Name, Customer Address, and Status. Two entries are listed:

Product Name	Product Photo	Product Price	Customer Name	Customer Address	Status
football		2000	a	a	<button>Deliver item</button>
pant		2000	a	a	<button>Item in review send item</button>

Showing 1 to 2 of 2 entries

Figure 31 frontend(ix)

The screenshot shows a message conversation between two users. The left sidebar includes links for Dashboard, My Profile, Shop Profile, and Messages. The main area shows a list of messages with senders 'a@a.com' and 's@s.com'. The messages are:

- s@s.com: hello sir
- a@a.com: hi sir
- a@a.com: how can i help you
- s@s.com: do you have xxl tshirt in red colour

At the bottom, there are buttons for 'Select Images' and 'Messages'.

Figure 32 frontend(x)

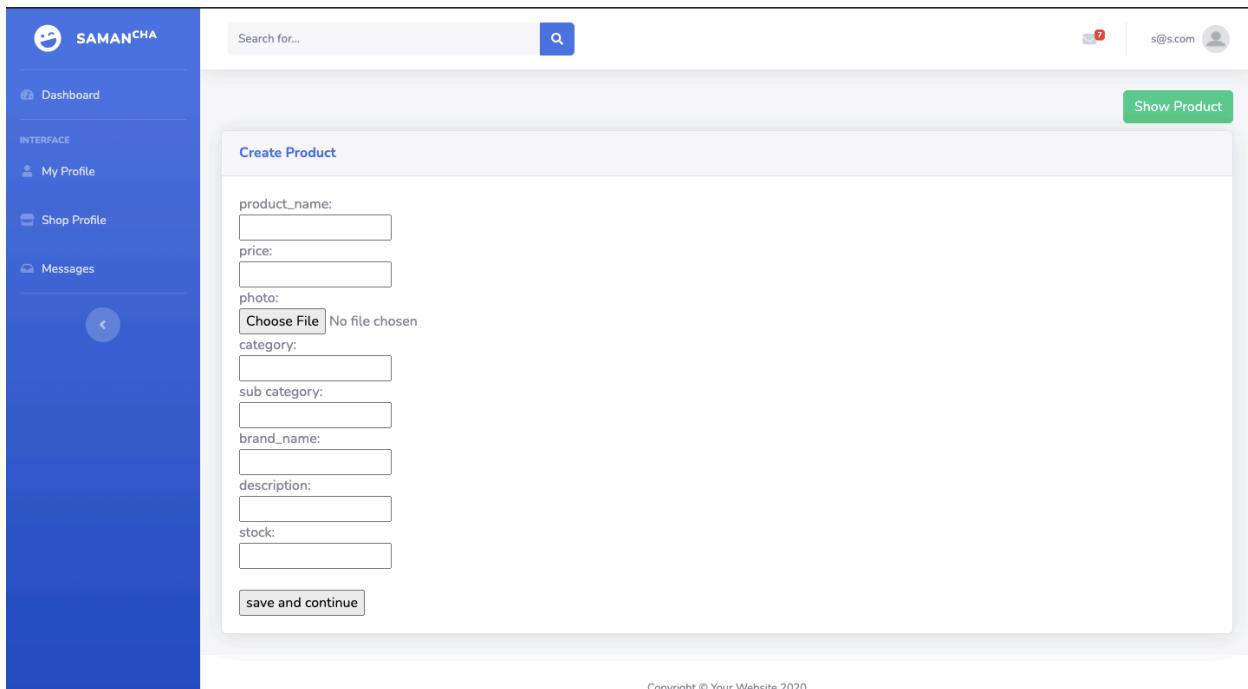


Figure 33 frontend(xi)

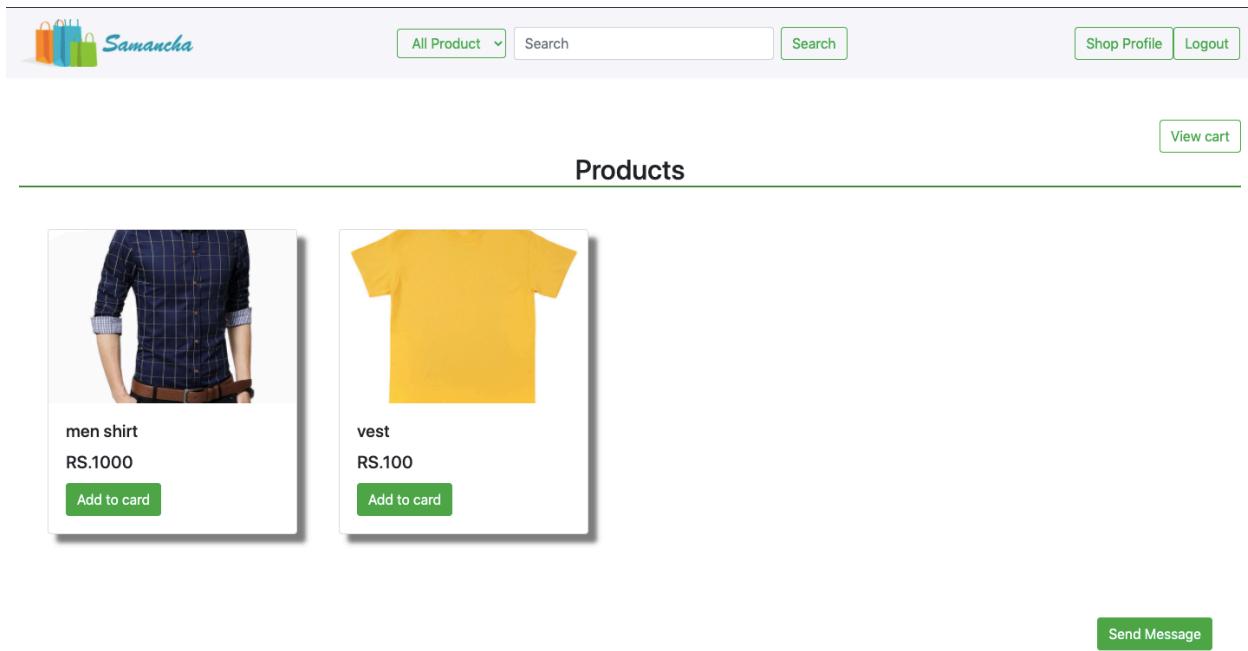


Figure 34 frontend(xii)

3.7.2 Backend

```

from django.contrib.auth.decorators import login_required
from django.shortcuts import render, redirect
from .forms import ProductsForm
from .models import Products
from shop_profile.models import ShopProfile
from ecommerce.decorators import is_seller
from django.core.exceptions import PermissionDenied

# Create your views here.

@login_required
@is_seller()
def product_create(request, username):
    if not ShopProfile.objects.filter(owner=request.user).filter(shop_user_name=username).exists():
        raise PermissionDenied
    if request.method == 'POST':
        if not ShopProfile.objects.filter(owner=request.user).filter(shop_user_name=username).exists():
            raise PermissionDenied

        forms = ProductsForm(request.POST, request.FILES)
        if forms.is_valid():
            product = Products(
                shop_product=ShopProfile.objects.get(shop_user_name=username),
                owner=request.user,
                product_name=request.POST.get('product_name'),
                price=request.POST.get('price'),
                photo=request.FILES.get('photo'),
                category=request.POST.get('category'),
                sub_category=request.POST.get('sub_category'),
                brand_name=request.POST.get('brand_name'),
                description=request.POST.get('description'),
                stock=request.POST.get('stock'),
            )
            product.save()
        else:
            print(forms.errors)
    return render(request, 'product/home.html', {"username": username})

@login_required()
@is_seller()
def show_product(request, username):
    if not ShopProfile.objects.filter(owner=request.user).filter(shop_user_name=username).exists():
        raise PermissionDenied
    product = Products.objects.filter(shop_product__shop_user_name=username)
    context = {"products": product, "username": username}
    return render(request, 'product/show_product.html', context)

```

Figure 35 backend(i)

```
from accounts.models import MyUser
from shop_profile.models import ShopProfile
from django.utils.text import slugify

# Create your models here.

class Products(models.Model):
    shop_product = models.ForeignKey(ShopProfile, on_delete=models.CASCADE)
    owner = models.ForeignKey(MyUser, on_delete=models.CASCADE)
    slug = models.SlugField(max_length=255)
    product_name = models.CharField(max_length=100)
    price = models.IntegerField()
    photo = models.ImageField()
    category = models.CharField(max_length=100)
    sub_category = models.CharField(max_length=100)
    brand_name = models.CharField(max_length=100)
    description = models.CharField(max_length=1000)
    stock = models.IntegerField()
    created_date = models.DateTimeField(auto_now_add=True)

    def save(self, *args, **kwargs):
        self.slug = slugify(self.product_name + "-" + str(self.id))
        super(Products, self).save(*args, **kwargs)
```

Figure 36 backend(ii)

```

class MyUser(AbstractBaseUser):
    email = models.EmailField(
        verbose_name='email address',
        max_length=255,
        unique=True,
    )
    username = models.CharField(
        verbose_name='user name',
        max_length=20,
        unique=True,
    )
    date_of_birth = models.DateField()
    is_email_verified = models.BooleanField(default=False)
    is_profile_verified = models.BooleanField(default=False)
    is_customer = models.BooleanField(default=False)
    is_seller = models.BooleanField(default=False)
    is_active = models.BooleanField(default=True)
    is_admin = models.BooleanField(default=False)

    objects = MyUserManager()

    USERNAME_FIELD = 'email'
    REQUIRED_FIELDS = ['date_of_birth', 'username']

    def __str__(self):
        return self.email

    def has_perm(self, perm, obj=None):
        """Does the user have a specific permission?"""
        # Simplest possible answer: Yes, always
        return True

    def has_module_perms(self, app_label):
        """Does the user have permissions to view the app `app_label`?"""
        # Simplest possible answer: Yes, always
        return True

    @property
    def is_staff(self):
        """Is the user a member of staff?"""
        # Simplest possible answer: All admins are staff
        return self.is_admin

```

Figure 37 backend(iii)

```

def customer_register(request):
    if request.user.is_authenticated:
        return redirect('landing_page:landing_page')
    if request.method == 'POST':
        data = request.POST.copy()
        forms = RegisterForm(data)
        if forms.is_valid():
            user = MyUser(
                email=data['email'],
                password=make_password(data['password']),
                username=data['username'],
                date_of_birth=data['date_of_birth'],
                is_customer=True
            )
            user.save()
            user = auth.authenticate(email=data['email'], password=data['password'])
            if user is not None:
                auth.login(request, user)
                return redirect('landing_page:landing_page')
            return render(request, 'accounts/customer_register.html')
        else:
            print(forms.errors)
            context = {
                "error": forms.errors,
                "email": data['email'],
                "username": data['username'],
                "date_of_birth": data['date_of_birth']
            }
            return render(request, 'accounts/customer_register.html', context)

    return render(request, 'accounts/customer_register.html')

```

Figure 38 backend(iv)

```

from django.db import models
from accounts.models import MyUser


# Create your models here.

class SellerProfile(models.Model):
    owner = models.OneToOneField(MyUser, primary_key=True, on_delete=models.CASCADE)
    first_name = models.CharField(max_length=50)
    last_name = models.CharField(max_length=50)
    mobile_number = models.PositiveIntegerField()
    citizenship_number = models.PositiveIntegerField()
    citizenship_front = models.ImageField()
    citizenship_back = models.ImageField()
    temp_add = models.CharField(max_length=50)
    per_add = models.CharField(max_length=50)
    is_verified = models.BooleanField(default=False)

```

Figure 39 backend(v)

```

def seller_register(request):
    if request.user.is_authenticated:
        return redirect('landing_page:landing_page')
    if request.method == 'POST':
        data = request.POST.copy()
        forms = RegisterForm(data)
        print(data['username'])
        if forms.is_valid():
            print("form is valid")
            user = MyUser(
                email=data['email'],
                password=make_password(data['password']),
                username=data['username'],
                date_of_birth=data['date_of_birth'],
                is_seller=True,
            )
            user.save()
            user = auth.authenticate(email=data['email'], password=data['password'])
            if user is not None:
                auth.login(request, user)
                return redirect('seller_profile:seller_profile_create')
            return render(request, 'accounts/seller_registration.html')
        else:
            context = {"errors": forms.errors}
            return render(request, 'accounts/seller_registration.html', context)
    return render(request, 'accounts/seller_registration.html')

```

Figure 40 backend(vi)

```
def login(request):
    form = {}
    if request.user.is_authenticated:
        return redirect('landing_page:landing_page')
    else:
        if request.method == 'POST':
            email = request.POST['email']
            password = request.POST['password']
            user = auth.authenticate(email=email, password=password)
            if user is not None:
                auth.login(request, user)
                return redirect('landing_page:landing_page')
            else:
                errors = "User name or password is incorrect"
                return render(request, 'accounts/login.html', {"errors": errors})
        return render(request, 'accounts/login.html', {"form": form})

def logout(request):
    auth.logout(request)
    return redirect('landing_page:landing_page')
```

Figure 41 backend(vii)

```

from django.contrib.auth.decorators import login_required
from django.shortcuts import render, redirect
from .forms import SellerProfileForm
from .models import SellerProfile
from ecommerce.decorators import is_seller

# Create your views here.
@login_required()
@is_seller()
def seller_profile_create(request):
    if SellerProfile.objects.filter(owner=request.user).exists():
        return redirect('seller_profile:seller_profile_view')
    if request.method == 'POST':
        forms = SellerProfileForm(request.POST, request.FILES)
        if forms.is_valid():
            seller_profile = SellerProfile(
                owner=request.user,
                first_name=request.POST['first_name'],
                last_name=request.POST['last_name'],
                mobile_number=request.POST['mobile_number'],
                citizenship_number=request.POST['citizenship_number'],
                temp_add=request.POST['temp_add'],
                per_add=request.POST['per_add'],
                citizenship_front=request.FILES['citizenship_front'],
                citizenship_back=request.FILES['citizenship_back'],
            )
            seller_profile.save()
            return redirect('shop_profile:shop_profile_show')
        else:
            context = {'errors': forms.errors}
            return render(request, 'seller_profile/seller_profile_create.html', context)
    return render(request, 'seller_profile/seller_profile_create.html')

@login_required()
@is_seller()
def seller_profile_view(request):
    if not SellerProfile.objects.filter(owner=request.user).exists():
        return redirect('seller_profile:seller_profile_create')
    profile = SellerProfile.objects.filter(owner=request.user)
    context = {"profile": profile}
    return render(request, 'seller_profile/seller_profile_show.html', context)

```

Figure 42 backend(viii)

```
from django.db import models
from accounts.models import MyUser

# Create your models here.

class ShopProfile(models.Model):
    owner = models.ForeignKey(MyUser, on_delete=models.CASCADE)
    shop_registered_name = models.CharField(max_length=100)
    shop_name = models.CharField(max_length=100)
    shop_user_name = models.CharField(max_length=10, unique=True)
    shop_owner_name = models.CharField(max_length=150)
    shop_address = models.CharField(max_length=100)
    shop_warehouse_address = models.CharField(max_length=100)
    shop_returned_address = models.CharField(max_length=100)
    shop_category = models.CharField(max_length=100)
    shop_detail_description = models.CharField(max_length=1000)
    shop_pan_card = models.ImageField()
    shop_phone_number = models.BigIntegerField()
    shop_owner_phone_number = models.BigIntegerField()
    is_verified = models.BooleanField(default=False)
    is_blocked = models.BooleanField(default=False)

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

Figure 43 backend(ix)

```

from django.db import models
from accounts.models import MyUser

# Create your models here.

class Chat(models.Model):
    send_from = models.ForeignKey(MyUser, on_delete=models.CASCADE, related_name='send_from')
    send_to = models.ForeignKey(MyUser, on_delete=models.CASCADE, related_name='send_to')
    message = models.CharField(max_length=500, blank=True)
    image = models.ImageField(blank=True)
    seen = models.BooleanField(default=False)
    send_date = models.DateTimeField(auto_now_add=True)
    seen_date = models.DateTimeField(blank=True, null=True)

    def __str__(self):
        return f'{self.send_from}_to_{self.send_to}'

```

Figure 44 backend(ix)

```

@login_required()
def send_messages(request, username):
    if not MyUser.objects.filter(username=username).exists():
        return HttpResponseRedirect('no user exist')
    send_to = MyUser.objects.get(username=username)

    chats = Chat.objects.filter(
        Q(
            Q(send_from=request.user) & Q(send_to=send_to)) |
        Q(
            Q(send_from=send_to) & Q(send_to=request.user)))
    .order_by('send_date')

    all_message = Chat.objects.filter(
        Q(send_to=request.user))
    .order_by('-send_date')

    if request.method == 'POST':
        message = request.POST.get('message')
        images = request.FILES['images'] if 'images' in request.FILES else False
        if not message and not images:
            return HttpResponseRedirect('nothing')
        forms = ChatForms(request.POST, request.FILES)
        if forms.is_valid():
            chat = Chat(
                send_from=request.user,
                send_to=send_to,
                message=message,
                image=images,
            )
            chat.save()

    context = {"username": username, "chats": chats, "all_messages": all_message}
    return render(request, 'chat/home.html', context)

```

Figure 45 backend(x)

```

from django.contrib.auth.decorators import login_required
from django.shortcuts import render, redirect
from .forms import ShopProfileForm
from .models import ShopProfile
from ecommerce.decorators import is_seller

# Create your views here.
@login_required
@is_seller()
def shop_profile_create(request):
    if request.method == 'POST':
        forms = ShopProfileForm(request.POST, request.FILES)
        if forms.is_valid():
            shop_profile = ShopProfile(
                owner=request.user,
                shop_registered_name=request.POST.get('shop_registered_name'),
                shop_name=request.POST.get('shop_name'),
                shop_user_name=request.POST.get('shop_user_name'),
                shop_owner_name=request.POST.get('shop_owner_name'),
                shop_address=request.POST.get('shop_address'),
                shop_warehouse_address=request.POST.get('shop_warehouse_address'),
                shop_returned_address=request.POST.get('shop_returned_address'),
                shop_category=request.POST.get('shop_category'),
                shop_detail_description=request.POST.get('shop_detail_description'),
                shop_pan_card=request.FILES.get('shop_pan_card'),
                shop_phone_number=request.POST.get('shop_phone_number'),
                shop_owner_phone_number=request.POST.get('shop_owner_phone_number'),
            )
            shop_profile.save()
            return redirect('shop_profile:shop_profile_show')
        else:
            context = {"error": forms.errors}
            return render(request, 'shop_profile/shop_profile_create.html', context)
    return render(request, 'shop_profile/shop_profile_create.html')

@login_required
@is_seller()
def show_shop_profile(request):
    my_shop_profile = ShopProfile.objects.filter(owner=request.user)
    context = {"shop_profile": my_shop_profile}
    return render(request, 'shop_profile/shop_profile_show.html', context)

```

Figure 46 backend(xi)

```

from django.shortcuts import render, HttpResponseRedirect
from products.models import Products

# Create your views here.
from shop_profile.models import ShopProfile


def landing_page(request):
    products = Products.objects.all().order_by('-created_date')
    context = {"products": products}
    return render(request, 'landing_page/landing_page.html', context)

def shop_landing_page(request, username):
    if not ShopProfile.objects.filter(shop_user_name=username).exists():
        return HttpResponseRedirect('Shop Profile Doesnot Exist')
    products = Products.objects.filter(shop_product__shop_user_name=username)
    context = {"products": products, "username": username}
    return render(request, 'landing_page/shop_landing_page.html', context)

def search(request):
    search_type = request.GET.get('search_type')
    search_key = request.GET.get('search')
    context = {}
    if search_type == 'shop':
        search_result = ShopProfile.objects.filter(shop_user_name=search_key).order_by('-id')[:10]
        context['search_result'] = search_result
    elif search_type == 'product':
        search_result = Products.objects.filter(product_name=search_key).order_by('-id')[:10]
        context['search_result'] = search_result
    print(context)
    return render(request, 'landing_page/search.html', context)

```

Figure 47 backend(xii)

```
from django.db import models
from django.contrib.auth.models import (
    BaseUserManager, AbstractBaseUser
)

class MyUserManager(BaseUserManager):
    def create_user(self, email, date_of_birth, username, password=None):
        """
        Creates and saves a User with the given email, date of
        birth and password.
        """
        if not email:
            raise ValueError('Users must have an email address')

        user = self.model(
            email=self.normalize_email(email),
            date_of_birth=date_of_birth,
            username=username,
        )

        user.set_password(password)
        user.save(using=self._db)
        return user
```

Figure 48 backend(xiii)

CHAPTER 4: TESTING AND ANALYSIS

4.1 TEST PLAN

4.1.1 UNIT TESTING, TEST PLAN

S.No	Test Case
1	Check If user can login with empty email or password
2	Check if user can login with invalid email format
3	Check if user can register with empty field
4	Check if user can register with password less than 7 digits
5	Check if user can register with email that already exist in database
6	Check if user can create profile with empty field
7	Check if user can put mobile number length more than 10 digit
8	Check if seller can upload .txt file instead of image in citizenship card
9	Check if seller can create shop with empty field
10	Check if seller can create product with empty field
11	Check if seller create product by putting character in price
12	Check if seller can put product price negative
13	Check if seller can put character in stock
14	Check if seller can put stock price negative
15	Check if one seller can access the shop information of another seller through URL
16	Check if customer can access to seller dashboard through URL
17	Check if user can add items to cart without logged in
18	Check if user can register as seller or customer
19	Check if user password is encrypted after successful registration of user
20	Check if user can login with valid email and password
21	Check if logged in seller can add shop
22	Check if logged in seller can add product to shop
23	Check if logged in seller can view all the product of their shop

24	Check if the added product is displayed in their particular shop landing page
25	Check if logged in user can remove product from the shop
26	Check if logged in user can add product to cart

Table 2 Unit Testing plan

4.1.2 SYSTEM TESTING, TEST PLAN

S.No	Test Cases
1	Check if user can create seller profile
2	Check if user can place an order
3	Check if seller can perform CRUD on shop product

Table 3 System testing plan

4.2 UNIT TESTING

4.2.1 Checking if user can login with empty email or password

Objective	To check if user can login with empty email or password
Action	Click on the login button without filling email and password field
Excepted Result	Warning message generate saying please fill out the field
Actual Result	Warning message generated saying please fill out the field
Conclusion	Test successful

Table 4 unit test 1

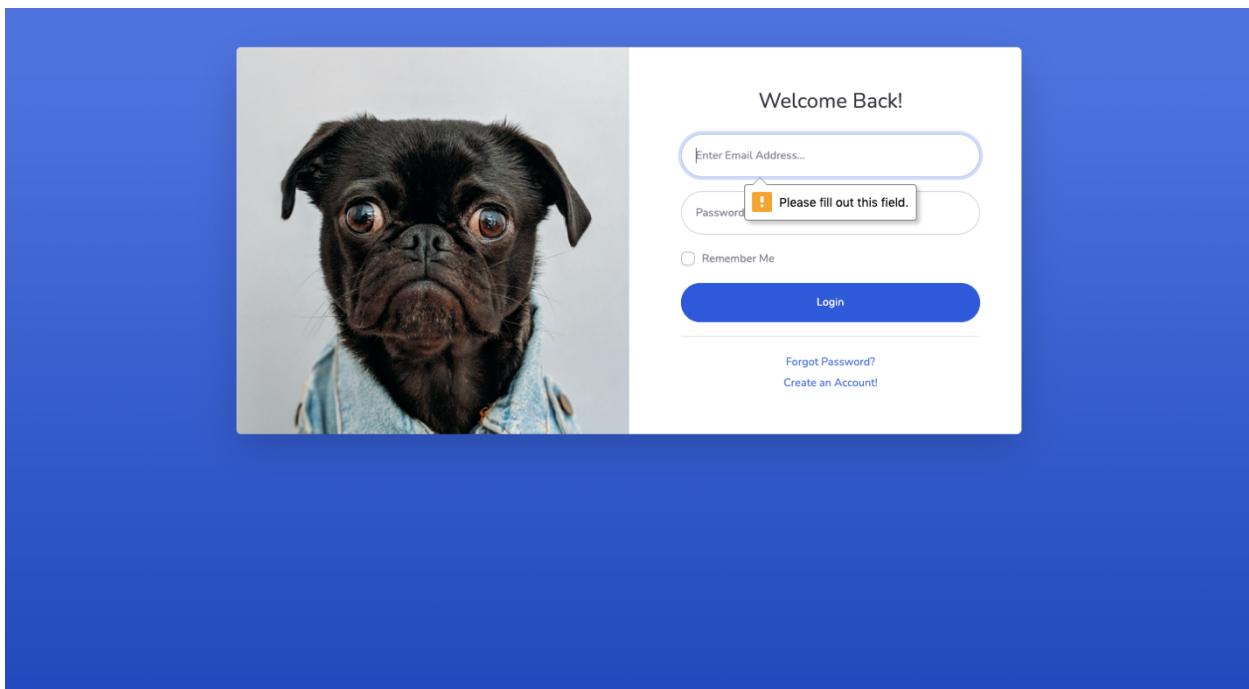


Figure 49 unit testing 1

4.2.2 Check if user can login with invalid email format

Objective	To Check if user can login with invalid email format
Action	<ul style="list-style-type: none"> • Input invalid email format • Enter password • Click on login button
Excepted Result	Warning message will generate saying please enter the valid email format followed by @
Actual Result	Warning message generated saying please enter the valid email address format saying please enter valid email address format followed by @
Conclusion	Test successful

Table 5 unit test 2

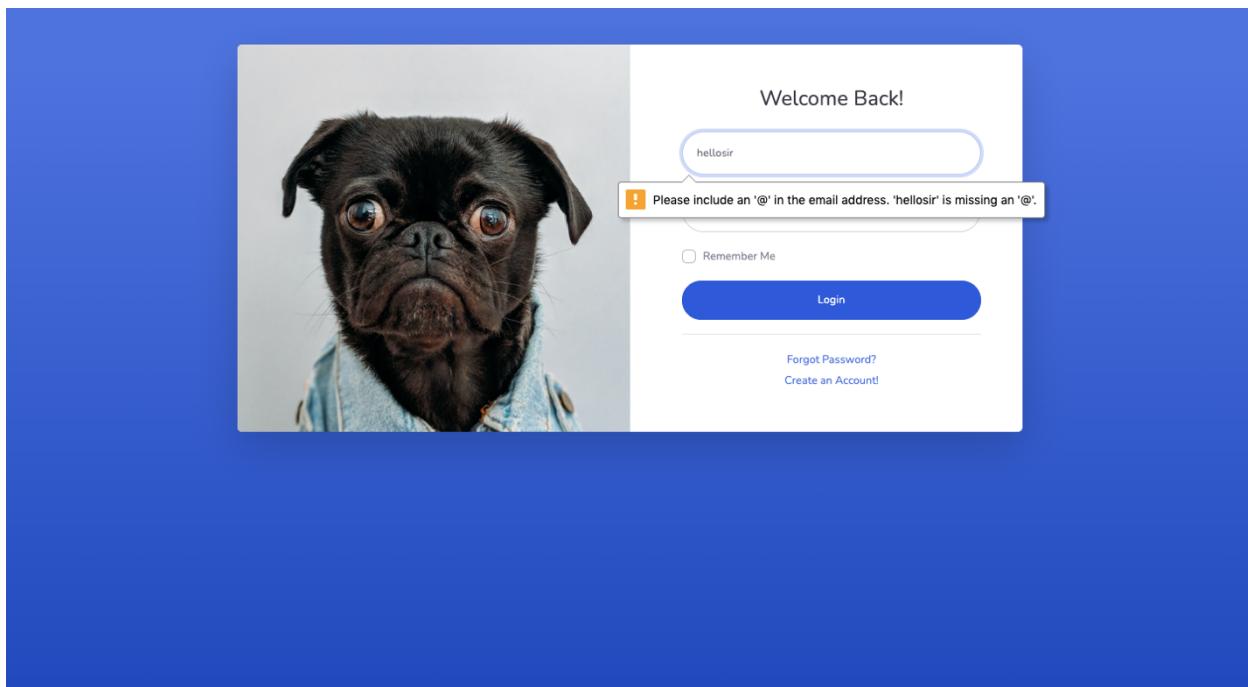


Figure 50 unit testing 2

4.2.3 Check if user can login with invalid credential

Objective	To Check if user can login with invalid credential
Action	<ul style="list-style-type: none"> Enter invalid email and password Click on login button
Excepted Result	Error message will be generated saying User name or password is incorrect.
Actual Result	Error message generated saying username or password is incorrect.
Conclusion	Test Successful

Table 6 unit test 3

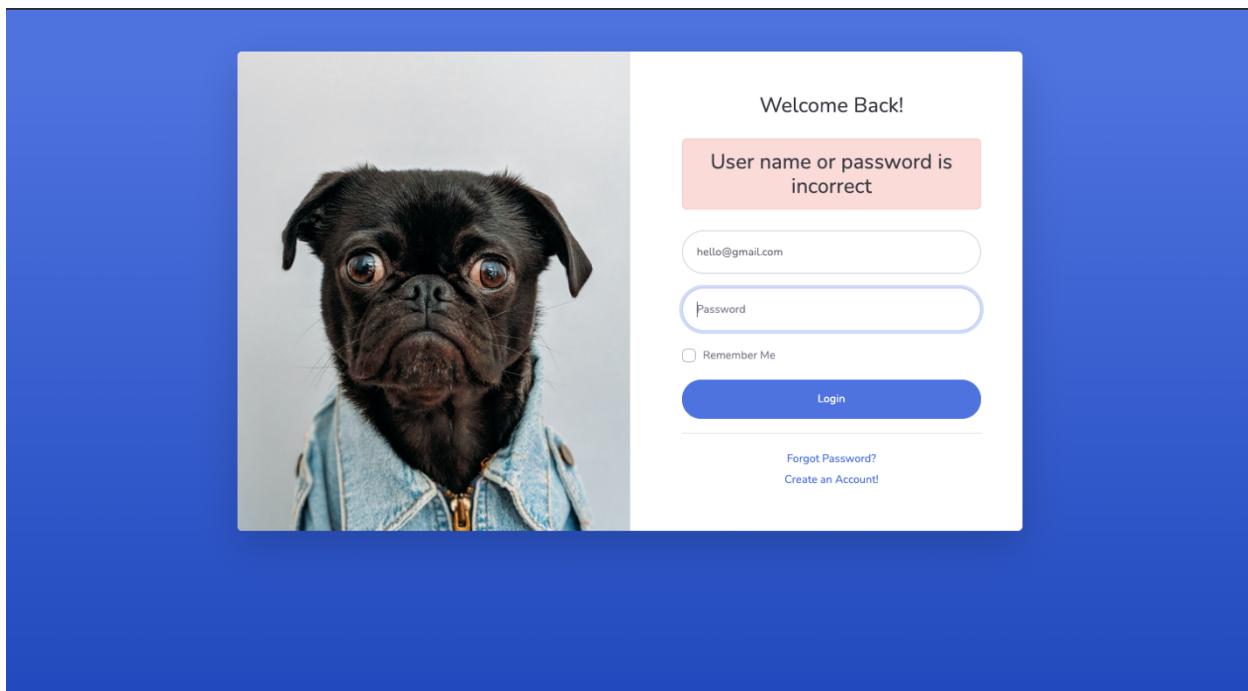


Figure 51 unit testing 3

4.2.4 Check if user can register with empty field

Objective	To Check if user can register with empty field
Action	<ul style="list-style-type: none"> Click on register button without filling any form
Excepted Result	Warning message will be generated saying please fill out this form
Actual Result	Warning message generated saying please fill out this form
Conclusion	Test Successful

Table 7 unit test 4

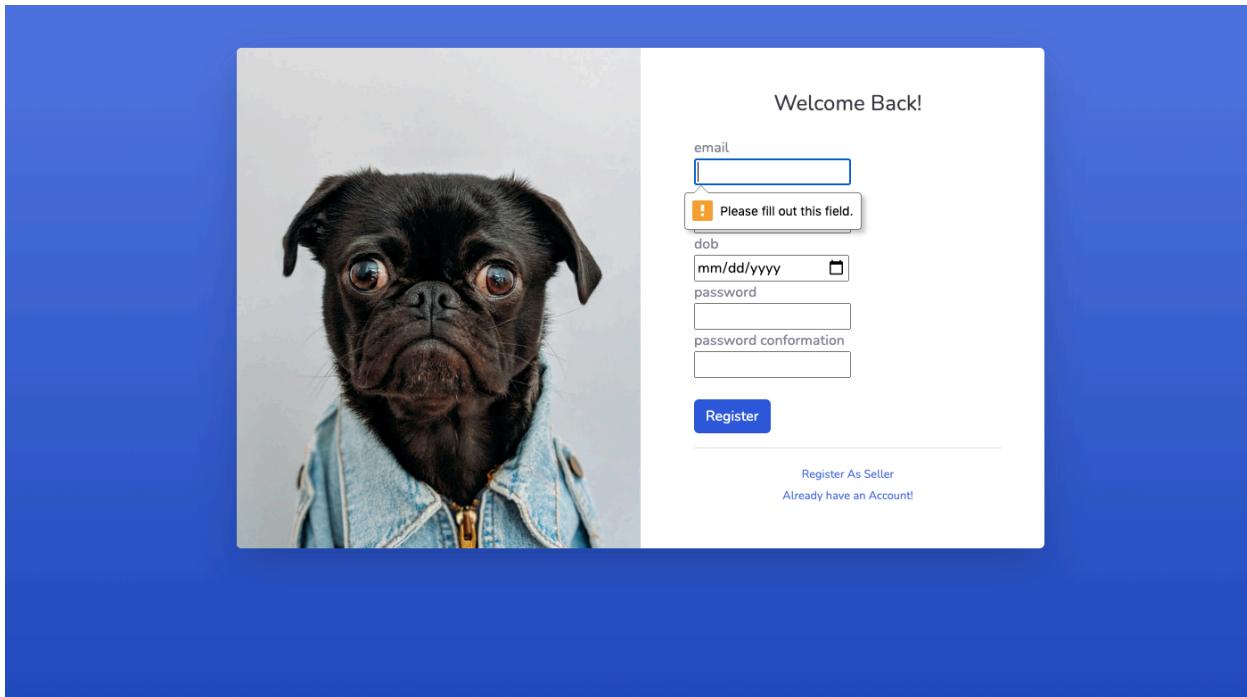


Figure 52 unit testing 4

4.2.5 Check if user can register with password less than 7 digits

Objective	To Check if user can register with password less than 7 digits
Action	<ul style="list-style-type: none"> Fill all the field Input password filled with less than 7 digits Click on register form
Excepted Result	Warning message will be generated saying password length must be at least 7 character.
Actual Result	Warning message generated saying length of password must be at least 7
Conclusion	Test Successful

Table 8 unit test 5

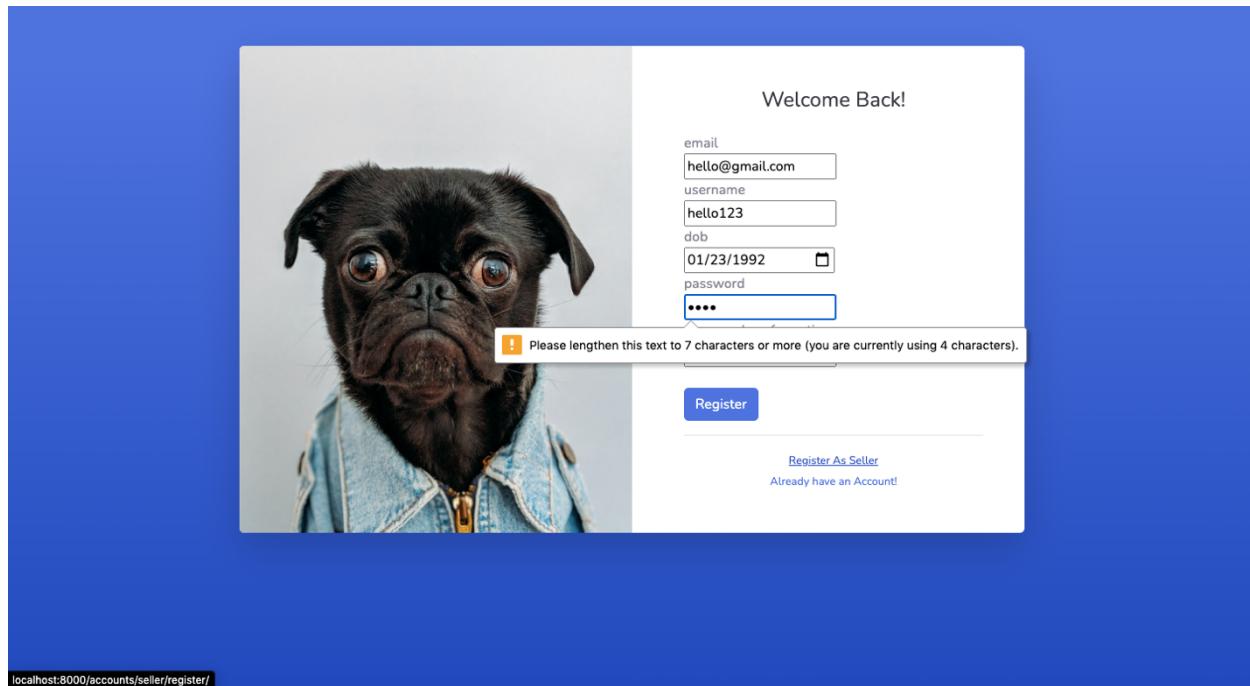


Figure 53 unit testign 5

4.2.6 Check if user can register with email that already exist in database

Objective	To Check if user can register with email that already exist in database
Action	<ul style="list-style-type: none"> Fill the email field with exiting email address Click on register form
Excepted Result	Error message should be generated saying user with this email already exist.
Actual Result	Error message generated saying hello@gmail.com email already exist
Conclusion	Test Successful

Table 9 unit test 6

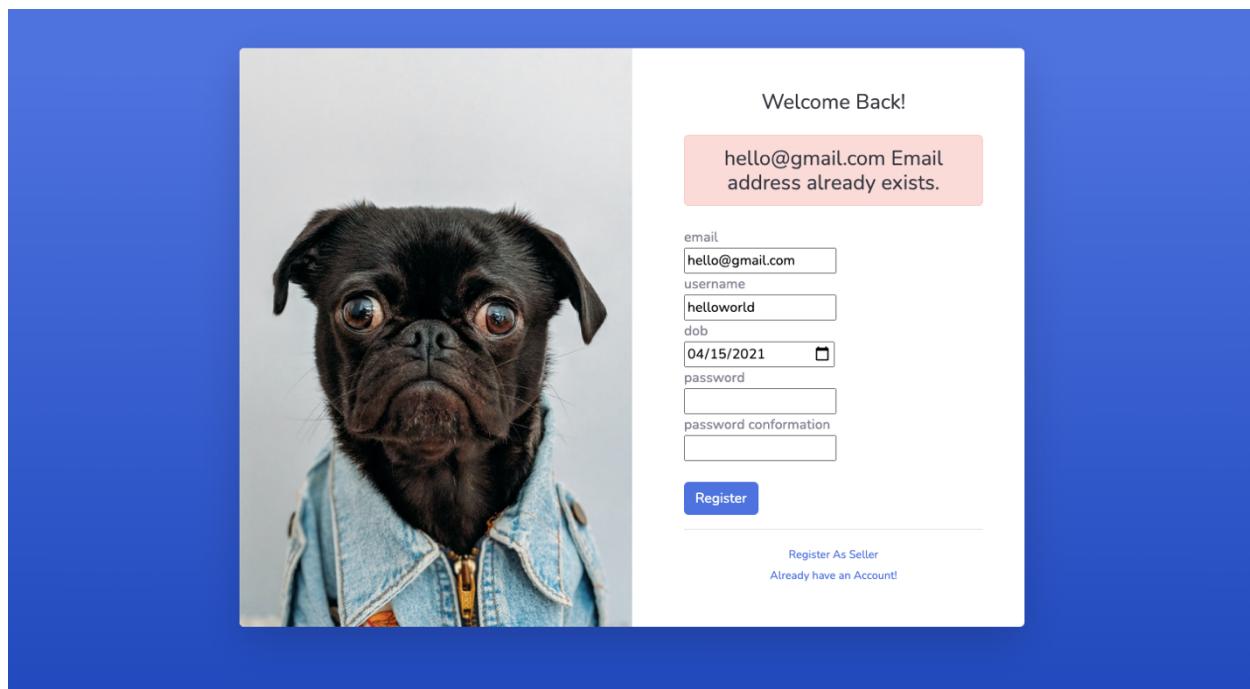


Figure 54 unit testing 6

4.2.7 Check if user can put mobile number length more than 10 digit

Objective	Check if user can put mobile number length more than 10 digit
Action	<ul style="list-style-type: none"> • Fill all the field • Fill mobile number less with less digit 10 • Click on
Excepted Result	Warning message should be generated saying mobile number should be of greater than or equals to 9800000000.
Actual Result	Warning message generated saying mobile number should be of greater than or equals to 9800000000.
Conclusion	Test Successful

Table 10 unit test 7

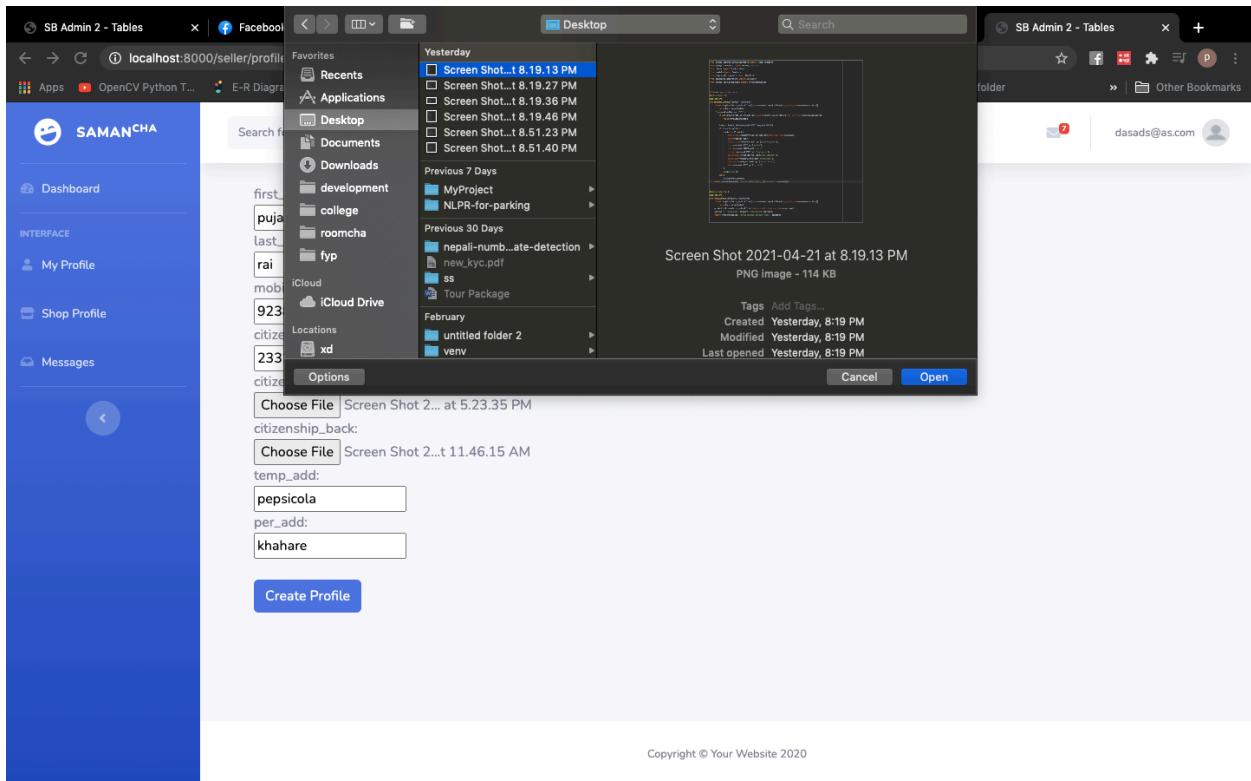
The screenshot shows a web interface for creating a profile. On the left, there's a sidebar with icons for Dashboard, My Profile, Shop Profile, and Messages. The main area has a search bar at the top. Below it, there are input fields for first_name ('pujan'), last_name ('rai'), and mobile_number ('9939303'). A validation error message 'Value must be greater than or equal to 9800000000.' is displayed next to the mobile_number field. There are also fields for citizenship_front ('Choose File') and citizenship_back ('Choose File'), both with placeholder text 'Screen Shot 2... at 5.09.10 PM'. Other fields include temp_add ('pepsicola') and per_add ('bkt'). At the bottom is a 'Create Profile' button.

Figure 55 unit testing7

4.2.8 Check if seller can upload .txt file instead of image in citizenship card

Objective	To Check if seller can upload other file instead of image in citizenship card
Action	<ul style="list-style-type: none"> Try to select file which doesn't have image extension
Excepted Result	User should be able to select others file except the image
Actual Result	Cannot select other file except the images
Conclusion	Test Successful

Table 11unit test 8



4.2.9 Check if seller create product by putting character in price

Objective	To Check if seller create product by putting character in price
Action	<ul style="list-style-type: none"> Try to input character instead of number
Excepted Result	User should not be allowed to enter any character except integer value
Actual Result	User can enter number
Conclusion	Test Successful

Table 12 unit test 9

The screenshot shows a 'Create Product' form with the following fields:

- product_name: rice
- price: (empty input field)
- photo: Choose File (No file chosen)
- category: food
- sub category: healty
- brand_name: good
- description: this is best food in towr
- stock: 12

At the bottom right of the form is a green 'Show Product' button.

Copyright © Your Website 2020

Figure 56 unit testing 9

4.2.10 Check if seller can put product price negative

Objective	To Check if seller can put product price negative
Action	<ul style="list-style-type: none"> Put negative number in price Click on save
Excepted Result	Waring message should be generated saying price should be 0 or greater
Actual Result	Warning message generated saying price should 0 or greater
Conclusion	Test Successful

Table 13 unit test 10

The screenshot shows a web-based application interface for managing products. On the left, there's a sidebar with navigation links: Dashboard, INTERFACE, My Profile, Shop Profile, and Messages. The main area is titled 'Create Product'. It has fields for product_name (rice), price (-1000), category (food), sub category (healty), brand_name (good), description (this is best food in towr), and stock (12). A validation error message, 'Value must be greater than or equal to 0.', is shown in an orange box next to the price input field. At the bottom right of the form is a 'save and continue' button.

Figure 57 unit testing 10

4.2.11 Check if one seller can access the shop information of another seller through URL

Objective	To Check if seller can put product price negative
Action	<ul style="list-style-type: none"> Enter the other seller shop URL to access the information of that shop.
Excepted Result	403 forbidden message should be displayed
Actual Result	403 forbidden message is displayed
Conclusion	Test Successful

Table 14 unit test 11



403 Forbidden

Figure 58 unit testing 11

4.2.12 Check if customer can access to seller dashboard through URL

Objective	To Check if customer can access to seller dashboard through URL
Action	<ul style="list-style-type: none"> Enter the other seller dashboard URL to access the information of that seller.
Excepted Result	403 forbidden message should be displayed
Actual Result	403 forbidden message is displayed
Conclusion	Test Successful

Table 15 unit test 12



403 Forbidden

Figure 59 unit testing 12

4.2.13 Check if user can add items to cart without logged in

Objective	To Check if user can add items to cart without logged in
Action	<ul style="list-style-type: none"> Click on Add to cart on product you like
Excepted Result	Redirect to login page with next redirect add to cart
Actual Result	Redirected to login page with next redirect url add that product to cart
Conclusion	Test Successful

Table 16 unit test 13

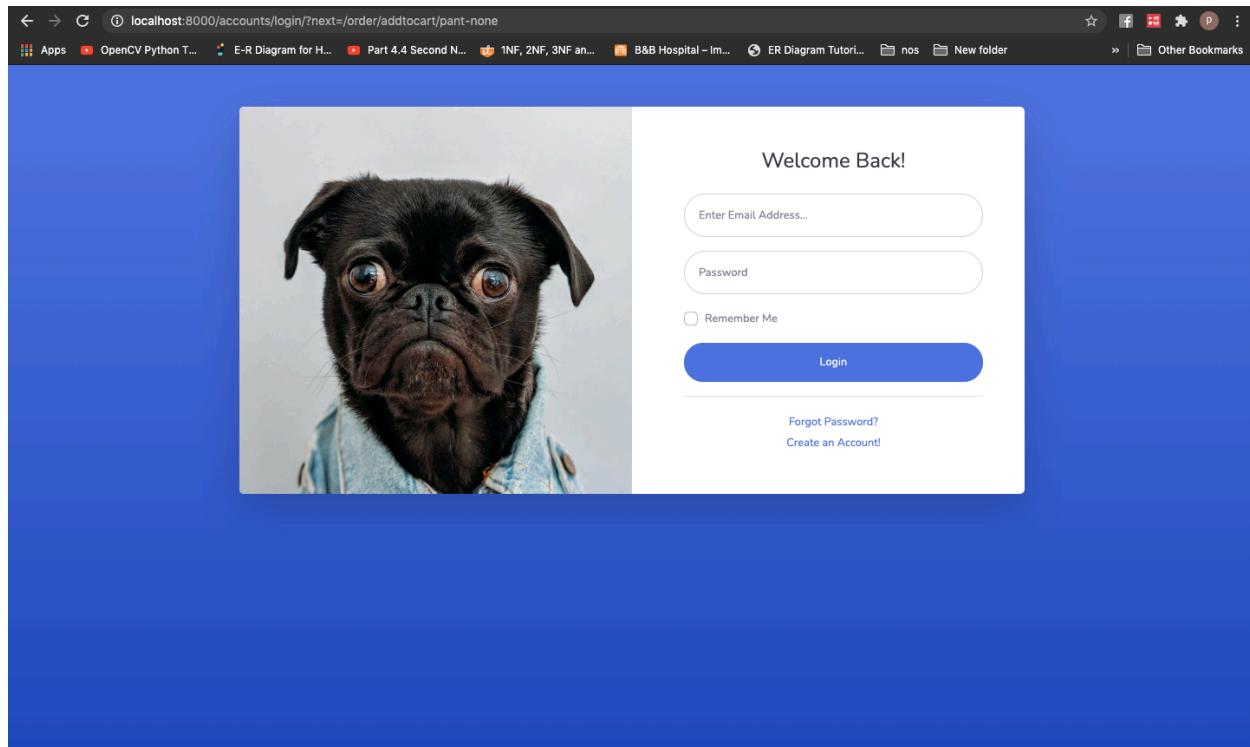


Figure 60 unit testing 13

4.2.14 Check if user can register as seller

Objective	To Check if user can register as seller
Action	<ul style="list-style-type: none"> Fill the registration form with valid information Click on signup
Excepted Result	Seller account should be created and save in database successfully.
Actual Result	Seller account created and saved in database successfully
Conclusion	Test Successful

Table 17 unit test 14

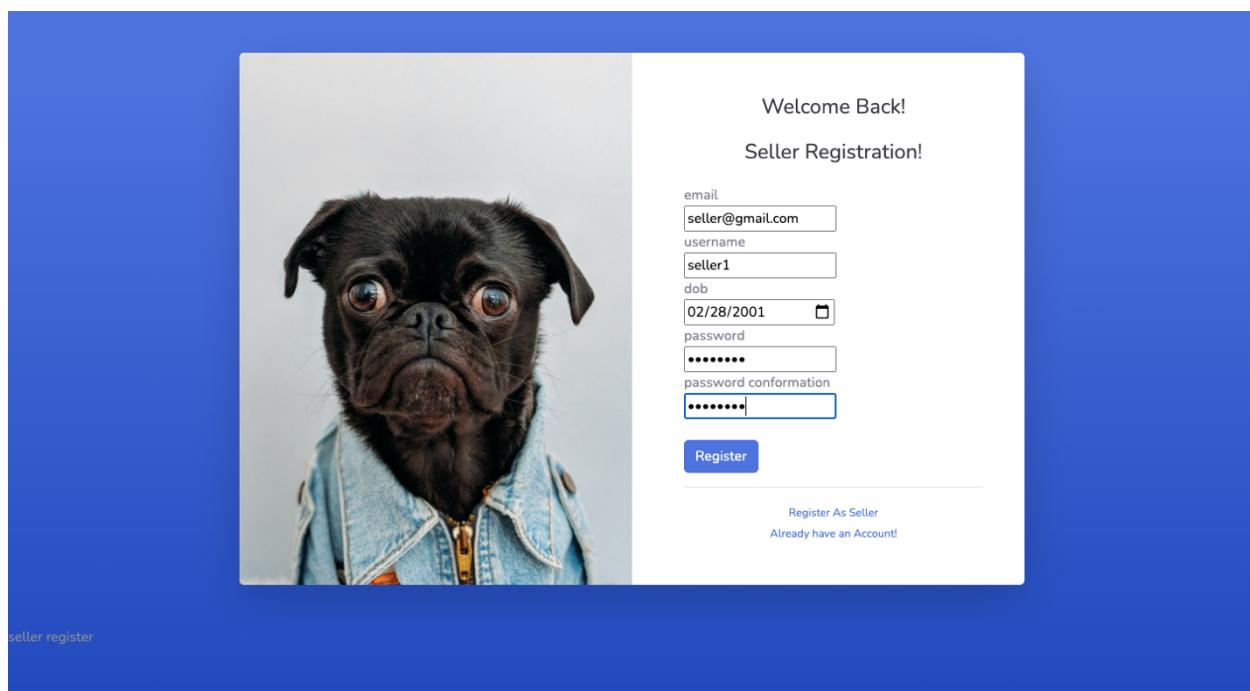


Figure 61 unit testing 14

User created successfully

first_name:

last_name:

mobile_number:

citizenship_number:

citizenship_front: Choose File | No file chosen

citizenship_back: Choose File | No file chosen

temp_add:

per_add:

[Create Profile](#)

Copyright © Your Website 2020

Figure 62 unit testing 14(ii)

Id	last_login	email	username	date_of_birth	is_email_verified	is_profile_verified	is_customer	is_seller	is_active	is_admin	pass
3	2021-04-22 14:36:37.748522	seller@gmail.com	seller1	2001-02-28	0	0	0	1	1	0	pbk

Figure 63 unit testing 14(iii)

4.2.15 Check if user password is encrypted after successful registration of user

Objective	To Check if user password is encrypted after successful registration of user
Action	<ul style="list-style-type: none"> Fill the registration form with valid information Click on signup
Excepted Result	password should be encrypted successfully and saved in database
Actual Result	Password encrypted successfully and saved in database
Conclusion	Test Successful

Table 18unit test 15

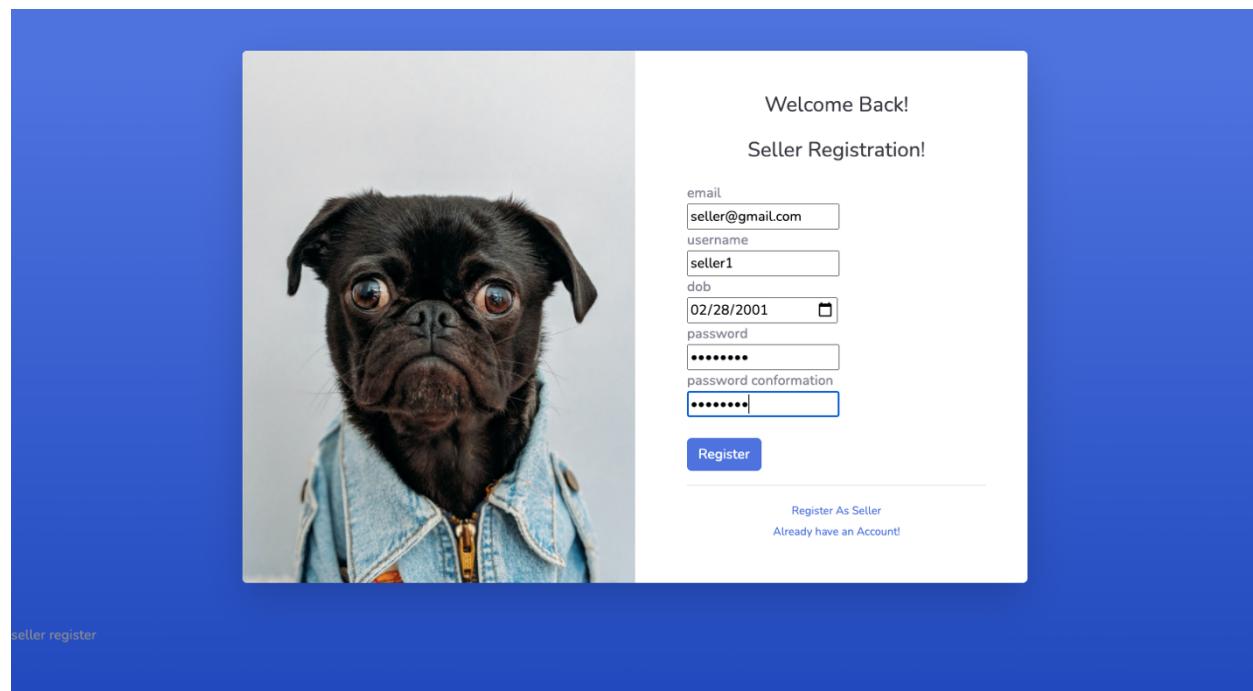


Figure 64 unit testing 15

				id	password
<input type="checkbox"/>	Edit	Copy	Delete	1	pbkdf2_sha256\$216000\$Q0Z7sYrrrPjx\$swHATAtUreX/xyTg...
<input type="checkbox"/>	Edit	Copy	Delete	2	pbkdf2_sha256\$216000\$TMSk2PZY7Glg\$/3Dsfw+4OC3q3Du4...
<input type="checkbox"/>	Edit	Copy	Delete	3	pbkdf2_sha256\$216000\$8HtIqraPos1\$D9ccDXFr8bME0ule...

Figure 65 unit testing 15(ii)

4.2.16 Check if logged in seller can add shop

Objective	To Check if logged in seller can add shop
Action	<ul style="list-style-type: none"> Fill the form with validate information Click on Add new shop
Excepted Result	Shop should be Added successfully
Actual Result	Shop Added successfully
Conclusion	Test Successful

Table 19 unit test 16

Shop registered name:
Ranko
shop_name:
Ranko bajar
shop_user_name:
ranko
shop_owner_name:
Pujan
shop_address:
pepsicola ktm
shop_warehouse_address:
pepsicola ktm
shop_returned_address:
pepsicola ktm
shop_category:
wholesale
shop_detail_description:
my shop is very beautifi
shop_pan_card:
Choose File 176734141_4...03809_n.jpeg
shop_phone_number:
9808282207
shop_owner_phone_number:
9808282208

submit

Copyright © Your Website 2020

Figure 66 unit testing 16

The screenshot shows the SAMANCHA application interface. On the left, there is a sidebar with a blue background containing the following menu items:

- Dashboard
- INTERFACE
 - My Profile
 - Shop Profile
 - Messages

At the top right, there is a search bar with placeholder text "Search for...", a magnifying glass icon, and a user profile icon with the email "seller123@gmail.com". A green button labeled "Add Shop" is located in the top right corner of the main content area.

The main content area has a title "Shop Profile" and contains a table with the following data:

Shop Name	Shop User Name	Shop Category	Shop Verification	Action
Ranko bajar	ranko	wholshale	False	view Shop Details
Shop Name	Shop User Name	Shop Category	Shop Verification	Action

Below the table, it says "Showing 1 to 1 of 1 entries". At the bottom right of the content area, there are buttons for "Previous", "1", and "Next".

Copyright © Your Website 2020

Figure 67 unit testing 16(ii)

4.2.17 Check if logged in seller can add product to shop

Objective	To Check if logged in seller can add product to shop
Action	<ul style="list-style-type: none"> Fill the form with validate information Click on Add Product
Excepted Result	Product should be Added successfully in shop
Actual Result	Product Added successfully
Conclusion	Test Successful

Table 20 unit test 17

Copyright © Your Website 2020

Figure 69 unit testing 17

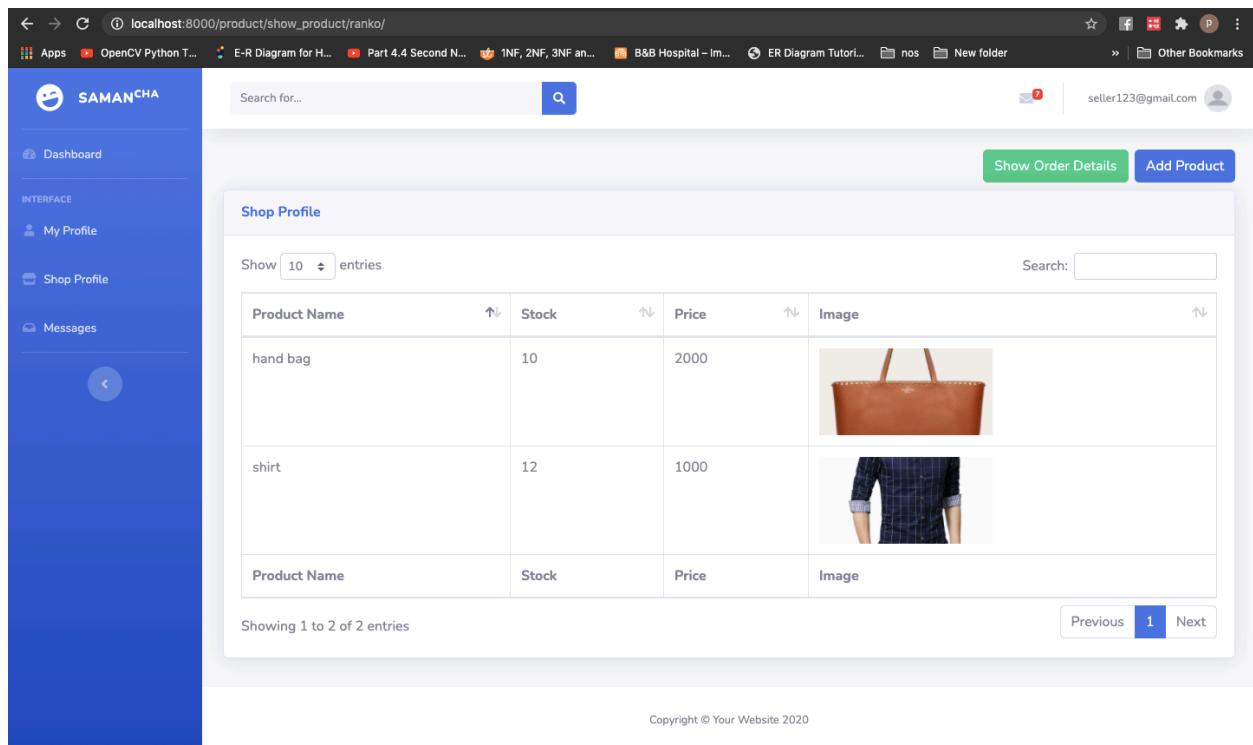
ID	slug	product_name	price	photo	category	sub_category	brand_name	description	stock	owner_id	shop_product_id	created_date
1	hand-bag-none	hand bag	2000	Screen_Shot_2021-04-23_at_5.04.16_PM.png	handbag	bag	godrez	this bag is really very good bag	10	4	1	2021-04-23 11:20:15.38208

Figure 68 unit testing 17(ii)

4.2.18 Check if logged in seller can view all the product of their shop

Objective	To Check if logged in seller can view all the product of their shop
Action	<ul style="list-style-type: none"> Click on View shop details
Excepted Result	Product should be displayed successfully
Actual Result	Product displayed successfully
Conclusion	Test Successful

Table 21 unit test 18



The screenshot shows a web browser window with the URL `localhost:8000/product/show_product/rank0/`. The page is titled "Shop Profile". On the left, there is a sidebar with a logo and links for "Dashboard", "My Profile", "Shop Profile", and "Messages". The main content area displays a table with two rows of products. Each row contains the Product Name, Stock, Price, and a thumbnail image.

Product Name	Stock	Price	Image
hand bag	10	2000	
shirt	12	1000	

At the bottom of the table, there is a footer with the text "Showing 1 to 2 of 2 entries" and navigation buttons for "Previous", "1", and "Next".

Figure 70 unit testing 18

4.2.19 Check if the added product is displayed in their particular shop landing page

Objective	To Check if the added product is displayed in their particular shop landing page
Action	<ul style="list-style-type: none"> • Go the URL of shop you want to visit
Excepted Result	Product should be displayed at that particular shop successfully
Actual Result	Product displayed at that particular shop successfully
Conclusion	Test Successful

Table 22 unit test 19

The screenshot shows a web-based shopping interface. At the top, there's a header with a logo consisting of three colorful bags (orange, green, blue) and the word "Samantha". To the right of the logo are buttons for "All Product" (with a dropdown arrow), "Search" (with a magnifying glass icon), and "Shop Profile" / "Logout". On the far right of the header is a "View cart" button.

Below the header, the main content area is titled "Products". It displays two items:

- hand bag** - RS.2000 - **Add to card** button
- shirt** - RS.1000 - **Add to card** button

At the bottom right of the products section, there is a "Send Message" button.

Figure 71 unit testing 19

4.2.20 Check if logged in user can remove product from the shop

Objective	To Check if logged in user can remove product from the shop
Action	<ul style="list-style-type: none"> Click on Delete button to delete the item
Excepted Result	Product Should be deleted and only not deleted item will be displayed
Actual Result	Product deleted and only not deleted item will be displayed
Conclusion	Test Successful

Table 23 unit test 20

Shop Profile					
Product Name				Stock	Price
Image				Image	
hand bag	10	2000		Delete	
shirt	12	1000		Delete	
Product Name		Stock	Price	Image	
				Image	

Showing 1 to 2 of 2 entries

Previous 1 Next

Figure 72 unit testing 20(i)

Shop Profile					
Product Name				Stock	Price
Image				Image	
shirt	12	1000		Delete	
Product Name		Stock	Price	Image	
				Image	

Showing 1 to 2 of 2 entries

Previous 1 Next

Figure 73 unit testing 20(ii)

4.2.21 Check if logged in user can add product to cart

Objective	To Check if logged in user can add product to cart
Action	<ul style="list-style-type: none"> Click on Add to cart on the product which you want to add to cart
Excepted Result	Product should be Added to cart
Actual Result	Product Added to cart successfully
Conclusion	Test Successful

Table 24 unit test 21

The screenshot shows a user interface for a shopping cart. At the top, there is a search bar with a magnifying glass icon and a user profile area showing 'seller123@gmail.com' and a profile picture. Below the header, a green button labeled 'Add More item' is visible.

The main content area is titled 'My Cart'. It includes a table with columns: Product Name, Product Photo, Product Price, and Action. A dropdown menu shows 'Show 10 entries'. A search bar is also present.

Product Name	Product Photo	Product Price	Action
hand bag		2000	<button>Remove From Cart</button>
Product Name	Product Photo	Product Price	Action

At the bottom, it says 'Showing 1 to 1 of 1 entries'. There are navigation buttons for 'Previous' (disabled), '1', and 'Next'. To the right, there is a large button labeled 'Total:2000' and a prominent 'Check Out' button.

Figure 74 unit testing 21

4.2.22 check if logged in user can remove the product from the cart

Objective	To check if logged in user can remove the product from the cart
Action	<ul style="list-style-type: none"> Click on remove from cart to remove item from
Excepted Result	Product Should be removed successfully from cart
Actual Result	Product removed from cart successfully
Conclusion	Test Successful

Table 25 unit test 22

The screenshot shows a user interface for managing a shopping cart. At the top, there is a search bar with a magnifying glass icon and a user profile area showing 'seller123@gmail.com' and a small profile picture. Below the header is a green button labeled 'Add More item'.

The main content area is titled 'My Cart'. It contains a table with the following data:

Product Name	Product Photo	Product Price	Action
hand bag		2000	<button>Remove From Cart</button>

Below the table, there is a message 'Showing 1 to 1 of 1 entries' and a navigation bar with 'Previous' and 'Next' buttons, currently showing page '1'. To the right of the table, there is a summary box containing 'Total:2000' and a large 'Check Out' button.

Figure 75 unit testing 22

My Cart					
Show 10 entries		Search:			
Product Name	Product Photo	Product Price	Action		
No data available in table					
Product Name	Product Photo	Product Price	Action		

Showing 0 to 0 of 0 entries

Previous Next

No Item Added To Cart

Figure 76 unit testing 22(ii)

4.2.23 Check if logged in user can checkout themselves after adding items to cart.

Objective	To Check if logged in user can checkout themselves
Action	<ul style="list-style-type: none"> • Add items to cart • Click on View cart • Click on checkout • Fill the payment details • Click on Pay
Excepted Result	Your order should be successfully placed
Actual Result	Ordered Placed successfully
Conclusion	Test Successful

Table 26 unit test 23

The screenshot shows a web-based shopping cart system. At the top, there is a search bar with a magnifying glass icon and a user profile area showing 'seller123@gmail.com'. Below the header is a green button labeled 'Add More item'.

The main content area is titled 'My Cart' and displays a table of items. The table has columns for Product Name, Product Photo, Product Price, and Action. One item is listed: 'shirt' with a price of '1000'. To the right of the item's price is a 'Remove From Cart' button. Below the table, it says 'Showing 1 to 1 of 1 entries'. At the bottom right of the cart area, it shows 'Total:1000' and a large 'Check Out' button.

At the very bottom of the page, there is a small copyright notice: 'Copyright © Your Website 2020'.

Figure 77 unit testing 23

Transaction Details	
EPAYTEST	NPR
Product Amount:	3,000.00
Tax Amount:	0.00
Service Charge:	0.00
Delivery Charge:	0.00
Total Amount:	3,000.00

User Info Confirmation	
Full Name	eSewa esewa esewa
eSewa ID:	9806800001
Contact:	9806800001
Address:	KTM, Baseri-1, Dhading, Bagmati
Email:	pujanr@gmail.com
CONFIRM	BACK

Figure 78 unit testing 23(ii)

My Order Details					
Show 10 entries		Search: <input type="text"/>			
Product Name	Product Photo	Product Price	Status		
hand bag		2000	In review		
shirt		1000	In review		
Product Name	Product Photo	Product Price	Status		

Showing 1 to 2 of 2 entries

Previous **1** Next

Figure 79 unit testing 23(iii)

4.2.24 Check if user can search product by their name

Objective	To Check if logged in user can checkout themselves
Action	<ul style="list-style-type: none"> Type the product name you want to search in search box Click on search Click on checkout
Excepted Result	Searched product should be displayed in screen
Actual Result	Searched product displayed in screen
Conclusion	Test Successful

Table 27 unit test 24



shirt
Rs. 1000
gucci
[View Details](#)

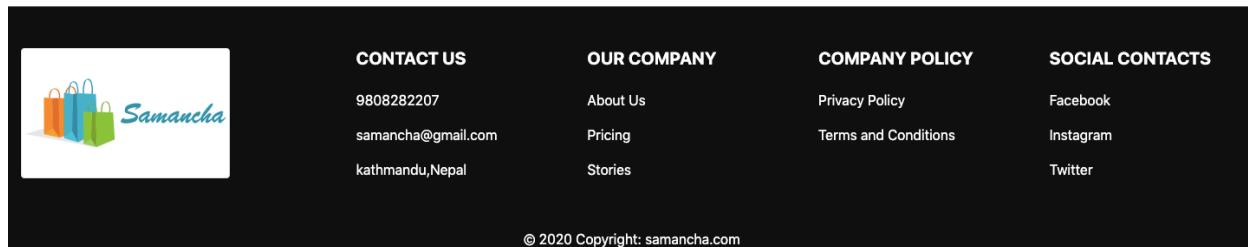


Figure 80 unit testing 24

4.2.25 Check if logged in customer can send message to seller

Objective	To Check if logged in customer can send message to seller
Action	<ul style="list-style-type: none"> • Go to shop landing page • Click on send message • Type message you want to send • Click on send
Excepted Result	Message should be sent successfully
Actual Result	Message sent successfully
Conclusion	Test Successful

Table 28 unit test 25

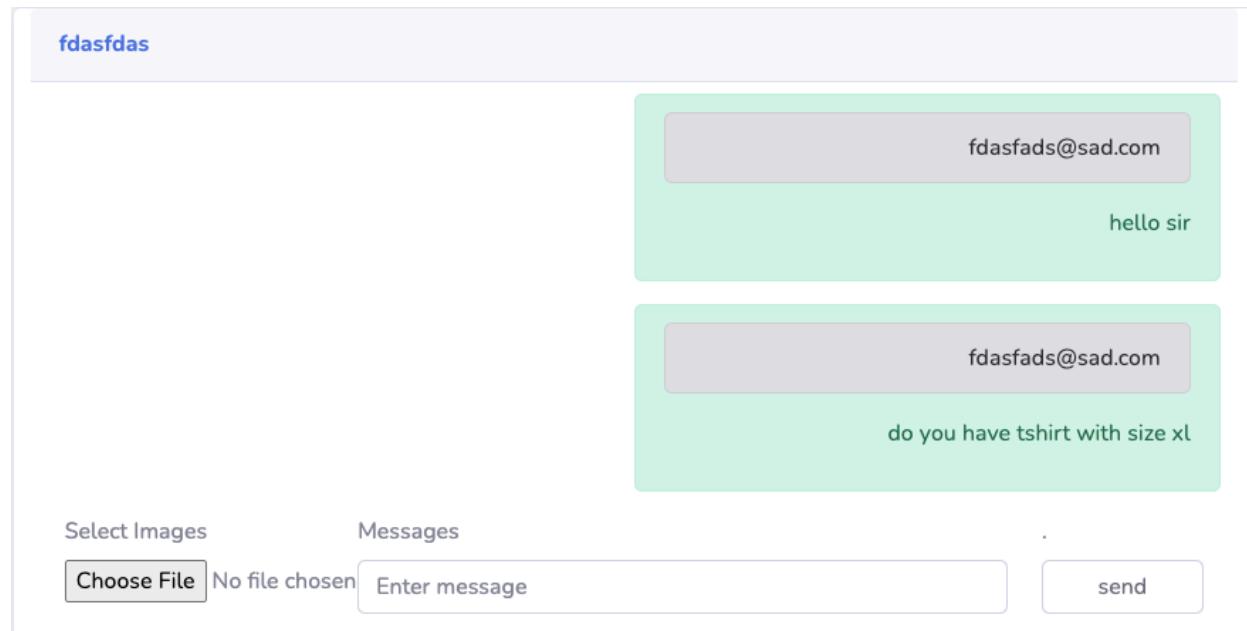


Figure 81 unit testing 25

4.2.26 Check if logged in customer can see message send by seller

Objective	To Check if logged in customer can send message to seller
Action	<ul style="list-style-type: none"> • Go to shop landing page • Click on messages
Excepted Result	Message send by shop should be displayed
Actual Result	Message sent by shop is successfully displayed
Conclusion	Test Successful

Table 29 unit test 26

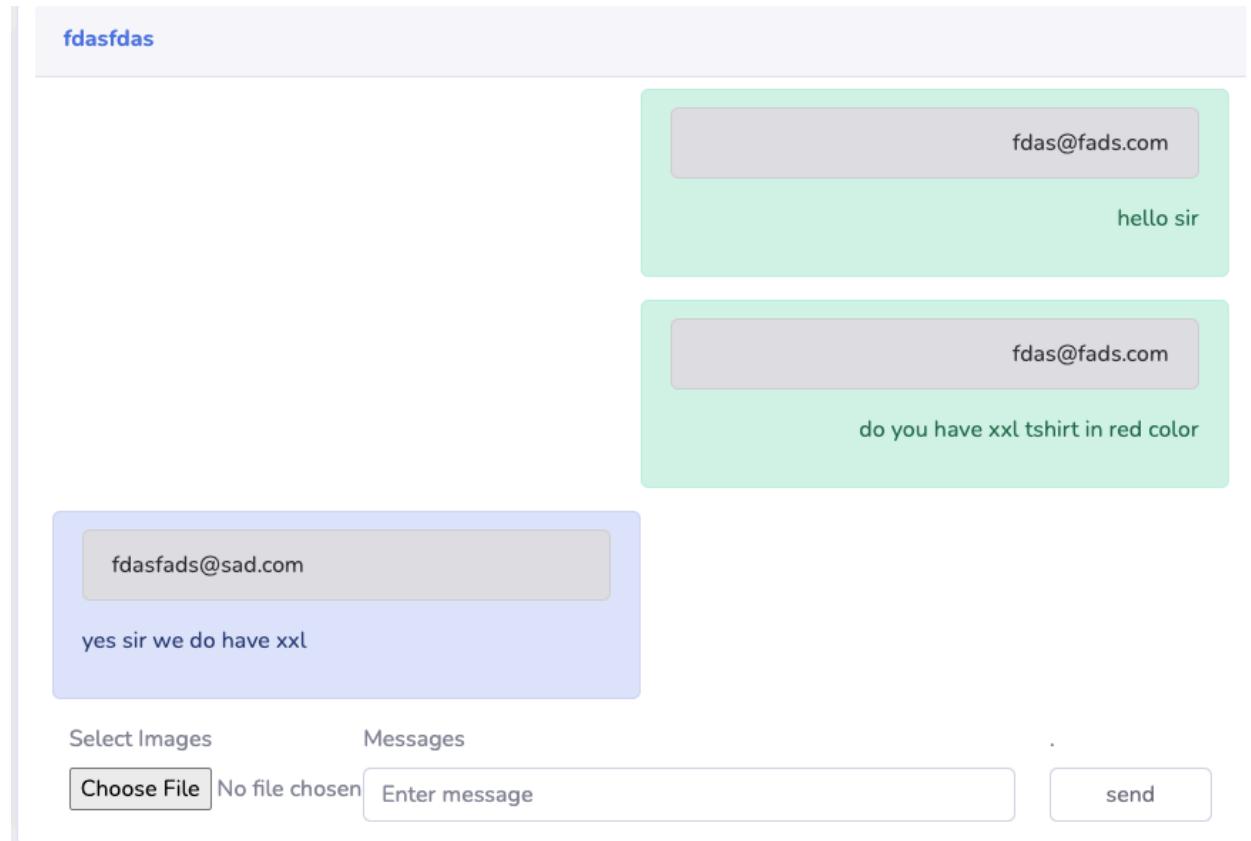


Figure 82 unit testing 26

4.3 System Testing

4.3.1 Check if user can create seller profile

Objective	To create seller profile
Action	<ul style="list-style-type: none"> • Go to register as seller page • Fill form with valid data • Click on register • Fill all of your details when create profile page appears • Click on submit
Excepted Result	Seller profile should be created
Actual Result	Seller profile created
Conclusion	Test Successful

Table 30 System testing 1

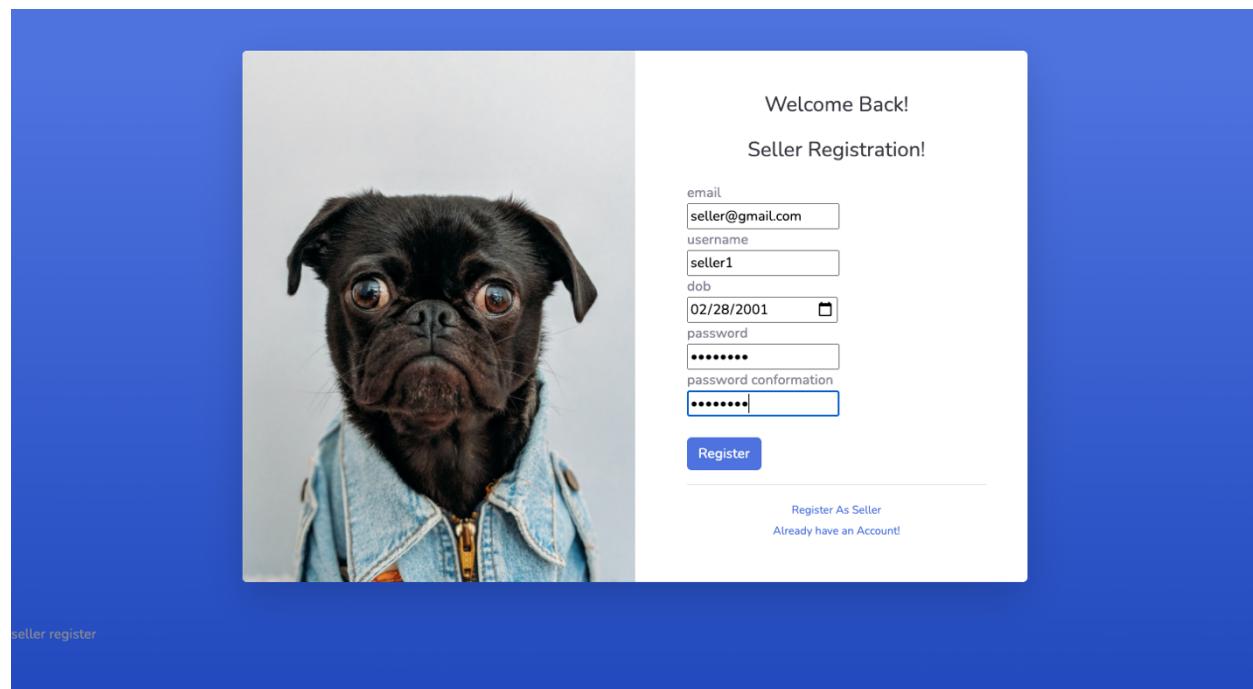


Figure 83 system tesing 1(i)

User created successfully

first_name:

last_name:

mobile_number:

citizenship_number:

citizenship_front:
 Screen Shot 2...t 11.34.04 AM

citizenship_back:
 Screen Shot 2...at 12.40.51 PM

temp_add:

per_add:

Figure 84 system tesing 1(ii)

2021-04-25 12:15:31.059820	aa1@aa.com	aaaaa12	2021-04-06	0	0	0	1	1	0
----------------------------	------------	---------	------------	---	---	---	---	---	---

Figure 85 system tesing 1(iii)

4.3.2 Check if user can buy place an order

Objective	To check if user can place an order
Action	<ul style="list-style-type: none"> • Go to login page • Enter valid credential for login • Click on login • Click add product to add items to cart • Click on view cart • Click on checkout • Fill your esewa login credential when esewa login page appears • Click on buy product
Expected Result	User should be able to place an order
Actual Result	User was able to place an order
Conclusion	Test Successful

Table 31 system tesing 2

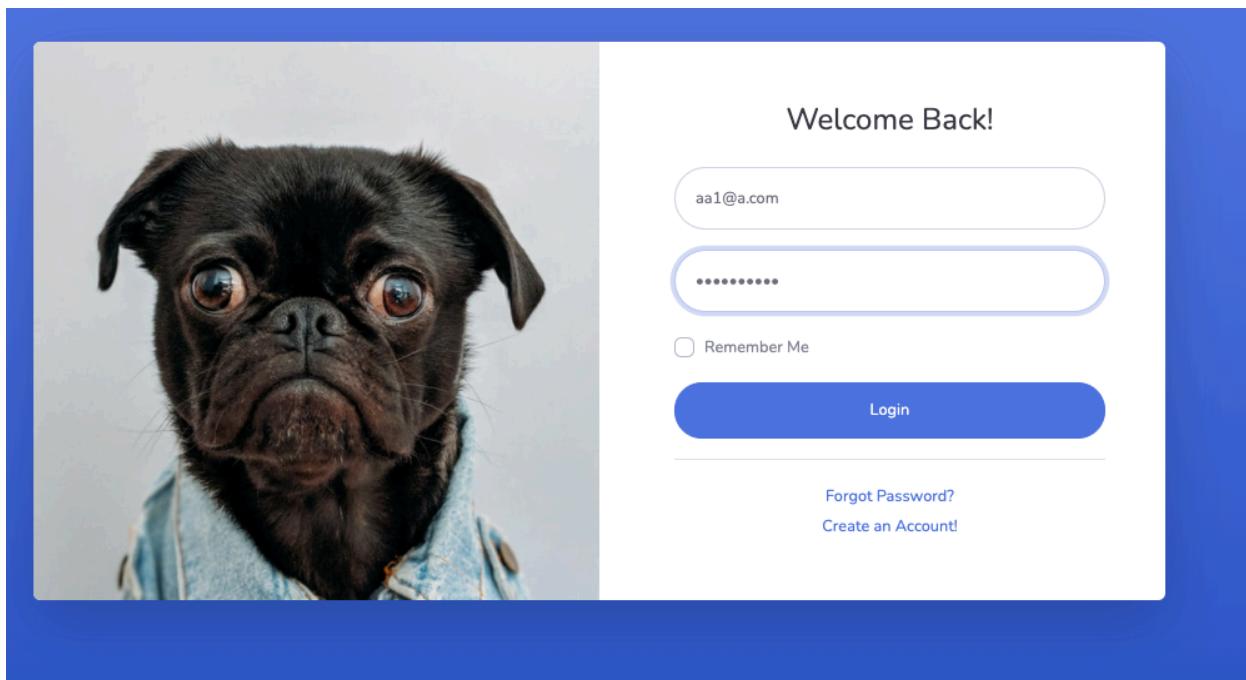


Figure 86 system tesing 2(i)

The screenshot shows the products page of the Samancha website. At the top, there is a navigation bar with a logo, a dropdown menu for 'All Product', a search bar, and buttons for 'Search', 'Shop Profile', 'Logout', and 'View cart'. Below the navigation bar, the title 'Products' is centered. There are four product cards displayed in a row:

- tshirtssss**
RS.2000
[Add to card](#)
- tshirt**
RS.1000
[Add to card](#)
- shirt**
RS.1000
[Add to card](#)
- hand bag**
RS.2000
[Add to card](#)

At the bottom of the page, there is a footer section with the Samancha logo, contact information (9808282207, samancha@gmail.com), company details (About Us, Pricing), policy links (Privacy Policy, Terms and Conditions), and social media links (Facebook, Instagram).

Figure 87 system tesing 2(ii)

The screenshot shows the 'My Cart' page. At the top, there is a header with the Samancha logo, a search bar, and a button for 'Add More item'. Below the header, the table has the following data:

Product Name	Product Photo	Product Price	Action
shirt		1000	Remove From Cart
tshirt		1000	Remove From Cart

At the bottom of the table, there is a summary row:

Product Name	Product Photo	Product Price	Action
--------------	---------------	---------------	--------

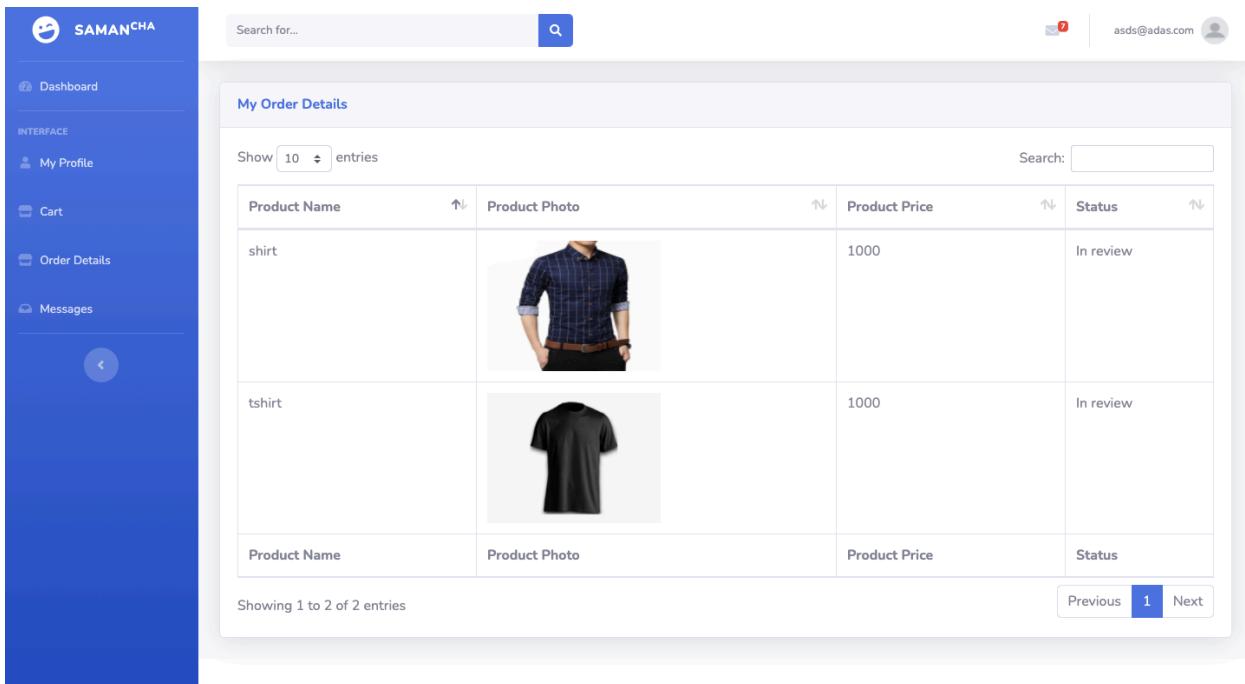
Below the table, there is a message 'Showing 1 to 2 of 2 entries' and a navigation bar with 'Previous', '1', and 'Next'. To the right of the table, there is a large button labeled 'Total:2000' and 'Check Out'.

Figure 88 system tesing 2(iii)

Transaction Details	
EPAYTEST	NPR
Product Amount:	3,000.00
Tax Amount:	0.00
Service Charge:	0.00
Delivery Charge:	0.00
Total Amount:	3,000.00

User Info Confirmation	
Full Name	eSewa esewa esewa
eSewa ID:	9806800001
Contact:	9806800001
Address:	KTM, Baseri-1, Dhading, Bagmati
Email:	pujanr@gmail.com
CONFIRM	BACK

Figure 89 system tesing 2(iv)



The screenshot shows the SAMANCHA user interface. On the left, a sidebar menu includes Dashboard, My Profile, Cart, Order Details, and Messages. The main content area is titled "My Order Details". It features a search bar at the top right. Below the search bar, there are filters for "Show 10 entries" and a "Search:" field. A table lists two items: a shirt and a tshirt. Each item has a "Product Name" (e.g., shirt), a "Product Photo" (e.g., a blue and white checkered shirt or a black t-shirt), a "Product Price" (1000), and a "Status" (In review). At the bottom of the table, it says "Showing 1 to 2 of 2 entries" and includes navigation buttons for "Previous" and "Next".

Figure 90 system tesing 2(v)

4.3.3 Check seller can perform CRUD operation on shop product

Objective	To Check seller can perform CRUD operation on shop product
Action	<ul style="list-style-type: none"> • Go to login page • Enter the valid credential of seller • Click on shop on dashboard • Click on the shop view details where you want add product • Click on add product • Fill product details • Click on save product • Click on view product • Click on edit button of the product • Update the information of the product you want update • Click on update button • Click on delete product when user is redirect to view product page
Excepted Result	Seller should be able to perform crud operation on shop product
Actual Result	Seller was able to perform crud operation on shop product
Conclusion	Test Successful

Table 32 system testing 3

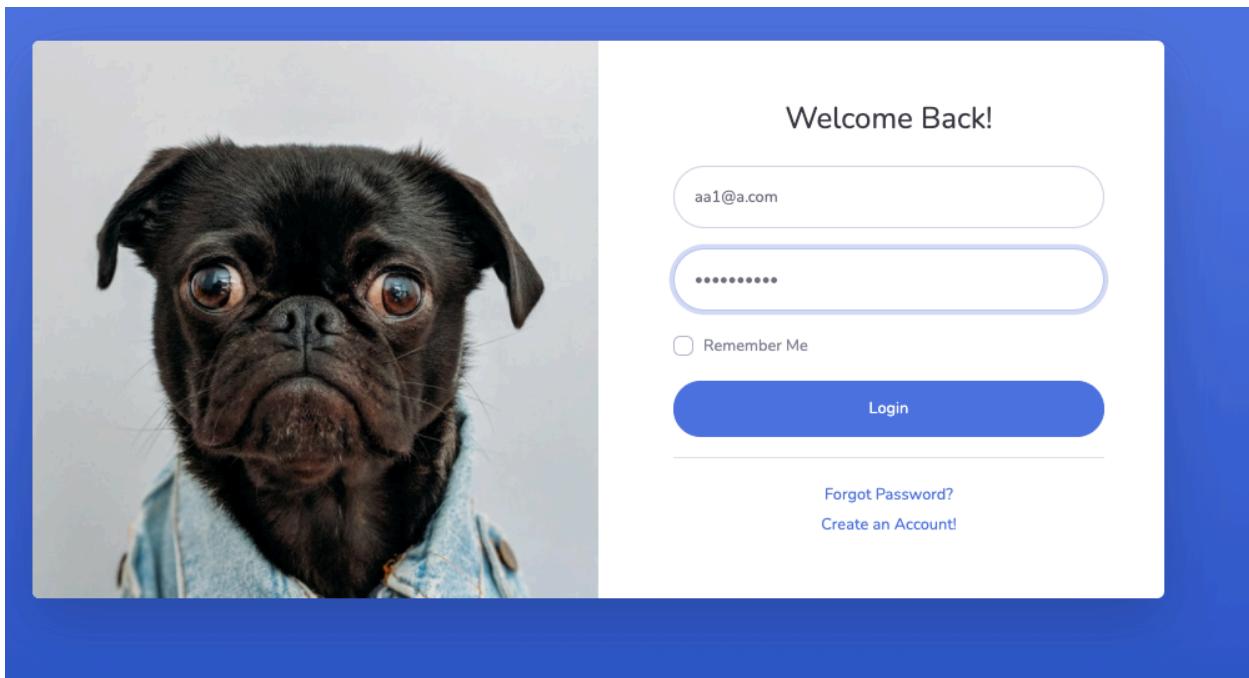


Figure 91 system tesing 3(i)

The image is a screenshot of a web browser showing multiple tabs open. The active tab displays a shop profile table titled "Shop Profile". The table has columns for Shop Name, Shop User Name, Shop Category, Shop Verification, and Action. One entry is visible: "kapda pasal" under Shop Name, "kapdapasal" under Shop User Name, "clothes" under Shop Category, and "False" under Shop Verification. The "Action" column contains a link labeled "view Shop Details". The sidebar on the left shows navigation links for Dashboard, My Profile, Shop Profile, and Messages. The top of the page includes a search bar and a user account section.

Shop Name	Shop User Name	Shop Category	Shop Verification	Action
kapda pasal	kapdapasal	clothes	False	view Shop Details
Shop Name	Shop User Name	Shop Category	Shop Verification	Action

Figure 92 system tesing 3(ii)

The screenshot shows a web-based application interface. On the left, there is a vertical sidebar with a dark blue background. It contains icons for Instagram, Paraphrasing Tool, Company Data, Company Annex, SB Admin 2 - Tables, and another Paraphrasing Tool tab. Below these are sections for 'INTERFACE' with 'My Profile', 'Shop Profile' (which is currently selected), and 'Messages'. At the top right, there is a user profile icon and an email address 'aa1@aa.com'. The main content area has a light gray header with the title 'Shop Profile'. Below it is a search bar with placeholder text 'Search for...' and a magnifying glass icon. To the right of the search bar are two buttons: a green one labeled 'Show Order Details' and a blue one labeled 'Add Product'. The main content area contains a table with columns: 'Product Name', 'Stock', 'Price', 'Image', and 'Image'. A message 'No data available in table' is displayed above the table. At the bottom of the content area, it says 'Showing 0 to 0 of 0 entries' and features 'Previous' and 'Next' navigation buttons. In the bottom right corner of the main content area, there is a small thumbnail image of a laptop screen displaying the same application.

Figure 93 system tesing 3(iii)

The screenshot shows a 'Create Product' form. The form fields are as follows:

- product_name: tshirt
- price: 1000
- photo: Choose File t2.jpg
- category: top wear
- sub category: cotton
- brand_name: ktm city
- description: this is good clothes
- stock: 11

At the bottom of the form is a 'save and continue' button. A green 'Show Product' button is located at the top right of the form area.

Figure 94 system tesing 3(iv)

						Show Order Details	Add Product
Shop Profile							
Show 10 entries						Search: <input type="text"/>	
Product Name	Stock	Price	Image			Image	
tshirt	11	1000				Delete	
Product Name	Stock	Price	Image			Image	

Showing 1 to 1 of 1 entries

Previous 1 Next

Copyright © Your Website 2020

Figure 95 system tesing 3(v)

						Show Product
Create Product						
product_name:	<input type="text" value="tshirt"/>					
price:	<input type="text" value="1000"/>					
photo:	<input type="button" value="Choose File"/> t2.jpg					
category:	<input type="text" value="top wear"/>					
sub category:	<input type="text" value="cotton"/>					
brand_name:	<input type="text" value="ktm city"/>					
description:	<input type="text" value="this is good clothes"/>					
stock:	<input type="text" value="11"/>					
<input type="button" value="save and continue"/>						

Figure 96 system tesing 3(vi)

Shop Profile						
Product Name				Stock	Price	Image
tshirtssss				11	2000	
						<button>Delete</button>
Product Name						
				Stock	Price	Image
						<button>Image</button>

Showing 1 to 2 of 2 entries

Previous 1 Next

Figure 97 system tesing 3(vii)

Shop Profile						
Product Name				Stock	Price	Image
NO PRODUCT TO SHOW						
						<button>Image</button>
						<button>Image</button>

Showing 1 to 2 of 2 entries

Previous 1 Next

Figure 98 system tesing 3(viii)

4.4 CRITICAL ANALYSIS

After the complete development of the system, we can clearly see the difference in the system from the described proposal. Though the project developed is exact to the proposed system but the core concept of the system which is to develop multi-vendor ecommerce site is exactly the same. Throughout the development process a lot of new features had been added while some of the idea had been left out due to its complexity, practicality, feasibility or inaccessibility. The system developed is dynamic, Realtime and very user-friendly to both customer and seller.

When looking back during the initial phase of the development, a lot of wrong assumptions were made. But as time passes by doing research and taking help of my supervisor all the mis concept were cleared and wrong assumption ideas were drop out. At first, I was thinking of making APIs and some of the development progress were also done on that but due to the lack of knowledge, proper guidance and lack of time, it was very difficult for me caught up so I had decided develop web app using Django framework only.

After the completion of project, the most difficult part was to make it real time. At first, I had used ajax call in each 10 MS to refresh the web page and it worked but it was my naive mistake of mine. Since making ajax call after each 10 MS made my system unstable, slow and it also consumed my huge resource due to the unwanted ajax call. Not only implementing this method made my project slow, it also cost a lot of time. After finding making ajax call in each 10 MS was not the ideal way; I did a lot of research about how to make my system real time and stable. After research I decided to use Web socket and Django built in ASGI Channel to make my website real time.

While the Project is totally in working condition, the GUI angle is as yet sketchy. It was intended to be natural and better in view of client experience, but still there actually have serious room for improvement.

At the end project includes all the major functionalities that should be in any multi-vendor ecommerce site.

CHAPTER 5: CONCLUSION

5.1 LEGAL, SOCIAL AND ETHICAL ISSUES

Developing a multi-vendor ecommerce site can be game changer how people do shop or how people sell their products. But only developing a system is not enough. After the developing system, we should think of all kind of issues which we may face in future should be identified and tackled. So, some of the probable legal, social and ethical issues are described below.

5.1.1 LEGAL ISSUES

The probable legal issues that my site may face are listed below:

GUI design:

In some parts I have integrated some design which are open sourced in internet. If the copied designed doesn't belong to the person who had uploaded, I might face the copyright strike by the owner. But to avoid that I have clearly given credit to the owner and also asked to mail if they want credit or removal for their contain.

Admin Deleting product:

Since admin have right to delete and update the shop product. If admin accidentally deleted or updated product without any legitimate reason which can cost loss to seller sale. Seller can sue case on us for not letting them sell authentic product by promising the features to sell product. To avoid this admin is trained not to do any unnecessary action. And sellers also are also requested to send email if something like this happen

Data privacy

Since we are collecting the user personal data. This can be legal issue on us by our customer for collecting their personal data. But to avoid this kind of legal issue we have clearly included we can collect user data in our privacy policy.

Domain Name:

It is necessary to choose and register a domain name before launching the website so that users can access it through a web browser. We must be careful not to indulge in cybersquatting when choosing a name. It is the act of registering a domain name that is confusingly identical to another company's trademark. This may result in trademark and copyright problems. Cybersquatting will result in the loss of a domain name.

Government Policies and Regulation

There are no definite Government rules and legislative mandates regulating the company and its operations, as there are for other e-commerce sites. The Nepalese government has adopted a reactive policy design framework, resulting in an uncertain and risky market climate for all online businesses, including multi-vendor ecommerce. As a result, this company will be exposed to potentially harmful government policies in the future. As a result, as owners and developers of a multi-vendor ecommerce system, we must exercise caution in operating the business and stay current on government rules and regulations.

5.1.2 SOCIAL ISSUES

Social issues associated with the program and the subsequent business ideas are as follows:

- Security and Privacy

For all businesses, particularly e-commerce, ensuring the protection and privacy of customer data is as important as it is difficult. Fraudsters could send spam or launch an attack on the web host server, infecting the website with viruses and potentially gaining access to confidential information about customers.

As the cyber world has evolved and progressed, cybercrime has become more sophisticated and advanced. As a result, our company must take the requisite measures to ensure the protection and privacy of client data.

- Customer Experience and Loyalty

Customer loyalty is critical for achieving long-term business growth, even with the best-designed website. Given the increased competition in the industry, gaining new customers and retaining their loyalty will be difficult. Furthermore, as an online company, it would take more effort to convert visitors to customers because it is difficult to develop consumer confidence and loyalty because the clients and the business do not know each other and cannot use their senses to gauge credibility as they would in face-to-face transactions.

As a result, it will be critical for our company to improve customer experience at any customer touch point and during the process, from service details to payment, in order to gain customer trust.

5.1.3 ETHICAL ISSUES

The importance of business ethics and success is equal to that of financial success. Every aspect of our company must be ethically driven in the sense that we must take responsibility for our conduct, accountability for our deliveries and commitments, and liability for our actions' consequences. Certain ethical dilemmas, on the other hand, are inevitable, and here are a few that our business may face:

- Property Right infringement

It is really easy to copy content and be inspired by ideas on the internet. However, we must be cautious not to allow innovation to become plagiarized. When designing our website and providing services, we must ensure that only legitimate ideas are used, or that we give credit where credit is due in accordance with legal protocols.

- Governance

We must ensure good governance as a company in order to instill and communicate principles and beliefs with everyone we work with. We must also ensure that all parties to our business, including vendors and restaurant owners, are not working against the company's and its customers' best interests through legal and governance mechanisms.

- Public Safety and Welfare

Protecting public safety and protection by sharing and presenting facts and the true nature of our programs, rather than selling false products and services, is another ethical issue we may face. To do so, we must ensure that only genuine information is collected from our suppliers and partners, and that our data extraction process is conducted with caution. We must also make certain that correct information is posted and transmitted to our clients.

We must ensure that we apply sufficient protection to our system to ensure that confidential information is not leaked, as well as take adequate steps to minimize the effects of any attacks on our system, in addition to exercising caution in authenticating information.

5.2 5.2 ADVANTAGES

The major advantages of multi-vendor ecommerce are listed below:

- Selling product 24hr:

After seller upload their product online, people can buy their product any time from anywhere when the product is available. It helps shop owner to maximizing the sales of their shop with no limited working hours.

- Easy Order:

People can easily order anything they like. For ordering they just have to add all the product they like to cart and checkout after paying the amount of their order.

- Saving Cost and Manpower:

Seller can manage multiple shop with very low manpower which also resultant in costing less money

- More Opportunity to startup:

Startup don't have huge fund to open shop at main city area. Since our website provide start up to lunch their shop and after lunching their shop their product can be accessed by anywhere around the world.

- User friendly:

Since our website is responsible and very easy to use as it uses all kind of modern technologies to make it more interactive to user.

- Better quality product:

Our website provides where people can review the product. So, after watching review people can easily identify the quality of the product

5.3 LIMITATION

Some of the limitations of the application are as follows:

- To access this website, we need good internet connection. So, without the use of internet our website cannot be accessed.
- There is only one payment gateway option “ESEWA” other payment option is not supported by the website
- People might get scam as seller may not give exact same product shown in the product picture
- Since there is no user manual for customer they might get stuck in some part.
- If seller accidentally remove their product from the shop there is no way that they can undo that step. To add that product, they again have to fill all the add product form and submit.
- There is no option to block the user by seller so they can spam their message box.
- Users doesn't get notified about their order. They have to manually check the status of their order.

5.4 FUTURE WORK

The first phase of the development is completed but there are still room for improvement in a lot of place. Some of the further work which I will be completing are listed below:

- Integrating as many payment systems as possible to the system:
Since the online payment is getting popular in world as well as around the world. So, it is very necessary to integrate the as many payment systems as possible to increase the sales product.
- Adding Notification features to the system:
Our application lacks the notification features which make seller and buyer unaware about their order. So, in order to make out website more user friendly and interactive, adding notification is in my top priorities.
- Recommendation System:
Since this is an age of beginning of machine learning and we are collecting a lot of data of user. So, adding recommendation system will be very attractive to user and can be key features to increase our sale.
- Adding blog section on product:
Since we have review system on website and people can guess the quality of the product. But if we add blog section where expert can place the thoughts and review on the particular product.

CHAPTER 6: REFERENCES (colorlib, 2020)

Bibliography

- shopify. (2020). *Ecommerce Definition - What is Ecommerce*. From shopify: <https://www.shopify.com/encyclopedia/what-is-e-commerce>
- amazon. (2020). *About Amazon*. From Amazon: <https://www.aboutamazon.com/>
- Rashad Yazdanifard, M. T. (2014). *globaljournals*. From The Review of Alibaba's Online Business Marketing Strategies Which Navigate them to Present Success: https://globaljournals.org/GJMBR_Volume14/4-The-Review-of-Alibabas-Online-Business-Marketing.pdf
- crunchbase. (2020). *crunchbase*. From Daraz Nepal - Crunchbase Company Profile & Funding: <https://www.crunchbase.com/organization/daraz-nepal>
- Sharma, A. (2020, december 20). *New Business Age (P) Ltd*. From Sasto Deal | New Business Age | Leading English Monthly Business Magazine of Nepal: https://www.newbusinessage.com/magazine_articles/view/874
- CodeLabII Ben . (2016). *CodeLabII Ben Preedy 312409: Image*. From Preedy: https://codelab2.files.wordpress.com/2016/11/waterfall_model.jpg
- writepass. (2012). *writepass*. From Throwaway-Prototyping.jpg: <https://i2.wp.com/writepass.com/journal/wp-content/uploads/2012/12/Throwaway-Prototyping.jpg>
- rahmania, a. e. (2012, june 2). *RUP (Rational Unified Process)* . From theexplorationofmyworlds: <https://theexplorationofmyworlds.wordpress.com/author/angdee240884/>
- django. (2020). *django*. From django: <https://www.djangoproject.com/>
- DRF. (2020). *Home - Django REST framework*. From Django REST framework: <https://www.django-rest-framework.org/>
- stackoverflow. (2020). *stackoverflow*. From stackoverflow: <https://stackoverflow.com/>
- colorlib. (2020). From colorlib: colorlib.com

Siguencia, L., Marzano, & Gródek-Szostak,. (2018) Online Booking: The Case of Booking a Hotel in Ogrodzieniec using the Booking.com Platform. In *EMoNT 2018*. Vrnjačka Banja, Serbia , 2018.

Skedda. (2020) *Skedda* [Online]. Available from: <https://www.skedda.com/> [Accessed January February 2020].

TechnologyUK. (2019) *Development Methodologies* [Online]. Available from: <http://www.technologyuk.net/computing/software-development/systemsanalysis/methodologies.shtml> [Accessed 4 January 2020].

TechTerm. (2020) *RUP* [Online]. Available from: <https://techterms.com/definition/rup> [Accessed 30 May 2020].

TRYQA. (2013) *What is Prototype model- advantages, disadvantages and when to use it?* [Online]. Available from: <http://tryqa.com/what-is-prototype-model-advantages-disadvantages-and-when-to-use-it/> [Accessed 4 January 2020].

tutorialspoint. (2019) *SDLC - Waterfall Model* [Online]. Available from: https://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm [Accessed 28 December 2019].

w3School. (2020) *Python Introduction* [Online]. Available from: https://www.w3schools.com/python/python_intro.asp [Accessed 1 June 2020].

GoodFirms. (2020) *The Best 7 Free Venue Management Software Solutions* [Online]. Available from: https://www.goodfirms.co/blog/best-free-venue-management-software-solutions?utm_source=facebook&utm_medium=social&utm_campaign=social_media&fbclid=IwAR1WcrYnZnO0mx3vePYZHq2LfXOgeMpxD9c5wuEVJopMbrnmmyH-8_lzk8 [Accessed 2 January 2020].

Guru99. (2020) *What is System Testing? Types & Definition with Example* [Online]. Available from: <https://www.guru99.com/system-testing.html>

[testing.html#:~:text=SYSTEM%20TESTING%20is%20a%20level,a%20larger%20computer%2Dbased%20system.](#) [Accessed 4 June 2020].

Hasan, S. (2014) DOCUMENTATION OF ONLINE BOOKING SYSTEM., 2014. Higgins,. (2019) *Venue Ideas: Choosing the Best Event Location* [Online]. Available from:

<https://www.eventbrite.com/blog/venue-ideas-ds00/> [Accessed 19 April 2020]. Johnston,. (2020) *Why Use an Online Booking and Online Scheduling System?* [Online].

Available from: <https://www.bookinglive.com/blog/why-use-an-online-booking-system>. Kunze, C.A. (1998) Web Site Legal Issues. *Santa Clara High Technology Law Journal*,

14(2), p.28.

Li, J., Cao , , Meng , & Li, C. (2017) *Design and Implementation of Online Booking System of University Sports Venues*. Zhengzho, China: Zhengzhou Universit.

Lonely Planet. (2017) *Nepal in Detail* [Online]. Available from: <https://www.lonelyplanet.com/nepal/background/other-features/a/nar/a580b59b-e011-49f28147-ff6ef5d7f8c4/357105> [Accessed 17 November 2019].

Maja, M. (2019) *Rational Unified Process (RUP)* [Online]. Available from: <https://activecollab.com/blog/project-management/rational-unified-process-rup> [Accessed 27 December 2019].

Microsoft. (2020) *Documentation for Visual Studio Code* [Online]. Available from: <https://code.visualstudio.com/docs> [Accessed 30 May 2020].

Morse, P. (2016) *Waterfall Model: What Is It and When Should You Use It?* [Online]. Available from: <https://airbrake.io/blog/sdlc/waterfall-model> [Accessed 27 December 2019].

Regoli , N. (2019) *6 Advantages and Disadvantages of Scrum Methodology* [Online]. Available from: <https://connectusfund.org/6-advantages-and-disadvantages-of-scrum-methodology> [Accessed 27 December 2019].

Scott, K. (2019) *The Unified Process Explained*. Indianapolis: Pearson Education Corporate Sales Division.

Bajracharya, N. & Ghimire, S. (2017) *Changing Trends in Teej Celebration*. Kathmandu: myRepublica.

balsamiq. (2020) *Introduction to Balsamiq Cloud* [Online]. Available from: <https://balsamiq.com/wireframes/cloud/docs/intro/> [Accessed 31 May 2020].

Chandana. (2019) *Scrum Project Management: Pros and Cons* [Online]. Available from: <https://www.simplilearn.com/scrum-project-management-article> [Accessed 17 December 2019].

Cobb, C. (2019) *What Are the Advantages and Disadvantages of Agile and Scrum?* [Online]. Available from: <https://managedagile.com/what-are-the-advantages-and-disadvantages-of-agile-scrum/> [Accessed 28 December 2019].

CopyLeft. (2020) *System Testing Fundamentals* [Online]. Available from: <http://softwaretestingfundamentals.com/unit-testing/#:~:text=UNIT%20TESTING%20is%20a%20level,and%20usually%20a%20single%20output>. [Accessed 4 June 2020].

Dango Stars. (2019) *Why We Use Django Framework & What Is Django Used For* [Online]. Available from: <https://djangostars.com/blog/why-we-use-django-framework/> [Accessed 1 November 2019].

docsity. (2019) *online vanue booking management system*. Dr. A.P.J. Abdul Kalam Technical University.

CHAPTER 7: APPENDIX

7.1.1 PRE-SURVEY FORM

The screenshot shows a survey form with the following fields:

- Multi vendor ecommerce site** (Section title)
- * Required** (Text indicating a required field)
- name *** (Text label) followed by a text input field containing "Your answer".
- email *** (Text label) followed by a text input field containing "Your answer".
- Do you like to do online shopping? *** (Text label)
 - Yes
 - No

Figure 99 pre-survey form 1

The screenshot shows a survey form with the following questions:

- how often you do online shopping? ***
 - always
 - sometimes
 - never
- have you ever thought of selling product online ***
 - Yes
 - No
 - Maybe
- suppose you have more than one shop. would like to have different account for different shop or single account for you all shop ***
 - single
 - multiple
 - i dont know

Figure 100 pre-survey form 2

The screenshot shows a Google Form titled "PRE-SURVEY FORM3". It consists of three stacked questions, each enclosed in a light gray box with rounded corners.

- Question 1:** "which payment gateway do you use most of the time? *".
Options: esewa, khalti, imipay, fonepay, Other: _____.
- Question 2:** "would you like to chat to your customer if they have any queries? *".
Options: Yes, No, Maybe.
- Question 3:** "would you like to see your order progress of the product you have bought? *".
Options: Yes, No, Maybe.

At the bottom right of the form area, there is a small circular icon containing a blue pen nib.

Figure 101 PRE-SURVEY FORM3

The screenshot shows a Google Form titled "PRE-SURVEY FORM 4". It features a single question box with a light gray background and rounded corners.

Question: any suggestion?
Your answer _____

Below the question box is a purple rectangular button labeled "Submit".

At the bottom of the page, there is a note: "Never submit passwords through Google Forms." followed by "This content is neither created nor endorsed by Google. [Report Abuse](#) · [Terms of Service](#) · [Privacy Policy](#)".

On the left side, there is a small gray speech bubble icon with a white exclamation mark. On the right side, there is a small circular icon containing a blue pen nib.

Figure 102 PRE-SURVEY FORM 4

7.1.2 SAMPLE OF FILLED PRE-SURVEY FORMS

Multi vendor ecommerce site

* Required

name *

pujan raj rai

email *

pujanr0809@gmail.com

Do you like to do online shopping? *

Yes

No

Figure 103 SAMPLE OF FILLED PRE-SURVEY FORMS 1

how often you do online shopping? *

always

sometimes

never

have you ever thought of selling product online *

Yes

No

Maybe

suppose you have more than one shop. would like to have different account for different shop or single account for you all shop *

single

multiple

i dont know

Figure 104SAMPLE OF FILLED PRE-SURVEY FORMS 2

which payment gateway do you use most of the time *

esewa
 khalti
 imepay
 fonepay
 Other: _____

would you like to chat to your customer if they have any queries? *

Yes
 No
 Maybe

would you like to see your order progress of the product you have bought? *

Yes
 No
 Maybe

Figure 105AMPLE OF FILLED PRE-SURVEY FORMS 3

any suggestion?

please add review system on every product

Submit

Never submit passwords through Google Forms.

This content is neither created nor endorsed by Google. [Report Abuse](#) - [Terms of Service](#) - [Privacy Policy](#)

Google Forms

Figure 106 AMPLE OF FILLED PRE-SURVEY FORMS 4

7.1.3 PRE-SURVEY RESULT

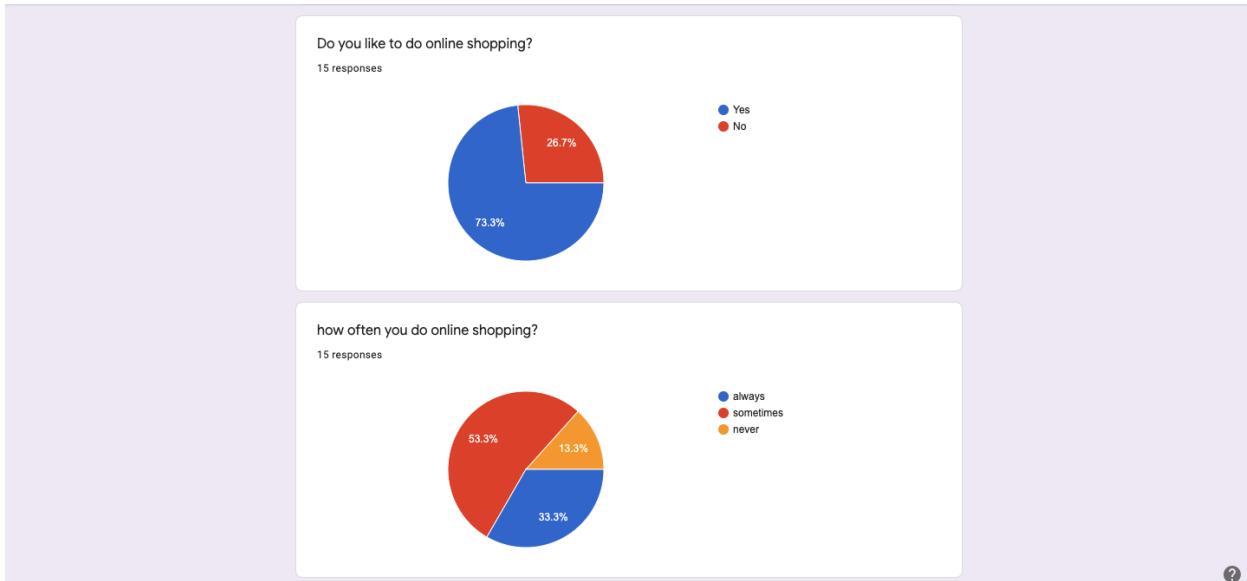


Figure 107 PRE-SURVEY RESULT 1

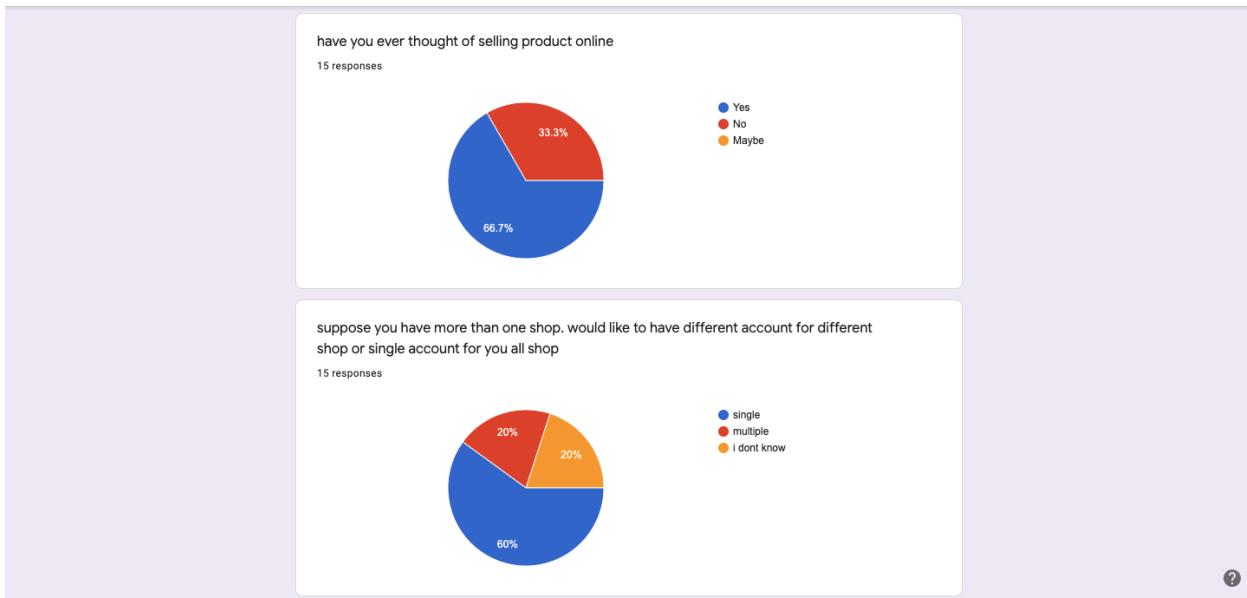


Figure 108 PRE-SURVEY RESULT 2

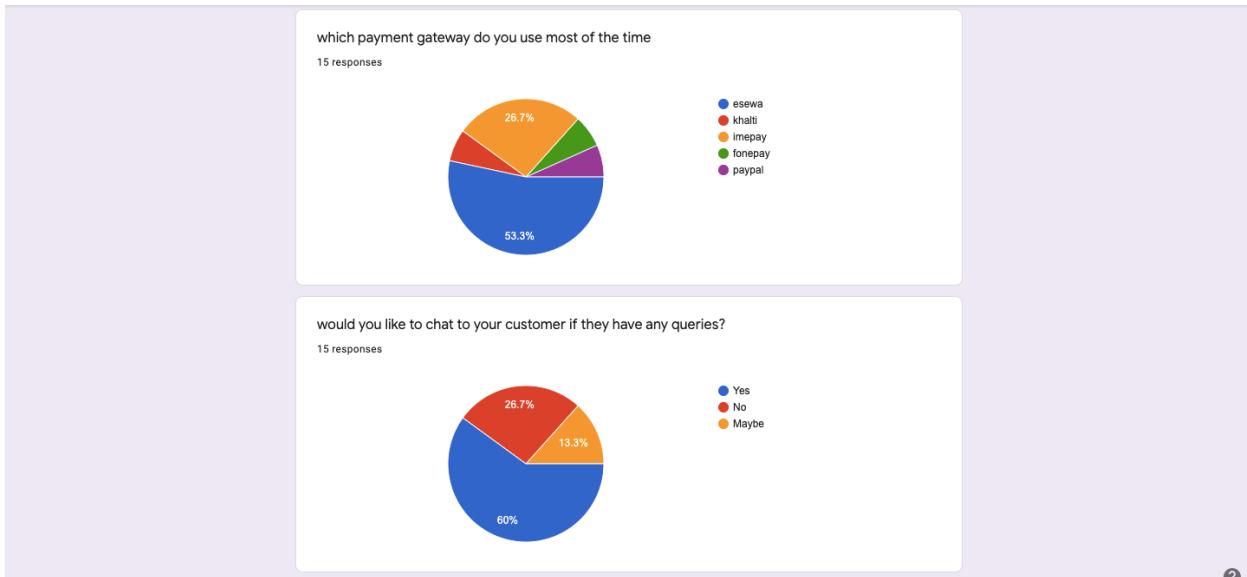


Figure 109 PRE-SURVEY RESULT 3

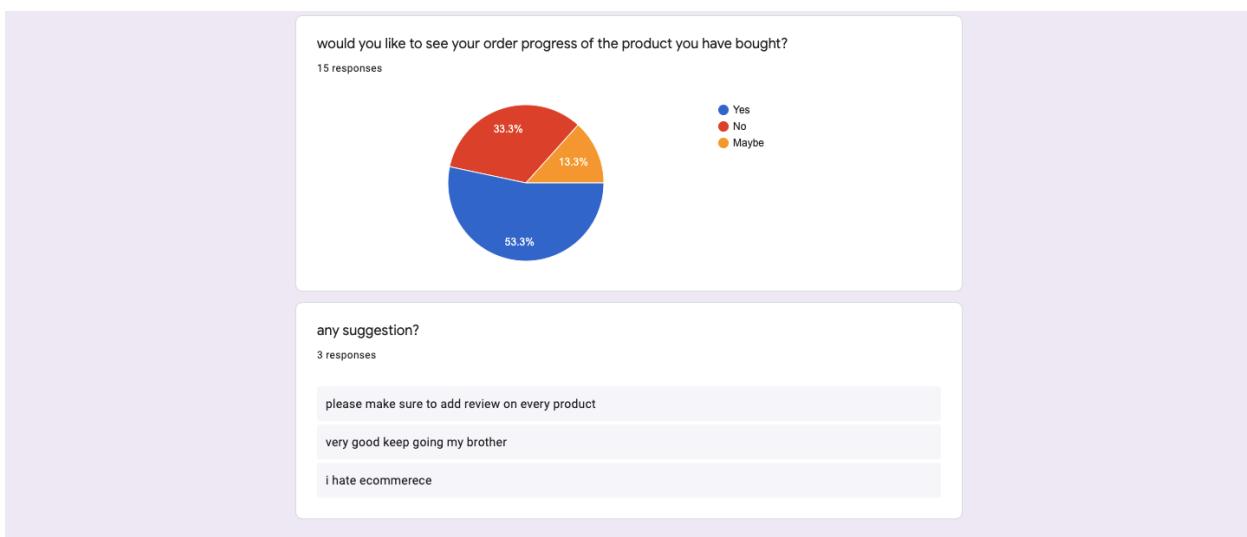


Figure 110 PRE-SURVEY RESULT 4

7.2 APPENDIX B: POST-SURVEY

7.2.1 POST-SURVEY FORM

multi-vendor ecommerce site

Form description

name *
Short answer text

email *
Short answer text

Figure 111 POST-SURVEY FORM 1

what was your over all experience using this website? *

very good

good

fair

poor

what do you find most attractive in this e-commerce website? *

UI

Functionality

Simplicity

Figure 112 POST-SURVEY FORM 2

Would you like to buy product in this website *

- Yes
- No
- Maybe

would you like to sell product in this website? *

- Yes
- No
- Maybe

how was the application to user *

- very easy
- easy
- difficult

Figure 113 POST-SURVEY FORM 3

would you like to recommend this website to your friend? *

Yes

No

Maybe

any suggestion?

Short answer text

Required

Form toolbar icons: +, -, T, R, M, V, E, ...

Figure 114 POST-SURVEY FORM 4

7.2.2 SAMPLE OF FILLED POST-SURVEY FORMS

The form consists of three stacked sections. The top section has a purple header bar with the title 'multi-vendor ecommerce site'. Below it is a red note 'Required'. The middle section asks for 'name *' and contains the entry 'pujan raj rai'. The bottom section asks for 'email *' and contains the entry 'pujanr0809@gmail.com'. The third section is a question about overall experience with multiple choice options: 'very good', 'good', 'fair', and 'poor'. The 'fair' option is selected with a blue circle.

multi-vendor ecommerce site

* Required

name *

pujan raj rai

email *

pujanr0809@gmail.com

what was your over all experience using this website? *

very good

good

fair

poor

Figure 115 SAMPLE OF FILLED POST-SURVEY FORMS 1

what do you find most attractive in this e-commerce website? *

- UI
- Functionality
- Simplicity

Would you like to buy product in this website *

- Yes
- No
- Maybe

would you like to sell product in this website? *

- Yes
- No
- Maybe

Figure 116 SAMPLE OF FILLED POST-SURVEY FORMS 2

how was the application to user *

very easy
 easy
 difficult
 very difficult

would you like to recommend this website to your friend? *

Yes
 No
 Maybe

any suggestion?

very good website waiting for launching

Submit

Figure 117 SAMPLE OF FILLED POST-SURVEY FORMS 3

7.2.3 POST-SURVEY RESULT

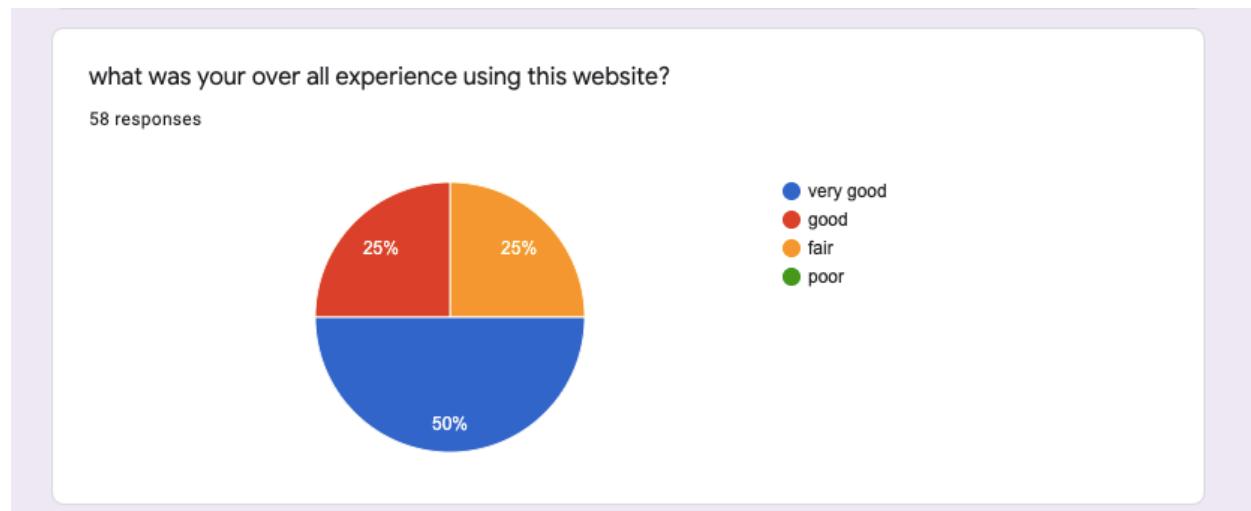


Figure 118 POST-SURVEY RESULT 1

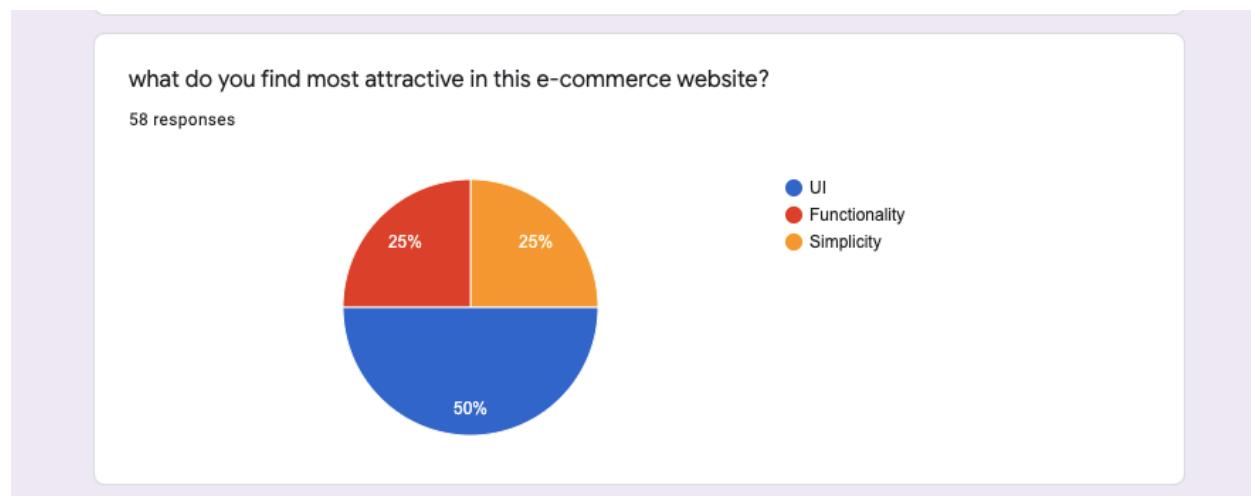


Figure 119 POST-SURVEY RESULT 2



Figure 120 POST-SURVEY RESULT 3

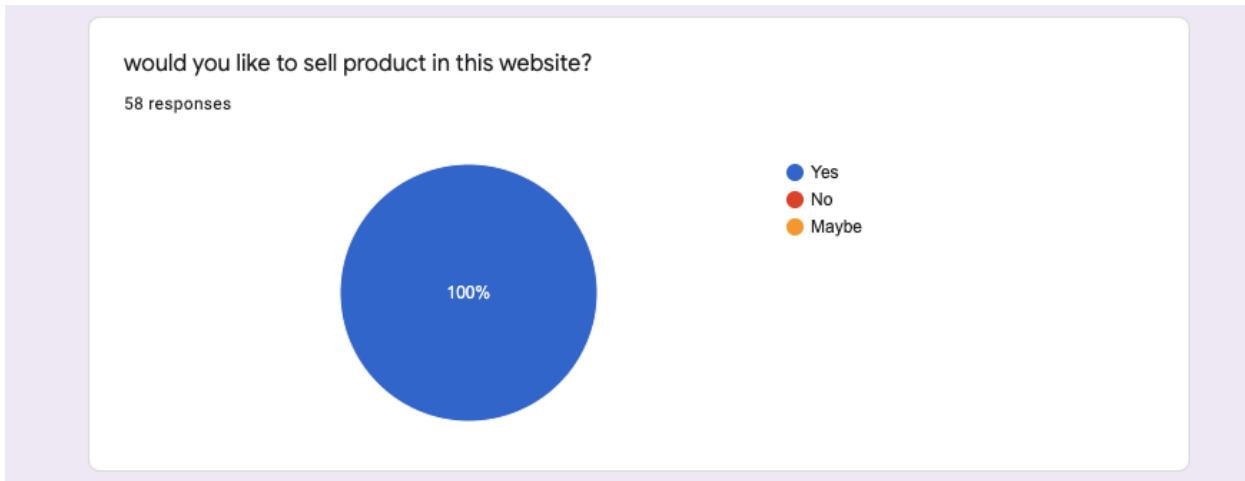


Figure 121 POST-SURVEY RESULT 4

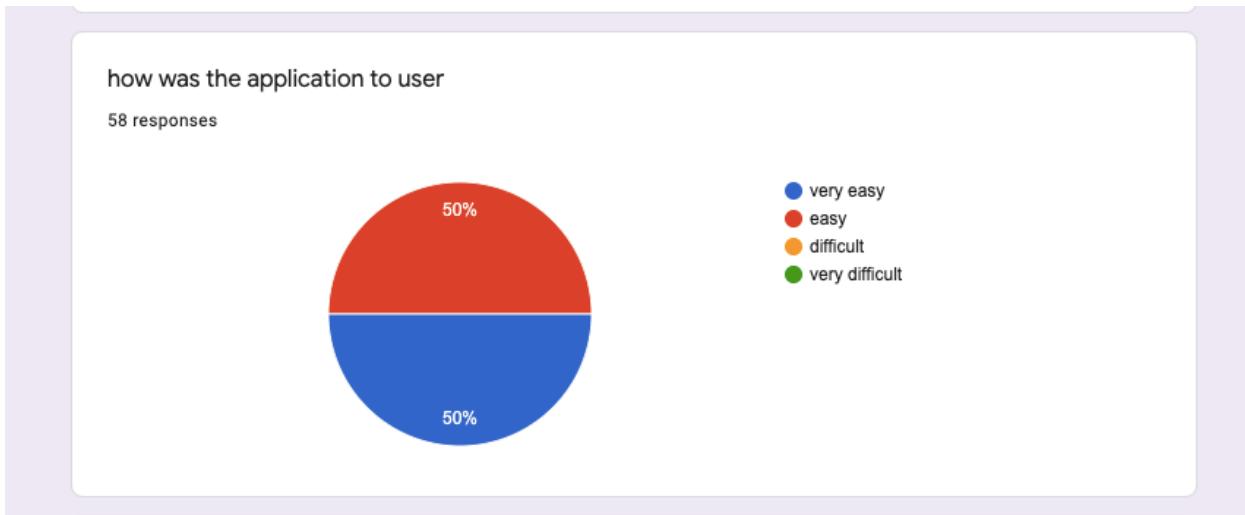


Figure 122 POST-SURVEY RESULT 5

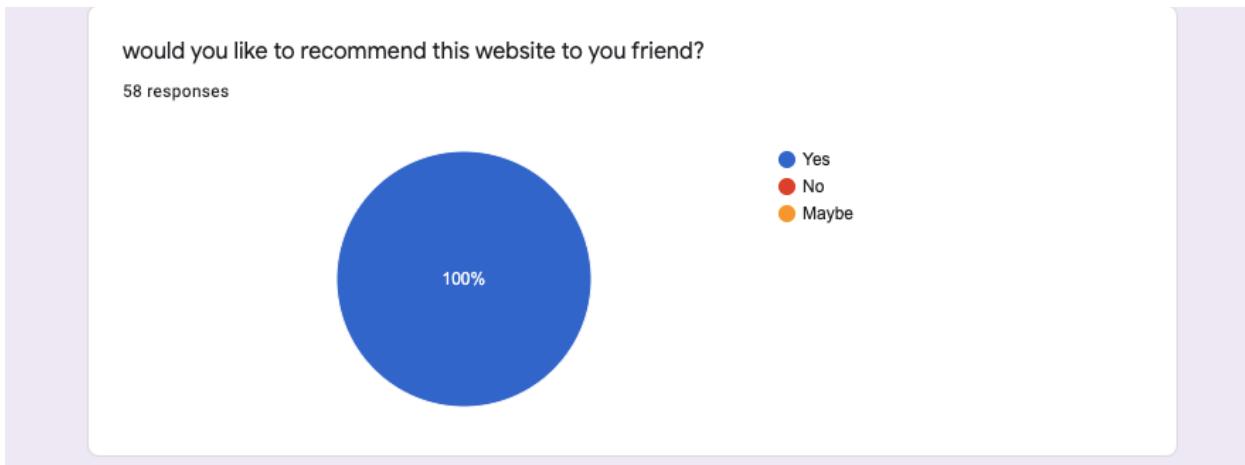


Figure 123 POST-SURVEY RESULT 6

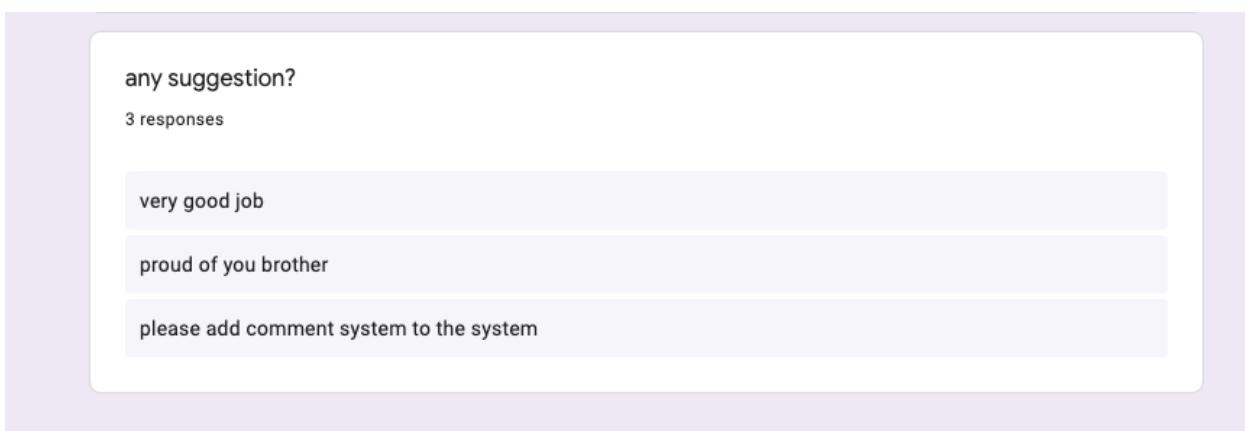


Figure 124 POST-SURVEY RESULT 7

7.3 APPENDIX C: SAMPLE CODES

7.3.1 SAMPLE CODE OF THE UI

```
{% load static %}

<!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, shrink-to-fit=no">
<meta name="description" content="">
<meta name="author" content="">

<title>SB Admin - 2 - Tables</title>

<!-- Custom fonts for this template -->
<link href="{% static 'vendor/fontawesome-free/css/all.min.css' %}" rel="stylesheet" type="text/css">
<link href="https://fonts.googleapis.com/css?family=Nunito:200,200i,300,300i,400,400i,600,600i,700,700i,800,800i,900,900i" rel="stylesheet">

<!-- Custom styles for this template -->
<link href="{% static 'css/sb-admin-2.min.css' %}" rel="stylesheet">

<!-- Custom styles for this page -->
```

```

<link href="{% static 'vendor/datatables/dataTables.bootstrap4.min.css' %}"
rel="stylesheet">

</head>

<body id="page-top">

<!-- Page Wrapper -->
<div id="wrapper">

    <!-- Sidebar -->
    <ul class="navbar-nav bg-gradient-primary sidebar sidebar-dark accordion" id="accordionSidebar">

        <!-- Sidebar - Brand -->
        <a class="sidebar-brand d-flex align-items-center justify-content-center" href="index.html">
            <div class="sidebar-brand-icon rotate-n-15">
                <i class="fas fa-laugh-wink"></i>
            </div>
            <div class="sidebar-brand-text mx-3">Saman<sup>sup</sup>Ch</div>
        </a>

        <!-- Divider -->
        <hr class="sidebar-divider my-0">

        <!-- Nav Item - Dashboard -->
        <li class="nav-item" style="background-color: #f8d7da; border-radius: 10px; padding: 10px; margin-bottom: 10px;">
            <a class="nav-link" href="#" style="color: black; text-decoration: none; font-weight: bold; font-size: 1.2em; border-radius: 10px; padding: 10px; margin-bottom: 5px;">
                <i class="fas fa-tachometer-alt" style="font-size: 1.2em; margin-right: 10px; border-radius: 10px; padding: 5px; border: 1px solid black; margin-bottom: 5px;"></i>
                <span>Dashboard</span>
            </a>
        </li>
    </ul>

```

```

</li>

<!-- Divider -->
<hr class="sidebar-divider">

<!-- Heading -->
<div class="sidebar-heading">
    Interface
</div>

<!-- Nav Item - Profile -->
<li class="nav-item">
    <a class="nav-link" href="{% url 'seller_profile:seller_profile_view' %}>
        <i class="fas fa-fw fa-user"></i>
        <span>My Profile</span>
    </a>
</li>

<!-- Nav Item - Shop -->
<li class="nav-item">
    <a class="nav-link" href="{% url 'shop_profile:shop_profile_show' %}>
        <i class="fas fa-fw fa-store"></i>
        <span>Shop Profile</span>
    </a>
</li>

<!-- Nav Item - Messages -->
<li class="nav-item">
    <a class="nav-link" href="#">
        <i class="fas fa-fw fa-inbox"></i>
        <span>Messages</span>
    </a>
</li>

```

```

<!-- Divider -->
<hr class="sidebar-divider d-none d-md-block">

<!-- Sidebar Toggler (Sidebar) -->
<div class="text-center d-none d-md-inline">
    <button class="rounded-circle border-0" id="sidebarToggle"></button>
</div>

</ul>
<!-- End of Sidebar -->

<!-- Content Wrapper -->
<div id="content-wrapper" class="d-flex flex-column">

<!-- Main Content -->
<div id="content">

    <!-- Topbar -->
    <nav class="navbar navbar-expand navbar-light bg-white topbar mb-4 static-top shadow">

        <!-- Sidebar Toggle (Topbar) -->
        <form class="form-inline">
            <button id="sidebarToggleTop" class="btn btn-link d-md-none rounded-circle mr-3">
                <i class="fa fa-bars"></i>
            </button>
        </form>

        <!-- Topbar Search -->
        <div class="topbar-search" data-search="search-form">
            <input type="text" class="form-control" placeholder="Search" aria-label="Search" data-search="search-input"/>
        </div>
    </nav>
</div>

```

```

<form
      class="d-none d-sm-inline-block form-inline mr-auto ml-md-3 my-2 my-md-0 mw-100 navbar-search">
  <div class="input-group">
    <input type="text" class="form-control bg-light border-0 small"
placeholder="Search" aria-label="Search" aria-describedby="basic-addon2">
    <div class="input-group-append">
      <button class="btn btn-primary" type="button">
        <i class="fas fa-search fa-sm"></i>
      </button>
    </div>
  </div>
</form>

<!-- Topbar Navbar -->
<ul class="navbar-nav ml-auto">
  <!-- Nav Item - Messages -->
  <li class="nav-item dropdown no-arrow mx-1">
    <a class="nav-link dropdown-toggle" href="#" id="messagesDropdown" role="button"
       data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
      <i class="fas fa-envelope fa-fw"></i>
      <!-- Counter - Messages -->
      <span class="badge badge-danger badge-counter">7</span>
    </a>
  </li>
</ul>

```

```

<!--          Dropdown          -          Messages          -->

</li>

<div          class="topbar-divider"          d-none          d-sm-block"></div>

<!--          Nav          Item          -          User          Information          -->
<li          class="nav-item"          dropdown          no-arrow">
    <a          class="nav-link          dropdown-toggle"          href="#"          id="userDropdown"
        role="button"
        data-toggle="dropdown"          aria-haspopup="true"          aria-expanded="false">
        <span          class="mr-2          d-none          d-lg-inline          text-gray-600          small">{{
            request.user
        }}</span>
        
    </a>
    <!--          Dropdown          -          User          Information          -->

</li>

</ul>

</nav>
<!--          End          of          Topbar          -->
{%
    block          body
%}
{%
    endblock
%}

</div>
<!--          End          of          Main          Content          -->

```

```

<!-- Footer -->
<footer class="sticky-footer bg-white" style="background-color: #f5f5f5;">
  <div class="container" style="text-align: center; margin-top: 20px;">
    <div class="copyright" style="margin-bottom: 10px;">
      <span>Copyright © Your Website 2020</span>
    </div>
  </div>
</footer>

<!-- End of Footer -->

</div>

<!-- End of Content Wrapper -->

</div>

<!-- End of Page Wrapper -->

<!-- Scroll to Top Button -->
<a href="#page-top" class="scroll-to-top rounded" style="position: fixed; bottom: 10px; right: 10px; background-color: #f5f5f5; color: #f5f5f5; border-radius: 50%; padding: 10px; font-size: 16px; transition: all 0.3s ease-in-out; z-index: 1000; border: 1px solid #f5f5f5;">
  <i class="fas fa-angle-up" style="font-size: 1em; margin-right: 5px; vertical-align: middle;"/>
</a>

<!-- Bootstrap core JavaScript -->
<script src="{% static 'vendor/jquery/jquery.min.js' %}"></script>
<script src="{% static 'vendor/bootstrap/js/bootstrap.bundle.min.js' %}"></script>

<!-- Core plugin JavaScript -->
<script src="{% static 'vendor/jquery-easing/jquery.easing.min.js' %}"></script>

<!-- Custom scripts for all pages -->
<script src="{% static 'js/sb-admin-2.min.js' %}"></script>

```

```

<!--          Page          level          plugins          -->
<script    src="{% static 'vendor/datatables/jquery.dataTables.min.js' %}"></script>
<script    src="{% static 'vendor/datatables/dataTables.bootstrap4.min.js' %}"></script>

<!--          Page          level          custom          scripts          -->
<script    src="{% static 'js/demo/datatables-demo.js' %}"></script>

</body>

</html>
{%
    load          static          %}
<!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, shrink-to-fit=no">
    <meta           name="description" content="">
    <meta           name="author" content="">

    <title>SB          Admin          2          -          Dashboard</title>

    <!--          Custom          fonts          for          this          template-->
    <link href="{% static 'vendor/fontawesome-free/css/all.min.css' %}" rel="stylesheet" type="text/css">
    <link

```

```

    href="https://fonts.googleapis.com/css?family=Nunito:200,200i,300,300i,400,400i,600,6
00i,700,700i,800,800i,900,900i"
        rel="stylesheet">

    <!-- Custom styles for this template-->
    <link href="{% static 'css/sb-admin-2.min.css' %}" rel="stylesheet">

</head>

<body id="page-top">

<!-- Page Wrapper -->
<div id="wrapper">

    <!-- Sidebar -->
    <ul class="navbar-nav bg-gradient-primary sidebar sidebar-dark accordion" id="accordionSidebar">

        <!-- Sidebar - Brand -->
        <a class="sidebar-brand d-flex align-items-center justify-content-center" href="index.html">
            <div class="sidebar-brand-icon rotate-n-15">
                <i class="fas fa-laugh-wink"></i>
            </div>
            <div class="sidebar-brand-text mx-3">Saman<sup>Ch&lt;/sup></div>
        </a>

        <!-- Divider -->
        <hr class="sidebar-divider my-0">

    <!-- Nav Item - Dashboard -->

```

```

<li class="nav-item" href="#"><span>Dashboard</span></a>
</li>

<!-- Divider -->
<hr class="sidebar-divider">

<!-- Heading -->
<div class="sidebar-heading">Interface
<a class="nav-link collapsed" href="#" data-toggle="collapse" data-target="#collapseTwo">
<span>Components</span>
</a>
<div id="collapseTwo" class="collapse" aria-labelledby="headingTwo" data-parent="#accordionSidebar">
<div class="bg-white py-2 collapse-inner rounded">
<h6 class="collapse-header">Custom Components:</h6>
<a class="collapse-item" href="buttons.html">Buttons</a>
<a class="collapse-item" href="cards.html">Cards</a>
</div>
</div>
</li>

```

```

<!-- Nav Item - Utilities Collapse Menu -->
<li class="nav-item">
  <a class="nav-link collapsed" href="#" data-toggle="collapse" data-target="#collapseUtilities"
     aria-expanded="true" aria-controls="collapseUtilities">
    <i class="fas fa-fw fa-wrench"></i>
    <span>Utilities</span>
  </a>
  <div id="collapseUtilities" class="collapse" aria-labelledby="headingUtilities"
        data-parent="#accordionSidebar">
    <div class="bg-white py-2 collapse-inner rounded">
      <h6 class="collapse-header">Custom Utilities:</h6>
      <a class="collapse-item" href="utilities-color.html">Colors</a>
      <a class="collapse-item" href="utilities-border.html">Borders</a>
      <a class="collapse-item" href="utilities-animation.html">Animations</a>
      <a class="collapse-item" href="utilities-other.html">Other</a>
    </div>
  </div>
</li>

<!-- Divider -->
<hr class="sidebar-divider">

<!-- Heading -->
<div class="sidebar-heading">
  Addons
</div>

<!-- Nav Item - Pages Collapse Menu -->
<li class="nav-item">

```

```

        <a class="nav-link collapsed" href="#" data-toggle="collapse" data-
target="#collapsePages"
            aria-expanded="true" aria-controls="collapsePages">
            <i class="fas fa-fw fa-folder"></i>
            <span>Pages</span>
        </a>
        <div id="collapsePages" class="collapse" aria-labelledby="headingPages" data-
parent="#accordionSidebar">
            <div class="bg-white py-2 collapse-inner rounded">
                <h6 class="collapse-header">Login Screens:</h6>
                <a class="collapse-item" href="login.html">Login</a>
                <a class="collapse-item" href="register.html">Register</a>
                <a class="collapse-item" href="forgot-password.html">Forgot
Password</a>
                <div class="collapse-divider"></div>
                <h6 class="collapse-header">Other Pages:</h6>
                <a class="collapse-item" href="404.html">404 Page</a>
                <a class="collapse-item" href="blank.html">Blank Page</a>
            </div>
        </div>
    </li>

    <!-- Nav Item - Charts -->
    <li class="nav-item">
        <a class="nav-link" href="charts.html">
            <i class="fas fa-fw fa-chart-area"></i>
            <span>Charts</span>
        </a>
    </li>

    <!-- Nav Item - Tables -->
    <li class="nav-item active">

```

```

<a href="tables.html">
  <i class="fas fa-fw fa-table"></i>
  <span>Tables</span></a>
</li>

<!-- Divider -->
<hr class="sidebar-divider d-none d-md-block">

<!-- Sidebar Toggler (Sidebar) -->
<div class="text-center d-none d-md-inline">
  <button class="rounded-circle border-0" id="sidebarToggle"></button>
</div>

</ul>

<!-- End of Sidebar -->

<!-- Content Wrapper -->
<div id="content-wrapper" class="d-flex flex-column">

<!-- Main Content -->
<div id="content">

  <!-- Topbar -->
  <nav class="navbar navbar-expand navbar-light bg-white topbar mb-4 static-top shadow">

    <!-- Sidebar Toggle (Topbar) -->
    <form class="form-inline">
      <button id="sidebarToggleTop" class="btn btn-link d-md-none rounded-circle mr-3">
        <i class="fa fa-bars"></i>
      </button>
    </form>
  </nav>
</div>

```

```
</button>
</form>

<!-- Topbar Search -->
<form class="d-none d-sm-inline-block form-inline mr-auto ml-md-3 my-2 my-md-0 mw-100 navbar-search">
  <div class="input-group">
    <input type="text" class="form-control bg-light border-0 small" placeholder="Search" aria-label="Search" aria-describedby="basic-addon2" for...>
    <div class="input-group-append">
      <button class="btn btn-primary" type="button">
        <i class="fas fa-search fa-sm"></i>
      </button>
    </div>
  </div>
</form>

<!-- Topbar Navbar -->
<ul class="navbar-nav ml-auto">
  <!-- Nav Item - Search Dropdown (Visible Only XS) -->
  <li class="nav-item dropdown no-arrow d-sm-none">
    <a class="nav-link dropdown-toggle" href="#" id="searchDropdown" role="button" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
      <i class="fas fa-search fa-fw"></i>
    </a>
    <!-- Dropdown Messages -->
    <div class="dropdown-menu dropdown-menu-right p-3 shadow animated--grow-in" style="position: absolute; top: 100%; left: 0; width: 100%; z-index: 1000; background-color: #fff; border: 1px solid #dadce0; border-radius: 10px; padding: 10px; margin-top: 10px; font-size: 12px; font-weight: bold; color: #333; text-decoration: none; list-style-type: none; text-align: left; opacity: 0; visibility: hidden; transition: all 0.3s ease-in-out; position: absolute; top: 100%; left: 0; width: 100%; z-index: 1000; background-color: #fff; border: 1px solid #dadce0; border-radius: 10px; padding: 10px; margin-top: 10px; font-size: 12px; font-weight: bold; color: #333; text-decoration: none; list-style-type: none; text-align: left; opacity: 0; visibility: hidden; transition: all 0.3s ease-in-out;">
      <ul style="list-style-type: none; padding-left: 0; margin: 0; font-size: 12px; font-weight: bold; color: #333; text-decoration: none; text-align: left; opacity: 0; visibility: hidden; transition: all 0.3s ease-in-out;">
        <li style="margin-bottom: 5px;">Messages
      </ul>
    </div>
  </li>
</ul>
```

```

-grow-in"

    aria-labelledby="searchDropdown">
<form class="form-inline mr-auto w-100 navbar-search">
    <div class="input-group">
        <input type="text" class="form-control bg-light border-0 small"
            placeholder="Search for..." aria-label="Search"
            aria-describedby="basic-addon2">
        <div class="input-group-append">
            <button class="btn btn-primary" type="button">
                <i class="fas fa-search fa-sm"></i>
            </button>
        </div>
    </div>
</form>
</div>
</li>

<!-- Nav Item - Alerts -->
<li class="nav-item dropdown no-arrow mx-1">
    <a class="nav-link dropdown-toggle" href="#" id="alertsDropdown"
        role="button"
        data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
        <i class="fas fa-bell fa-fw"></i>
        <!-- Counter - Alerts -->
        <span class="badge badge-danger badge-counter">3+</span>
    </a>
    <!-- Dropdown - Alerts -->
    <div class="dropdown-list dropdown-menu dropdown-menu-right shadow
        animated--grow-in"
        aria-labelledby="alertsDropdown">
        <h6 class="dropdown-header">

```

```

    Alerts                                         Center
    </h6>
    <a   class="dropdown-item"   d-flex   align-items-center"   href="#">
        <div                                         class="mr-3">
            <div               class="icon-circle"           bg-primary">
                <i       class="fas"         fa-file-alt      text-white"></i>
            </div>
        </div>
        <div>
            <div   class="small  text-gray-500">December  12,  2019</div>
            <span class="font-weight-bold">A new monthly report is ready to
download!</span>
        </div>
    </a>
    <a   class="dropdown-item"   d-flex   align-items-center"   href="#">
        <div                                         class="mr-3">
            <div               class="icon-circle"           bg-success">
                <i       class="fas"         fa-donate      text-white"></i>
            </div>
        </div>
        <div>
            <div   class="small  text-gray-500">December  7,  2019</div>
            $290.29   has   been   deposited   into   your   account!
        </div>
    </a>
    <a   class="dropdown-item"   d-flex   align-items-center"   href="#">
        <div                                         class="mr-3">
            <div               class="icon-circle"           bg-warning">
                <i      class="fas"       fa-exclamation-triangle  text-white"></i>
            </div>
        </div>
    </a>

```

```

<div>
  <div class="small text-gray-500">December 2, 2019</div>
  Spending Alert: We've noticed unusually high spending for your
account.

  </div>
  </a>
  <a class="dropdown-item text-center small text-gray-500"
  href="#">Show All Alerts</a>
</div>
</li>

<!-- Nav Item - Messages -->
<li class="nav-item dropdown no-arrow mx-1">
  <a class="nav-link dropdown-toggle" href="#" id="messagesDropdown"
  role="button"
    data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
    <i class="fas fa-envelope fa-fw"></i>
    <!-- Counter - Messages -->
    <span class="badge badge-danger badge-counter">7</span>
  </a>
  <!-- Dropdown - Messages -->
  <div class="dropdown-list dropdown-menu dropdown-menu-right shadow
  animated--grow-in"
    aria-labelledby="messagesDropdown">
    <h6 class="dropdown-header">
      Message Center
    </h6>
    <a class="dropdown-item d-flex align-items-center" href="#">
      <div class="dropdown-list-image mr-3">
        
      </div>
    </a>
  </div>
</li>

```

```

        <div class="status-indicator bg-success"></div>
      </div>
      <div class="font-weight-bold">
        <div class="text-truncate">Hi there! I am wondering if you can help
me with a
problem I've been having.
      </div>
      <div class="small text-gray-500">Emily Fowler · 58m</div>
    </div>
  </a>
  <a class="dropdown-item d-flex align-items-center" href="#">
    <div class="dropdown-list-image mr-3">
      
      <div class="status-indicator"></div>
    </div>
    <div>
      <div class="text-truncate">I have the photos that you ordered last
month, how
would you like them sent to you?
    </div>
    <div class="small text-gray-500">Jae Chun · 1d</div>
  </div>
</a>
<a class="dropdown-item d-flex align-items-center" href="#">
  <div class="dropdown-list-image mr-3">
    
    <div class="status-indicator bg-warning"></div>
  </div>
  <div>

```

```

<div class="text-truncate">Last month's report looks great, I am  

very happy with  

the progress so far, keep up the good work!  

</div>  

<div class="small text-gray-500">Morgan Alvarez · 2d</div>  

</div>  

</a>  

<a class="dropdown-item d-flex align-items-center" href="#">  

<div class="dropdown-list-image mr-3">  

<div class="status-indicator bg-success"></div>  

</div>  

<div>  

<div class="text-truncate">Am I a good boy? The reason I ask is  

because someone  

told me that people say this to all dogs, even if they aren't good...  

</div>  

<div class="small text-gray-500">Chicken the Dog · 2w</div>  

</div>  

</a>  

<a class="dropdown-item text-center small text-gray-500"  

href="#">Read More Messages</a>  

</div>  

</li>  

<div class="topbar-divider d-none d-sm-block"></div>  

<!-- Nav Item - User Information -->  

<li class="nav-item dropdown no-arrow">

```

```

        <a class="nav-link dropdown-toggle" href="#" id="userDropdown"
role="button"

            data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
            <span class="mr-2 d-none d-lg-inline text-gray-600 small">Douglas
McGee</span>
            
        </a>
        <!-- Dropdown - User Information -->
        <div class="dropdown-menu dropdown-menu-right shadow animated--grow-in"
aria-labelledby="userDropdown">
            <a class="dropdown-item" href="#>
                <i class="fas fa-user fa-sm fa-fw mr-2 text-gray-400"></i>
                Profile
            </a>
            <a class="dropdown-item" href="#>
                <i class="fas fa-cogs fa-sm fa-fw mr-2 text-gray-400"></i>
                Settings
            </a>
            <a class="dropdown-item" href="#>
                <i class="fas fa-list fa-sm fa-fw mr-2 text-gray-400"></i>
                Activity
            </a>
            <div class="dropdown-divider"></div>
            <a class="dropdown-item" href="#" data-toggle="modal" data-
target="#logoutModal">
                <i class="fas fa-sign-out-alt fa-sm fa-fw mr-2 text-gray-400"></i>
                Logout
            </a>
        </div>
    
```

```

        </li>

    </ul>

</nav>
<!-- End of Topbar -->

<!-- Begin Page Content -->
<div class="container-fluid">

    <!-- Page Heading -->
    <h1 class="h3 mb-2 text-gray-800">Tables</h1>
    <p class="mb-4">DataTables is a third party plugin that is used to generate the
    demo table below.

    For more information about DataTables, please visit the <a target="_blank"
    href="https://datatables.net">official
    DataTables documentation</a>.</p>

    <!-- DataTables Example -->
    <div class="card shadow mb-4">
        <div class="card-header py-3">
            <h6 class="m-0 font-weight-bold text-primary">DataTables
            Example</h6>
        </div>
        <div class="card-body">
            <div class="table-responsive">
                <table class="table table-bordered" id="dataTable" width="100%">
                    <thead>
                        <tr>
                            <th>Name</th>

```


<td>2011/07/25</td>		
<td>\$170,750</td>		
</tr>		
<tr>		
<td>Ashton		Cox</td>
<td>Junior	Technical	Author</td>
<td>San		Francisco</td>
<td>66</td>		
<td>2009/01/12</td>		
<td>\$86,000</td>		
</tr>		
<tr>		
<td>Cedric		Kelly</td>
<td>Senior	Javascript	Developer</td>
<td>Edinburgh</td>		
<td>22</td>		
<td>2012/03/29</td>		
<td>\$433,060</td>		
</tr>		
<tr>		
<td>Airi		Satou</td>
<td>Accountant</td>		
<td>Tokyo</td>		
<td>33</td>		
<td>2008/11/28</td>		
<td>\$162,700</td>		
</tr>		
<tr>		
<td>Brielle		Williamson</td>
<td>Integration		Specialist</td>
<td>New		York</td>

<td>61</td>	
<td>2012/12/02</td>	
<td>\$372,000</td>	
</tr>	
<tr>	
<td>Herrod	Chandler</td>
<td>Sales	Assistant</td>
<td>San	Francisco</td>
<td>59</td>	
<td>2012/08/06</td>	
<td>\$137,500</td>	
</tr>	
<tr>	
<td>Rhona	Davidson</td>
<td>Integration	Specialist</td>
<td>Tokyo</td>	
<td>55</td>	
<td>2010/10/14</td>	
<td>\$327,900</td>	
</tr>	
<tr>	
<td>Colleen	Hurst</td>
<td>Javascript	Developer</td>
<td>San	Francisco</td>
<td>39</td>	
<td>2009/09/15</td>	
<td>\$205,500</td>	
</tr>	
<tr>	
<td>Sonya	Frost</td>
<td>Software	Engineer</td>

<td>Edinburgh</td>	
<td>23</td>	
<td>2008/12/13</td>	
<td>\$103,600</td>	
</tr>	
<tr>	
<td>Jena	Gaines</td>
<td>Office	Manager</td>
<td>London</td>	
<td>30</td>	
<td>2008/12/19</td>	
<td>\$90,560</td>	
</tr>	
<tr>	
<td>Quinn	Flynn</td>
<td>Support	Lead</td>
<td>Edinburgh</td>	
<td>22</td>	
<td>2013/03/03</td>	
<td>\$342,000</td>	
</tr>	
<tr>	
<td>Charde	Marshall</td>
<td>Regional	Director</td>
<td>San	Francisco</td>
<td>36</td>	
<td>2008/10/16</td>	
<td>\$470,600</td>	
</tr>	
<tr>	
<td>Haley	Kennedy</td>

<td>Senior	Marketing	Designer</td>
<td>London</td>		
<td>43</td>		
<td>2012/12/18</td>		
<td>\$313,500</td>		
</tr>		
<tr>		
<td>Tatyana	Fitzpatrick</td>	
<td>Regional	Director</td>	
<td>London</td>		
<td>19</td>		
<td>2010/03/17</td>		
<td>\$385,750</td>		
</tr>		
<tr>		
<td>Michael	Silva</td>	
<td>Marketing	Designer</td>	
<td>London</td>		
<td>66</td>		
<td>2012/11/27</td>		
<td>\$198,500</td>		
</tr>		
<tr>		
<td>Paul	Byrd</td>	
<td>Chief	Financial	Officer
<td>New	(CFO)</td>	
<td>64</td>		York</td>
<td>2010/06/09</td>		
<td>\$725,000</td>		
</tr>		
<tr>		

<td>Gloria	Little</td>
<td>Systems	Administrator</td>
<td>New	York</td>
<td>59</td>	
<td>2009/04/10</td>	
<td>\$237,500</td>	
</tr>	
<tr>	
<td>Bradley	Greer</td>
<td>Software	Engineer</td>
<td>London</td>	
<td>41</td>	
<td>2012/10/13</td>	
<td>\$132,000</td>	
</tr>	
<tr>	
<td>Dai	Rios</td>
<td>Personnel	Lead</td>
<td>Edinburgh</td>	
<td>35</td>	
<td>2012/09/26</td>	
<td>\$217,500</td>	
</tr>	
<tr>	
<td>Jenette	Caldwell</td>
<td>Development	Lead</td>
<td>New	York</td>
<td>30</td>	
<td>2011/09/03</td>	
<td>\$345,000</td>	
</tr>	

<td>Yuri			Berry</td>
<td>Chief	Marketing	Officer	(CMO)</td>
<td>New			York</td>
<td>40</td>			
<td>2009/06/25</td>			
<td>\$675,000</td>			
</tr>			
<tr>			
<td>Caesar		Vance</td>	
<td>Pre-Sales		Support</td>	
<td>New		York</td>	
<td>21</td>			
<td>2011/12/12</td>			
<td>\$106,450</td>			
</tr>			
<tr>			
<td>Doris		Wilder</td>	
<td>Sales		Assistant</td>	
<td>Sidney</td>			
<td>23</td>			
<td>2010/09/20</td>			
<td>\$85,600</td>			
</tr>			
<tr>			
<td>Angelica		Ramos</td>	
<td>Chief	Executive	Officer	(CEO)</td>
<td>London</td>			
<td>47</td>			
<td>2009/10/09</td>			
<td>\$1,200,000</td>			

</tr>		
<tr>		
<td>Gavin		Joyce</td>
<td>Developer</td>		
<td>Edinburgh</td>		
<td>42</td>		
<td>2010/12/22</td>		
<td>\$92,575</td>		
</tr>		
<tr>		
<td>Jennifer		Chang</td>
<td>Regional		Director</td>
<td>Singapore</td>		
<td>28</td>		
<td>2010/11/14</td>		
<td>\$357,650</td>		
</tr>		
<tr>		
<td>Brenden		Wagner</td>
<td>Software		Engineer</td>
<td>San		Francisco</td>
<td>28</td>		
<td>2011/06/07</td>		
<td>\$206,850</td>		
</tr>		
<tr>		
<td>Fiona		Green</td>
<td>Chief	Operating	Officer (COO)</td>
<td>San		Francisco</td>
<td>48</td>		
<td>2010/03/11</td>		

<td>\$850,000</td>	
</tr>	
<tr>	
<td>Shou	Itou</td>
<td>Regional	Marketing</td>
<td>Tokyo</td>	
<td>20</td>	
<td>2011/08/14</td>	
<td>\$163,000</td>	
</tr>	
<tr>	
<td>Michelle	House</td>
<td>Integration	Specialist</td>
<td>Sidney</td>	
<td>37</td>	
<td>2011/06/02</td>	
<td>\$95,400</td>	
</tr>	
<tr>	
<td>Suki	Burks</td>
<td>Developer</td>	
<td>London</td>	
<td>53</td>	
<td>2009/10/22</td>	
<td>\$114,500</td>	
</tr>	
<tr>	
<td>Prescott	Bartlett</td>
<td>Technical	Author</td>
<td>London</td>	
<td>27</td>	

<td>2011/05/07</td>	
<td>\$145,000</td>	
</tr>	
<tr>	
<td>Gavin	Cortez</td>
<td>Team	Leader</td>
<td>San	Francisco</td>
<td>22</td>	
<td>2008/10/26</td>	
<td>\$235,500</td>	
</tr>	
<tr>	
<td>Martena	Mccray</td>
<td>Post-Sales	support</td>
<td>Edinburgh</td>	
<td>46</td>	
<td>2011/03/09</td>	
<td>\$324,050</td>	
</tr>	
<tr>	
<td>Unity	Butler</td>
<td>Marketing	Designer</td>
<td>San	Francisco</td>
<td>47</td>	
<td>2009/12/09</td>	
<td>\$85,675</td>	
</tr>	
<tr>	
<td>Howard	Hatfield</td>
<td>Office	Manager</td>
<td>San	Francisco</td>

<td>51</td>	
<td>2008/12/16</td>	
<td>\$164,500</td>	
</tr>	
<tr>	
<td>Hope	Fuentes</td>
<td>Secretary</td>	
<td>San	Francisco</td>
<td>41</td>	
<td>2010/02/12</td>	
<td>\$109,850</td>	
</tr>	
<tr>	
<td>Vivian	Harrell</td>
<td>Financial	Controller</td>
<td>San	Francisco</td>
<td>62</td>	
<td>2009/02/14</td>	
<td>\$452,500</td>	
</tr>	
<tr>	
<td>Timothy	Mooney</td>
<td>Office	Manager</td>
<td>London</td>	
<td>37</td>	
<td>2008/12/11</td>	
<td>\$136,200</td>	
</tr>	
<tr>	
<td>Jackson	Bradshaw</td>
<td>Director</td>	

<td>New		York</td>
<td>65</td>		
<td>2008/09/26</td>		
<td>\$645,750</td>		
</tr>		
<tr>		
<td>Olivia		Liang</td>
<td>Support		Engineer</td>
<td>Singapore</td>		
<td>64</td>		
<td>2011/02/03</td>		
<td>\$234,500</td>		
</tr>		
<tr>		
<td>Bruno		Nash</td>
<td>Software		Engineer</td>
<td>London</td>		
<td>38</td>		
<td>2011/05/03</td>		
<td>\$163,500</td>		
</tr>		
<tr>		
<td>Sakura		Yamamoto</td>
<td>Support		Engineer</td>
<td>Tokyo</td>		
<td>37</td>		
<td>2009/08/19</td>		
<td>\$139,575</td>		
</tr>		
<tr>		
<td>Thor		Walton</td>

<td>Developer</td>	
<td>New	York</td>
<td>61</td>	
<td>2013/08/11</td>	
<td>\$98,540</td>	
</tr>	
<tr>	
<td>Finn	Camacho</td>
<td>Support	Engineer</td>
<td>San	Francisco</td>
<td>47</td>	
<td>2009/07/07</td>	
<td>\$87,500</td>	
</tr>	
<tr>	
<td>Serge	Baldwin</td>
<td>Data	Coordinator</td>
<td>Singapore</td>	
<td>64</td>	
<td>2012/04/09</td>	
<td>\$138,575</td>	
</tr>	
<tr>	
<td>Zenaida	Frank</td>
<td>Software	Engineer</td>
<td>New	York</td>
<td>63</td>	
<td>2010/01/04</td>	
<td>\$125,250</td>	
</tr>	
<tr>	

<td>Zorita	Serrano</td>
<td>Software	Engineer</td>
<td>San	Francisco</td>
<td>56</td>	
<td>2012/06/01</td>	
<td>\$115,000</td>	
</tr>	
<tr>	
<td>Jennifer	Acosta</td>
<td>Junior	Javascript
<td>Edinburgh</td>	Developer</td>
<td>43</td>	
<td>2013/02/01</td>	
<td>\$75,650</td>	
</tr>	
<tr>	
<td>Cara	Stevens</td>
<td>Sales	Assistant</td>
<td>New	York</td>
<td>46</td>	
<td>2011/12/06</td>	
<td>\$145,600</td>	
</tr>	
<tr>	
<td>Hermione	Butler</td>
<td>Regional	Director</td>
<td>London</td>	
<td>47</td>	
<td>2011/03/21</td>	
<td>\$356,250</td>	
</tr>	

<td><td>Lael</td> <td>Greer</td></td>	<td>Lael	Greer</td>
<td><td>Systems</td> <td>Administrator</td></td>	<td>Systems	Administrator</td>
<td><td>London</td></td> <td></td>	<td>London</td>	
<td><td>21</td></td> <td></td>	<td>21</td>	
<td><td>2009/02/27</td></td> <td></td>	<td>2009/02/27</td>	
<td><td>\$103,500</td></td> <td></td>	<td>\$103,500</td>	
</tr>		
<tr>		
<td><td>Jonas</td> <td>Alexander</td></td>	<td>Jonas	Alexander</td>
<td><td>Developer</td></td> <td></td>	<td>Developer</td>	
<td><td>San</td> <td>Francisco</td></td>	<td>San	Francisco</td>
<td><td>30</td></td> <td></td>	<td>30</td>	
<td><td>2010/07/14</td></td> <td></td>	<td>2010/07/14</td>	
<td><td>\$86,500</td></td> <td></td>	<td>\$86,500</td>	
</tr>		
<tr>		
<td><td>Shad</td> <td>Decker</td></td>	<td>Shad	Decker</td>
<td><td>Regional</td> <td>Director</td></td>	<td>Regional	Director</td>
<td><td>Edinburgh</td></td> <td></td>	<td>Edinburgh</td>	
<td><td>51</td></td> <td></td>	<td>51</td>	
<td><td>2008/11/13</td></td> <td></td>	<td>2008/11/13</td>	
<td><td>\$183,000</td></td> <td></td>	<td>\$183,000</td>	
</tr>		
<tr>		
<td><td>Michael</td> <td>Bruce</td></td>	<td>Michael	Bruce</td>
<td><td>Javascript</td> <td>Developer</td></td>	<td>Javascript	Developer</td>
<td><td>Singapore</td></td> <td></td>	<td>Singapore</td>	
<td><td>29</td></td> <td></td>	<td>29</td>	
<td><td>2011/06/27</td></td> <td></td>	<td>2011/06/27</td>	
<td><td>\$183,000</td></td> <td></td>	<td>\$183,000</td>	

```

        </tr>
        <tr>
            <td>Donna
            <td>Customer
            <td>New
            <td>27</td>
            <td>2011/01/25</td>
            <td>$112,000</td>
        </tr>
    </tbody>
</table>
</div>
</div>
</div>

</div>
<!-- .container-fluid -->

</div>
<!-- End of Main Content -->

<!-- Footer -->
<footer class="sticky-footer bg-white">
    <div class="container my-auto">
        <div class="copyright text-center my-auto">
            <span>Copyright © Your Website 2020</span>
        </div>
    </div>
</footer>
<!-- End of Footer -->

```

```

</div>
<!-- End of Content Wrapper -->

</div>
<!-- End of Page Wrapper -->

<!-- Scroll to Top Button-->
<a class="scroll-to-top rounded" href="#page-top">
    <i class="fas fa-angle-up"></i>
</a>

<!-- Logout Modal-->
<div class="modal fade" id="logoutModal" tabindex="-1" role="dialog" aria-labelledby="exampleModalLabel"
     aria-hidden="true">
    <div class="modal-dialog" role="document">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-title" id="exampleModalLabel">Ready to Leave?</h5>
                <button class="close" type="button" data-dismiss="modal" aria-label="Close">
                    <span aria-hidden="true">×</span>
                </button>
            </div>
            <div class="modal-body">Select "Logout" below if you are ready to end your current session.</div>
            <div class="modal-footer">
                <button class="btn btn-secondary" type="button" data-dismiss="modal">Cancel</button>
                <a class="btn btn-primary" href="login.html">Logout</a>
            </div>
    </div>
</div>

```

```

</div>
</div>

<!-- Bootstrap core JavaScript -->
<script src="vendor/jquery/jquery.min.js"></script>
<script src="vendor/bootstrap/js/bootstrap.bundle.min.js"></script>

<!-- Core plugin JavaScript -->
<script src="vendor/jquery-easing/jquery.easing.min.js"></script>

<!-- Custom scripts for all pages -->
<script src="js/sb-admin-2.min.js"></script>

<!-- Page level plugins -->
<script src="vendor/chart.js/Chart.min.js"></script>

<!-- Page level custom scripts -->
<script src="js/demo/chart-area-demo.js"></script>
<script src="js/demo/chart-pie-demo.js"></script>

<script>
  $('#myTable').dataTable(
    {
      ajax: '/api/data',
      scrollY: 200,
      deferRender: true,
      scroller: true
    });
</script>

</body>

```

```

</html>

{%
    extends      'dashboard/base.html'
%}

{%
    block       body
%}

<div class="container">
    {{ error }}
    <form   enctype="multipart/form-data"   method="post"   action="{% url
'shop_profile:shop_profile_create' %}>
        {% csrf_token %}
        Shop registered name: <br><input type="text"
name="shop_registered_name"><br>
            shop_name: <br><input type="text" name="shop_name"><br>
            shop_user_name:<br> <input type="text" name="shop_user_name"><br>
            shop_owner_name: <br><input type="text" name="shop_owner_name"><br>
            shop_address:<br> <input type="text" name="shop_address"><br>
            shop_warehouse_address:<br> <input type="text"
name="shop_warehouse_address"><br>
            shop_returned_address:<br> <input type="text"
name="shop_returned_address"><br>
            shop_category: <br><input type="text" name="shop_category"><br>
            shop_detail_description:<br> <input type="text"
name="shop_detail_description"><br>
            shop_pan_card: <br><input type="file" name="shop_pan_card"
accept="image/*"><br>
            shop_phone_number:<br> <input type="number"
name="shop_phone_number"><br>
            shop_owner_phone_number: <br><input type="number"
name="shop_owner_phone_number"><br><br>
            <input type="submit" value="submit" class="btn btn-success">
    </form>

```

```

</div>

{% endblock %}

{% extends 'dashboard/base.html' %}

{% block body %}



# Add Shop



###### Shop Profile



| Shop Name | User Name | Category   | Verification   |
|-----------|-----------|------------|----------------|
| Shop 1    | User 1    | Category 1 | Verification 1 |
| Shop 2    | User 2    | Category 2 | Verification 2 |
| Shop 3    | User 3    | Category 3 | Verification 3 |
| Shop 4    | User 4    | Category 4 | Verification 4 |


```

```

<th>Action</th>
</tr>
</thead>
<tfoot>

<tr>
    <th>Shop Name</th>
    <th>Shop User Name</th>
    <th>Shop Category</th>
    <th>Shop Verification</th>
    <th>Action</th>
</tr>
</tfoot>
<tbody>
    {%
        for profile in shop_profile %}
            <tr>
                <td>{{ profile.shop_name }}</td>
                <td>{{ profile.shop_user_name }}</td>
                <td>{{ profile.shop_category }}</td>
                <td>{{ profile.is_verified }}</td>
                <td><a href="#">view Details</a></td>
            </tr>
    {% endfor %}
</tbody>
</table>
</div>
</div>
</div>

```

```

</div>
<!-- .container-fluid -->

{% endblock %}

{% extends 'dashboard/base.html' %}

{% block body %}



# User created successfully


{{ errors }}
<form enctype="multipart/form-data" method="post" action="{% url 'seller_profile:seller_profile_create' %}>
    {% csrf_token %}
    first_name:<br><input type="text" name="first_name"><br>
    last_name:<br><input type="text" name="last_name"><br>
    mobile_number:<br><input type="number" minlength="10" min="9800000000" max="9999999999" name="mobile_number"><br>
    citizenship_number:<br><input type="text" name="citizenship_number"><br>
    citizenship_front:<br><input type="file" name="citizenship_front" accept="image/*"><br>
    citizenship_back:<br><input type="file" name="citizenship_back" accept="image/*"><br>
    temp_add:<br><input type="text" name="temp_add"><br>
    per_add:<br><input type="text" name="per_add"><br><br>
    <input type="submit" value="Create Profile" class="btn btn-primary">
</form>

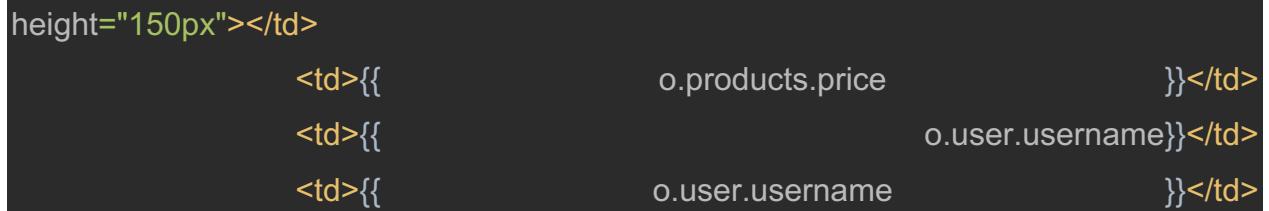
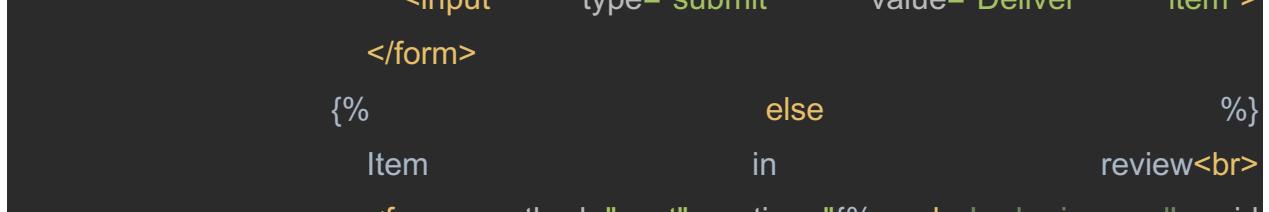
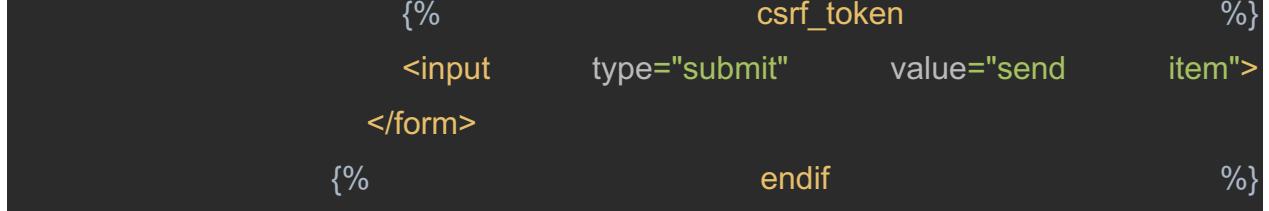

{% endblock %}

{% extends 'dashboard/base.html' %}

{% block body %}
```

```
<!-- Begin Page Content -->
<div class="container-fluid">

<!-- DataTales Example -->
<div class="card shadow mb-4">
    <div class="card-header py-3">
        <h6 class="m-0 font-weight-bold text-primary">My Order Details</h6>
    </div>
    <div class="card-body">
        <div class="table-responsive">
            <table class="table table-bordered" id="dataTable" width="100%" cellspacing="0">
                <thead>
                    <tr>
                        <th>Product Name</th>
                        <th>Product Photo</th>
                        <th>Product Price</th>
                        <th>Customer Name</th>
                        <th>Customer Address</th>
                        <th>Status</th>
                    </tr>
                </thead>
                <tfoot>
                    <tr>
                        <th>Product Name</th>
                        <th>Product Photo</th>
                        <th>Product Price</th>
                    </tr>
                </tfoot>
            </table>
        </div>
    </div>
</div>
```

Customer Name	Customer Address
Status	
	
	
	
	
	

```

<th>Customer Name</th>
<th>Customer Address</th>
<th>Status</th>
</tr>
</tfoot>
<tbody>
{%
    for o in order %
        <tr>
            <td>{{ o.products.product_name }}</td>
            <td></td>
            <td>{{ o.products.price }}</td>
            <td>{{ o.user.username }}</td>
            <td>{{ o.user.username }}</td>
            <td>
                {%
                    if o.is_delivered %
                        Delivered
                    elif o.is_send and not o.is_delivered %
                        <form method="post" action="{% url 'order:is_delivered' o.id %}">
                            {% csrf_token %}
                            <input type="submit" value="Deliver" item>
                        </form>
                    else %
                        Item in review<br>
                        <form method="post" action="{% url 'order:is_send' o.id %}">
                            {% csrf_token %}
                            <input type="submit" value="send" item>
                        </form>
                {%
                    endif %
                }
            
```

```

        </td>
    </tr>
    {%
        endfor
    %}

    </tbody>
</table>
</div>
</div>

</div>

{%
    extends 'customer/base.html'
    block body
%}



###### My Order Details


```

```

<thead>
<tr>
    <th>Product Name</th>
    <th>Product Photo</th>
    <th>Product Price</th>
    <th>Status</th>
</tr>
</thead>
<tfoot>

<tr>
    <th>Product Name</th>
    <th>Product Photo</th>
    <th>Product Price</th>
    <th>Status</th>
</tr>
</tfoot>
<tbody>
{%
    for o in order %}
        <tr>
            <td>{{ o.products.product_name }}</td>
            <td></td>
            <td>{{ o.products.price }}</td>
            <td>
                {% if o.is_delivered %}
                    Delivered
                {% elif o.is_send and not o.is_delivered %}
                    On the Way
                {% else %}
                    In review
                {%endif%}
            
```

```

        {%
            endif
        %}

    </td>
</tr>
{%
    endfor
%}

</tbody>
</table>
</div>
</div>

</div>

{%
    endblock
%}

<table>
<tr>

</tr>

</table>
{#<h1 align="right">Total:{{
{#    <form method="post" action="{% url 'order:checkout' %}">#
{#        {% csrf_token %}
{#        <button type="submit">Check Out</button>#
{#    </form>#

```

```

{#</h1>#}

</body>
</html>

{%
    extends      'customer/base.html'
%}
{%
    block      body
%}
<!-- Begin Page Content -->
<div class="container-fluid">

<!-- Page Heading -->
<h1 class="h3 mb-2 text-gray-800" style="text-align: right">
    <a href="{% url 'landing_page:landing_page' %}" class="btn btn-success mb-2">Add More item</a>
</h1>

<!-- DataTales Example -->
<div class="card shadow mb-4">
    <div class="card-header py-3">
        <h6 class="m-0 font-weight-bold text-primary">My Cart</h6>
    </div>
    <div class="card-body">
        <div class="table-responsive">
            <table class="table table-bordered" id="dataTable" width="100%" cellspacing="0">
                <thead>
                    <tr>
                        <th>Product Name</th>
                        <th>Product Photo</th>
                        <th>Product Price</th>
                        <th>Action</th>
                    </tr>
                </thead>

```

```

        </thead>
        <tfoot>

        <tr>
            <th>Product Name</th>
            <th>Product Photo</th>
            <th>Product Price</th>
            <th>Action</th>
        </tr>
        </tfoot>
        <tbody>
            {% for cart in cart_item %}
                <tr>
                    <td>{{ cart.products.product_name }}</td>
                    <td></td>
                    <td>{{ cart.products.price }}</td>
                    <td>
                        <form method="post" action="{% url 'order:remove_from_cart' cart.id %}">
                            {% csrf_token %}
                            <button type="submit">Remove From Cart</button>
                        </form>
                    </td>
                </tr>
            {% endfor %}

        </tbody>
    </table>
    <h1 align="right">Total:{{ total }}</h1>
    <form method="post" action="{% url 'order:checkout' %}">

```

```

        {%
          csrf_token
        %}
      <button type="submit">Check Out</button>
    </form>
  </h1>

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

</div>
{% endblock %}

{%
  load static
<!doctype html>
<html lang="en">
<head>
  <!-- Required meta tags -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

  <!-- Bootstrap CSS -->
  <link href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJISAwGgFAW/dAiS6JXm" crossorigin="anonymous">

  <title>
    {%
      block title
    %}
  </title>
%}

```

```

<style>
    {%
        block
        style
    }{%
        endblock
    }%
</style>
<style>

.search {
    margin-right: 250px;
}

.search_bar {
    width: 400px;
}

footer {
    background-color: #0F0F0F;
    height: 334px;
    text-align: center;
    color: white;
}

a {
    color: white;
}

</style>
</head>
<body>
<nav class="navbar navbar-expand-lg navbar-light bg-light">
    <a class="navbar-brand" href="#">
        
    </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="navbar-toggler-icon"></span>
</button>

<div class="collapse navbar-collapse" id="navbarSupportedContent">
    <ul class="navbar-nav mr-auto">
        <li class="nav-item active">
            &#{#
                <a class="nav-link" href="#">nothing <span class="sr-
only">(current)</span></a>#}
            </li>
        </ul>
        <form class="form-inline my-2 my-lg-0 search" action="{% url 'landing_page:search' %}">
            <div class="dropdown">
                <select name="search_type" class="btn btn-outline-success dropdown-
toggle">
                    <option value="product">All Product</option>
                    <option value="shop">Shop</option>
                </select>
            </div>
            &nbsp;&nbsp;
            <input class="form-control" name="search" type="search" placeholder="Search" aria-label="Search" style="width: 300px">&nbsp;&nbsp;
            <button class="btn btn-outline-success my-2 my-sm-0" type="submit">Search</button>
        </form>
        &nbsp;&nbsp;&nbsp;
    </div>

```

```

{%
    if user.is_authenticated
    %}

        <a href="{% url 'shop_profile:shop_profile_show' %}" class="btn btn-outline-success my-2 my-sm-0"
            type="submit">Shop Profile</a>

        <a href="{% url 'accounts:logout' %}" class="btn btn-outline-success my-2 my-sm-0"
            type="submit">Logout</a>

{%
    else
    %}

        &nbsp;&nbsp;&nbsp;
        <a href="{% url 'accounts:login' %}" class="btn btn-outline-success my-2 my-sm-0"
            type="submit">Login</a>
        &nbsp;&nbsp;&nbsp;
        <a href="{% url 'accounts:customer_register' %}" class="btn btn-outline-success my-2 my-sm-0"
            type="submit">Register</a>

{%
    endif
    %}

</div>
</nav>
{%
    block body
    %}

{%
    endblock
    %}

<!--
<footer class="page-footer font-small lighten-3 pt-4">

    <!--
        Footer
        -->
    <!--
        Footer
        Links
        -->
-->
```

```
<div class="container-fluid" style="text-align: center; margin-left: auto; margin-right: auto; background-color: #0F0F0F">

    <!-- Grid Row -->
    <div class="row" style="background-color: #0F0F0F">

        <!-- Grid Column -->
        <div class="col-md-4 col-lg-3 mr-auto my-md-4 my-0 mt-4 mb-1">

            <!-- Content -->
            

        </div>

        <!-- Grid Column -->
        <div class="col-md-4 col-lg-3 mx-auto my-md-4 my-0 mt-4 mb-1" style="text-align: right; border-top: 1px solid black; padding-top: 10px">

            <hr class="clearfix w-100 d-md-none" style="border: none; border-top: 1px solid black; margin-bottom: 10px; width: 100%; height: 0; position: relative; overflow: visible; z-index: 1; margin-bottom: 10px; width: 100%; height: 0; position: relative; overflow: visible; z-index: 1;">

            <!-- Grid Column -->
            <div class="col-md-2 col-lg-2 mx-auto my-md-4 my-0 mt-4 mb-1" style="text-align: center; width: 100%; height: 0; position: relative; overflow: visible; z-index: 1; width: 100%; height: 0; position: relative; overflow: visible; z-index: 1;">

                <!-- Links -->
                <h5 class="font-weight-bold text-uppercase mb-4" style="margin-bottom: 0; font-size: 1.2em; color: #007bff; text-decoration: none; text-align: center; width: 100%; height: 0; position: relative; overflow: visible; z-index: 1; width: 100%; height: 0; position: relative; overflow: visible; z-index: 1;">Contact Us</h5>

                <ul style="list-style-type: none; padding-left: 0; margin: 0; width: 100%; height: 0; position: relative; overflow: visible; z-index: 1; width: 100%; height: 0; position: relative; overflow: visible; z-index: 1;" class="list-unstyled" style="width: 100%; height: 0; position: relative; overflow: visible; z-index: 1; width: 100%; height: 0; position: relative; overflow: visible; z-index: 1;">

                    <li>
                        <p style="margin: 0; font-size: 0.9em; color: #007bff; text-decoration: none; width: 100%; height: 0; position: relative; overflow: visible; z-index: 1; width: 100%; height: 0; position: relative; overflow: visible; z-index: 1;">9808282207</p>
                    </li>
                    <li>
                        <p style="margin: 0; font-size: 0.9em; color: #007bff; text-decoration: none; width: 100%; height: 0; position: relative; overflow: visible; z-index: 1; width: 100%; height: 0; position: relative; overflow: visible; z-index: 1;"></p>
                    </li>
                    <li>
                        <p style="margin: 0; font-size: 0.9em; color: #007bff; text-decoration: none; width: 100%; height: 0; position: relative; overflow: visible; z-index: 1; width: 100%; height: 0; position: relative; overflow: visible; z-index: 1;"></p>
                    </li>
                </ul>
            </div>
        </div>
    </div>
</div>
```

```
          samancha@gmail.com
      </p>
</li>
<li>
    <p>
        kathmandu,Nepal
    </p>
</li>

</ul>

</div>
<!-- Grid column -->
<!-- Grid column -->

<hr class="clearfix w-100 d-md-none">

<!-- Grid column -->
<div class="col-md-2 col-lg-2 mx-auto my-md-4 my-0 mt-4 mb-1">

<!-- Links -->
<h5 class="font-weight-bold text-uppercase mb-4">Our Company</h5>

<ul class="list-unstyled">
    <li>
        <p>
            <a href="#">About Us</a>
        </p>
    </li>
    <li>
        <p>
```

```

        <a href="#!">Pricing</a>
      </p>
    </li>
    <li>
      <p>
        <a href="#!">Stories</a>
      </p>
    </li>

  </ul>

</div>
<!-- Grid column --><!-- Grid column -->

<hr class="clearfix w-100 d-md-none">

<!-- Grid column -->
<div class="col-md-2 col-lg-2 mx-auto my-md-4 my-0 mt-4 mb-1">

  <!-- Links -->
  <h5 class="font-weight-bold text-uppercase mb-4">Company Policy</h5>

  <ul class="list-unstyled">
    <li>
      <p>
        <a href="#!">Privacy Policy</a>
      </p>
    </li>
    <li>
      <p>
        <a href="#!">Terms and Conditions</a>
      </p>
    </li>
  </ul>
</div>

```

```
</p>
</li>

</ul>

</div>
<!-- Grid column -->
<!-- Grid column -->

<hr class="clearfix w-100 d-md-none">

<!-- Grid column -->
<div class="col-md-2 col-lg-2 mx-auto my-md-4 my-0 mt-4 mb-1">

    <!-- Links -->
    <h5 class="font-weight-bold text-uppercase mb-4">Social Contacts</h5>

    <ul class="list-unstyled">
        <li>
            <p>
                <a href="#">Facebook</a>
            </p>
        </li>
        <li>
            <p>
                <a href="#">Instagram</a>
            </p>
        </li>
        <li>
            <p>
                <a href="#">Twitter</a>
            </p>
        </li>
    </ul>
</div>
```

```

        </p>
        </li>

    </ul>

</div>
<!-- Grid column -->

</div>
<!-- Grid row -->

</div>
<!-- Footer Links -->

<!-- Copyright -->
<div class="footer-copyright text-center py-3" style="background-color: #0F0F0F">©
2020 Copyright:
samancha.com</a>
</div>
<!-- Copyright -->

</footer>
<!-- Footer -->
{%
    block script
    %}{{% endblock %}
<!-- Optional JavaScript -->
<!-- jQuery first, then Popper.js, then Bootstrap JS -->
<script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"
integrity="sha384-KJ3o2DKtlkvYIK3UEzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN"
crossorigin="anonymous"></script>

```

```

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"
        integrity="sha384-
ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q"
        crossorigin="anonymous"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"
        integrity="sha384-
JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYI"
        crossorigin="anonymous"></script>

</body>
</html>

{%
    extends 'landing_page/base.html'
%}

{%
    block title
%}

Saman
Cha

{%
    endblock
%}

{%
    block body
%}

<div class="container-fluid mt-3 mb-2">
    <div style="text-align: right">
        <a href="{% url 'order:view_cart' %}" class="btn btn-outline-success my-2 my-sm-0">View
            cart</a>
    </div>
    <h2 class="mb-5" style="text-align: center; border-bottom: 2px solid #4FA846;">Products</h2>
    <div class="row">
        {%
            for product in products
        %}
            <div class="card ml-5 mb-5" style="width: 18rem; box-shadow: 10px 10px 5px grey;">
                {#
                    <a href="{% url 'ecom:product' product.slug|slugify %}">#}
                    
                {#}
            </div>
        {%
            endfor
        %}
    </div>
</div>

```

```

        alt="Card image cap" style="height: 200px; object-fit: cover">
    </a>
    <div class="card-body">
        {#
            <a href="{% url 'ecom:product' product.slug|slugify %}" style="color: black">#}
                {#
                    <h4 class="card-title">{{ product.name }}</h4>#}
                {#
                    </a>#}
                <h5 class="card-title">{{ product.product_name }}</h5>
                <h5 class="card-title">RS.{{ product.price }}</h5>
                <a href="{% url 'order:add_to_cart' product.slug|slugify %}" class="btn btn-success">Add to card</a>
            </div>
        </div>
        {%
            endfor %}
    </div>
</div>
{% endblock %}

{% extends "landing_page/base.html" %}

{% block body %}
    <div class="container m-5">
        {%
            for product in search_result %}
            <!-- Stack the columns on mobile by making one full-width and the other half-width -->
            <div class="row mb-5" style="border-bottom: 2px solid black">
                <div class="col-6 col-md-4">
                    <div class="card mb-4" style="width: 18rem; box-shadow: 10px 10px 5px grey;">
                        
                    </div>
                </div>
            </div>
        {% endfor %}
    </div>
</block>

```

```

        alt="Card image cap" style="height: 300px; width: 200px; object-fit: cover">
    </div>
</div>
<div class="col-12 col-md-8" style="border-left: 2px solid black">
    <h1 class="mt-5">{{ product.product_name }}</h1>
    <h3>Rs. {{ product.price }}</h3>
    <h4>{{ product.brand_name }}</h4>
    <a href="#" class="btn btn-success">View Details</a>
</div>

</div>

{%
    for product in products
%}
</div>
{% endblock %}

{% extends 'landing_page/base.html' %}
{% block title %}{{ shop_name }}{% endblock %}
{% block body %}

<div class="container-fluid mt-5 mb-5">
    <div style="text-align: right">
        <a href="{% url 'order:view_cart' %}" class="btn btn-outline-success my-2 my-sm-0">View cart</a>
    </div>
    <h2 class="mb-5" style="text-align: center; border-bottom: 2px solid #3F8939;">Products</h2>
    <div class="row">
        {% for product in products %}
            <div class="card ml-5 mb-5" style="width: 18rem; box-shadow: 10px 10px 5px grey;">

```

```

{# <a href="#"><% url 'ecom:product' product.slug|slugify %>#}

</a>
<div class="card-body">
{# <a href="#"><% url 'ecom:product' product.slug|slugify %>#}
style="color: black">#}
{#
<h4 class="card-title">>{{ product.name }}</h4>#
{#
<h5 class="card-title">>{{ product.product_name }}</h5>
<h5 class="card-title">>RS.{{ product.price }}</h5>
<a href="#"><% url 'order:add\_to\_cart' product.slug|slugify %># class="btn btn-success">Add to
card</a>
</div>
</div>
{%
endfor %}
</div>

</div>
<div class="m-5" style="text-align: right">
<a href="#"><% url 'chat:send\_message' username %># class="btn btn-success mb-5">Send Message</a>
</div>

{%
endblock %}
{%
extends 'dashboard/base.html' %}
{%
block body %}

```

```

<!-- Begin Page Content -->
<div class="container-fluid row">

    <div class="card shadow mb-4 col-2">
        <div class="card-header py-3">
            <h6 class="m-0 font-weight-bold text-primary">Messages</h6>
        <div>
            {% for messages in all_messages %}
                <div class="alert alert-secondary">
                    {{ messages.send_from }}
                </div>
            {% endfor %}
        </div>
        <div class="card shadow mb-4 ml-3 col-9">
            <div class="card-header py-3">
                <h6 class="m-0 font-weight-bold text-primary">{{ username }}</h6>
            </div>
        <div class="container mt-2">
            <div class="row">
                {% for chat in chats %}
                    {% if chat.message %}
                        {% if request.user == chat.send_from %}
                            <div class="col-6" style="text-align: right">
                                </div>
                            <div class="col-6 alert alert-success" style="text-align: right">
                                <p class="alert alert-dark">{{ chat.send_from }}</p>
                                <p>{{ chat.message }}</p>
                            </div>
                        {% endif %}
                    {% endif %}
                {% endfor %}
            </div>
        </div>
    </div>
</div>

```

```

        {%
            if      chat.image      !=  'False'   %
                
            {%
                endif
            %
        </div>

        {%
            else
            %
                <div  class="col-6  alert  alert-primary"  style="text-align:  left">
                    <div  class="alert  alert-dark">{{    chat.send_from    }}</div>

                    <p>{{          chat.message      }}</p>

                </div>
                <div  class="col-6"      "      style="text-align:  left">

                </div>

            {%
                endif
            %
            {%
                endif
            %
        </div>
    </div>
<div                                         class="container">
    <form  enctype="multipart/form-data"      action=""      method="post">
        {%
            csrf_token
        %
        <div
            <div
                <div
                    <label      for="images">Select      Images</label>
                    <input type="file" class="filestyle" name="images" id="images" data-
input="false"
                        data-iconName="fa fa-upload" data-buttonText="Upload File"/>

```

```
</div>
</div>
<div class="col-7">
    <div class="form-group">
        <label for="message">Messages</label>
        <input type="text" class="form-control" id="message" aria-describedby="message"
               placeholder="Enter message" name="message">
    </div>
</div>
<div class="col-2">
    <div class="form-group">
        <label for="send">.</label>
        <input type="submit" class="form-control" id="send" value="Send">
    </div>
</div>

</form>
</div>
</div>
</div>
{%- endblock %}
```

7.3.2 SAMPLE CODE FOR THE AUTOMATION SCRIPT

```

from django.contrib.auth.decorators import login_required
from django.shortcuts import render, HttpResponseRedirect
from django.db.models import Q, Count

from accounts.models import MyUser
from .forms import ChatForms
from .models import Chat

# Create your views here.
from shop_profile.models import ShopProfile


@login_required()
def send_messages(request, username):
    if not MyUser.objects.filter(username=username).exists():
        return HttpResponseRedirect('no user exist')
    send_to = MyUser.objects.get(username=username)

    chats = Chat.objects.filter(
        Q(
            Q(send_from=request.user) & Q(send_to=send_to)) |
        Q(
            Q(send_from=send_to) & Q(send_to=request.user))
    ).order_by('send_date')
    all_message = Chat.objects.filter(
        Q(send_to=request.user)
    ).order_by('-send_date')

    if request.method == 'POST':
        message = request.POST.get('message')

```

```

images = request.FILES['images'] if 'images' in request.FILES else False
if not message and not images:
    return JsonResponse('nothing')
forms = ChatForms(request.POST, request.FILES)
if forms.is_valid():
    chat = Chat(
        send_from=request.user,
        send_to=send_to,
        message=message,
        image=images,
    )
    chat.save()

context = {"username": username, "chats": chats, "all_messages": all_message}
return render(request, 'chat/home.html', context)

from django.shortcuts import render, HttpResponseRedirect
from products.models import Products

# Create your views here.
from shop_profile.models import ShopProfile


def landing_page(request):
    products = Products.objects.all().order_by('-created_date')
    context = {"products": products}
    return render(request, 'landing_page/landing_page.html', context)

def shop_landing_page(request, username):
    if not ShopProfile.objects.filter/shop_user_name=username.exists():
        return HttpResponseRedirect('Shop Profile Doesnot Exist')

```

```

products = Products.objects.filter(shop_product__shop_user_name=username)
context = {"products": products, "username": username}
return render(request, 'landing_page/shop_landing_page.html', context)

def search(request):
    search_type = request.GET.get('search_type')
    search_key = request.GET.get('search')
    context = {}
    if search_type == 'shop':
        search_result = ShopProfile.objects.filter(shop_user_name=search_key).order_by('-id')[:10]
        context['search_result'] = search_result
    elif search_type == 'product':
        search_result = Products.objects.filter(product_name=search_key).order_by('-id')[:10]
        context['search_result'] = search_result
    print(context)
    return render(request, 'landing_page/search.html', context)

def test(request):
    return render(request, 'dashboard/home.html')

from django.contrib.auth.decorators import login_required
from django.core.exceptions import PermissionDenied
from django.db.models import Q
from django.shortcuts import render, redirect, HttpResponseRedirect
from products.models import Products
from shop_profile.models import ShopProfile
from .forms import OrderForm
from .models import Order

```

```

from ecommerce.decorators import is_seller
from django.http import HttpResponseRedirect

# Create your views here.
from .models import Order

@login_required
def addtocart(request, slug):
    user = request.user
    products = Products.objects.get(slug=slug)
    form_data = {"products": products.id, 'user': user}
    form = OrderForm(form_data)
    if form.is_valid():
        form.save()
    return HttpResponseRedirect('/')

else:
    return HttpResponseRedirect("error")



@login_required
def view_cart(request):
    cart_item = Order.objects.filter(user=request.user).filter(is_bought=False)
    print(cart_item)
    total = 0
    for cart in cart_item:
        total += int(cart.products.price)
    print(total)
    context = {'cart_item': cart_item, 'total': total}
    return render(request, 'order/cart.html', context)

```

```

@login_required()
def remove_from_cart(request, card_id):
    if request.method == "POST":
        Order.objects.filter(id=card_id).delete()
        return redirect('order:view_cart')
    return redirect('/')

@login_required()
def checkout(request):
    if request.method == "POST":
        Order.objects.filter(user=request.user).update(is_bought=True)
        return redirect('landing_page:landing_page')

@login_required()
def order_details(request):
    order = Order.objects.filter(user=request.user)
    context = {"order": order}
    return render(request, 'order/order_details.html', context)

@login_required()
@is_seller()
#
# @is_order_owner()
def shop_order_details(request, username):
    if not ShopProfile.objects.filter(owner=request.user).filter(shop_user_name=username).exists():
        raise PermissionDenied

```

```

order = Order.objects.filter(products__shop_product__shop_user_name=username)
context = {"order": order, "username": username}
return render(request, 'order/shop_order_details.html', context)

@login_required()
@is_seller()
def is_send(request, id, username):
    if request.method == 'POST':
        Order.objects.filter(id=id).update(is_send=True)
    return HttpResponseRedirect(f'order/shop_order_details/{username}')
}

@login_required()
@is_seller()
def is_delivered(request, id, username):
    if request.method == 'POST':
        Order.objects.filter(id=id).update(is_delivered=True)
    return HttpResponseRedirect(f'order/shop_order_details/{username}')


from django.contrib.auth.decorators import login_required
from django.shortcuts import render, redirect
from .forms import ProductsForm
from .models import Products
from shop_profile.models import ShopProfile
from ecommerce.decorators import is_seller
from django.core.exceptions import PermissionDenied

# Create your views here.

@login_required

```

```

@is_seller()
def product_create(request, username):
    if not ShopProfile.objects.filter(owner=request.user).filter(shop_user_name=username).exists():
        raise PermissionDenied
    if request.method == 'POST' and not ShopProfile.objects.filter(owner=request.user).filter(shop_user_name=username).exists():
        raise PermissionDenied

    forms = ProductsForm(request.POST, request.FILES)
    if forms.is_valid():
        product = Products(
            shop_product=ShopProfile.objects.get(shop_user_name=username),
            owner=request.user,
            product_name=request.POST.get('product_name'),
            price=request.POST.get('price'),
            photo=request.FILES.get('photo'),
            category=request.POST.get('category'),
            sub_category=request.POST.get('sub_category'),
            brand_name=request.POST.get('brand_name'),
            description=request.POST.get('description'),
            stock=request.POST.get('stock'),
        )
        product.save()
else:
    print(forms.errors)
return render(request, 'product/home.html', {"username": username})

```

```

@login_required()
@is_seller()
def show_product(request, username):
    if not ShopProfile.objects.filter(owner=request.user).filter(shop_user_name=username).exists():
        raise PermissionDenied
    product = Products.objects.filter(shop_product__shop_user_name=username)
    context = {"products": product, "username": username}
    return render(request, 'product/show_product.html', context)

from django.contrib.auth.decorators import login_required
from django.shortcuts import render, redirect
from .forms import SellerProfileForm
from .models import SellerProfile
from ecommerce.decorators import is_seller

# Create your views here.

@login_required()
@is_seller()
def seller_profile_create(request):
    if SellerProfile.objects.filter(owner=request.user).exists():
        return redirect('seller_profile:seller_profile_view')
    if request.method == 'POST':
        forms = SellerProfileForm(request.POST, request.FILES)
        if forms.is_valid():
            seller_profile = SellerProfile(
                owner=request.user,
                first_name=request.POST['first_name'],
                last_name=request.POST['last_name'],
            )
            seller_profile.save()
            return redirect('seller_profile:seller_profile_view')
    else:
        forms = SellerProfileForm()
    return render(request, 'seller_profile/seller_profile_form.html', {'form': forms})

```

```

        mobile_number=request.POST['mobile_number'],
        citizenship_number=request.POST['citizenship_number'],
        temp_add=request.POST['temp_add'],
        per_add=request.POST['per_add'],
        citizenship_front=request.FILES['citizenship_front'],
        citizenship_back=request.FILES['citizenship_back'],
    )
    seller_profile.save()
    return redirect('shop_profile:shop_profile_show')
else:
    context = {'errors': forms.errors}
    return render(request, 'seller_profile/seller_profile_create.html', context)

return render(request, 'seller_profile/seller_profile_create.html')


@login_required()
@is_seller()
def seller_profile_view(request):
    if not SellerProfile.objects.filter(owner=request.user).exists():
        return redirect('seller_profile:seller_profile_create')
    profile = SellerProfile.objects.filter(owner=request.user)
    context = {"profile": profile}
    return render(request, 'seller_profile/seller_profile_show.html', context)

from django.contrib.auth.decorators import login_required
from django.shortcuts import render, redirect
from .forms import ShopProfileForm
from .models import ShopProfile
from ecommerce.decorators import is_seller

```

```

# Create your views here.

@login_required
@is_seller()

def shop_profile_create(request):
    if request.method == 'POST':
        forms = ShopProfileForm(request.POST, request.FILES)
        if forms.is_valid():
            shop_profile = ShopProfile(
                owner=request.user,
                shop_registered_name=request.POST.get('shop_registered_name'),
                shop_name=request.POST.get('shop_name'),
                shop_user_name=request.POST.get('shop_user_name'),
                shop_owner_name=request.POST.get('shop_owner_name'),
                shop_address=request.POST.get('shop_address'),
                shop_warehouse_address=request.POST.get('shop_warehouse_address'),
                shop_returned_address=request.POST.get('shop_returned_address'),
                shop_category=request.POST.get('shop_category'),
                shop_detail_description=request.POST.get('shop_detail_description'),
                shop_pan_card=request.FILES.get('shop_pan_card'),
                shop_phone_number=request.POST.get('shop_phone_number'),
                shop_owner_phone_number=request.POST.get('shop_owner_phone_number'),
            )
            shop_profile.save()
            return redirect('shop_profile:shop_profile_show')
    else:
        context = {"error": forms.errors}
        return render(request, 'shop_profile/shop_profile_create.html', context)
    return render(request, 'shop_profile/shop_profile_create.html')

```

```
@login_required  
@is_seller()  
def show_shop_profile(request):  
    my_shop_profile = ShopProfile.objects.filter(owner=request.user)  
    context = {"shop_profile": my_shop_profile}  
    return render(request, 'shop_profile/shop_profile_show.html', context)
```

7.4 APPENDIX D: DESIGNS

7.4.1 GANTT CHART

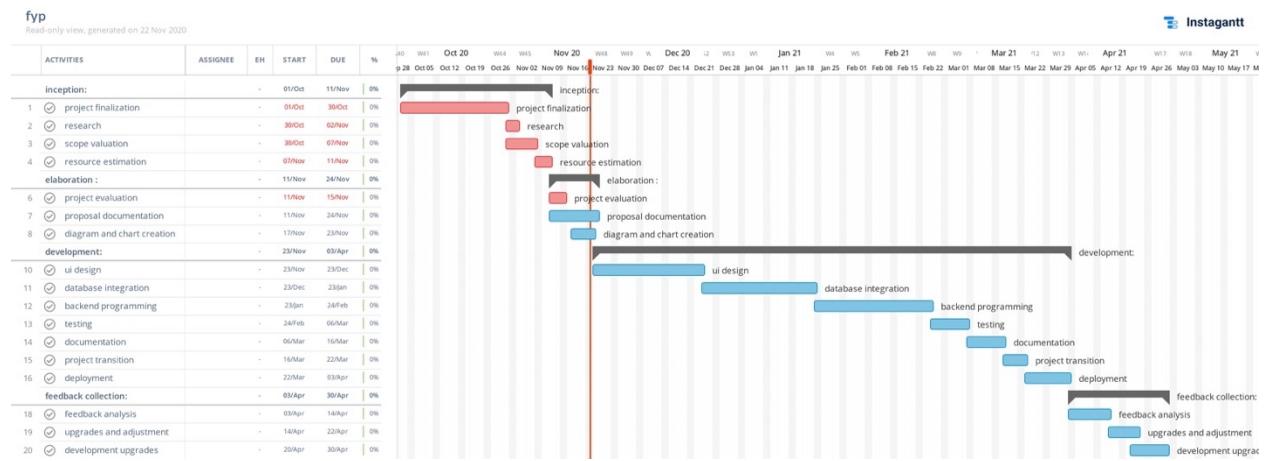


Figure 125 gantt chart

7.4.2 WORK BREAKDOWN STRUCTURE

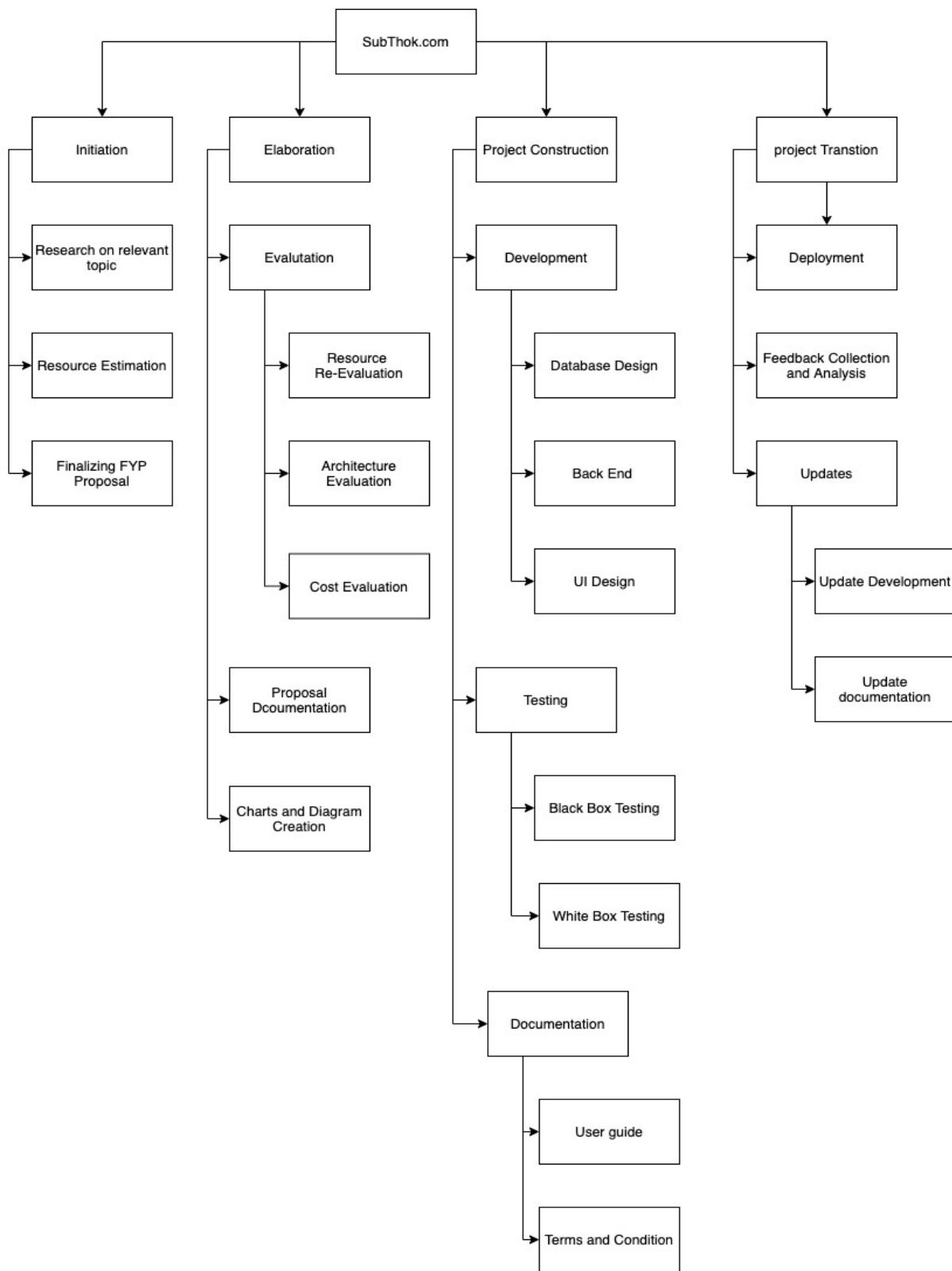


Figure 126 work breakdown

7.4.5 HARDWARE ARCHITECTURE

The architecture of the structure that this project will adopt is depicted in the diagram below. Model, View, and Template are the three key components of the software design pattern.

Model: A model is a tool that aids in the management of a database. MySQL is the database that will be used. The data will be stored in the MySQL database on the server. The presentation layer is made up of the prototype. All user interfaces are handled by this layer. This layer contains HTML and CSS.

The view layer is used to hold data and render templates, as well as to execute business logic and communicate with the model.

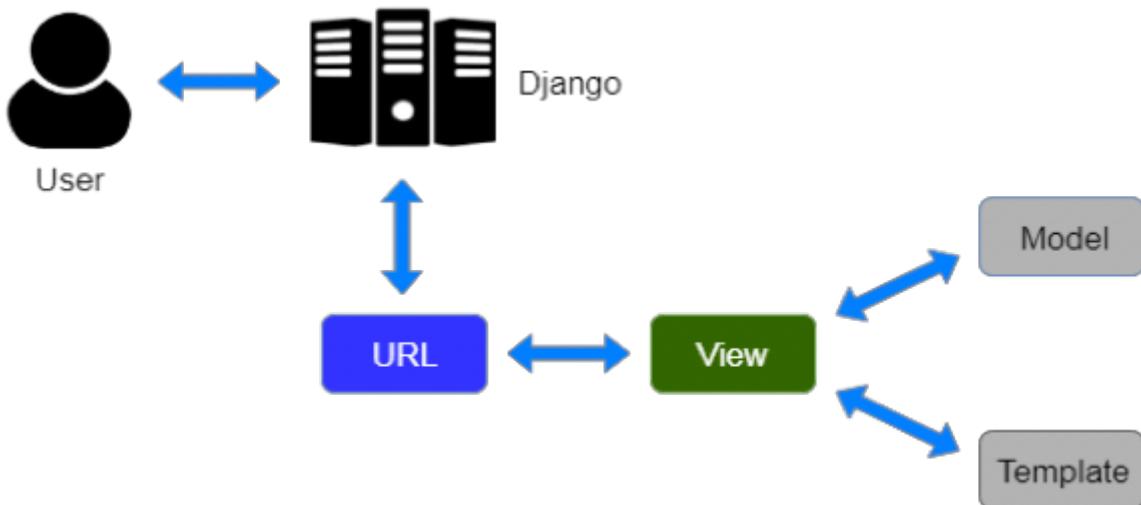


Figure 127 Hardware Architecture

7.4.6 DATA FLOW DIAGRAMS

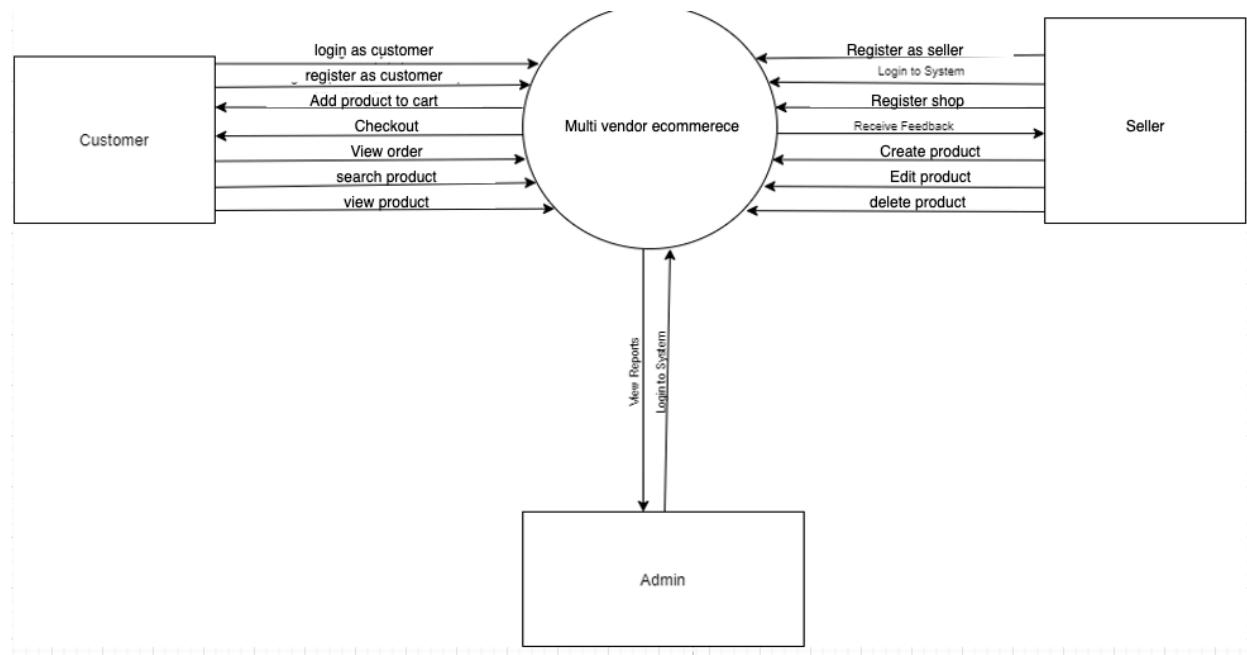


Figure 128 DFD

7.4.7 USE CASE

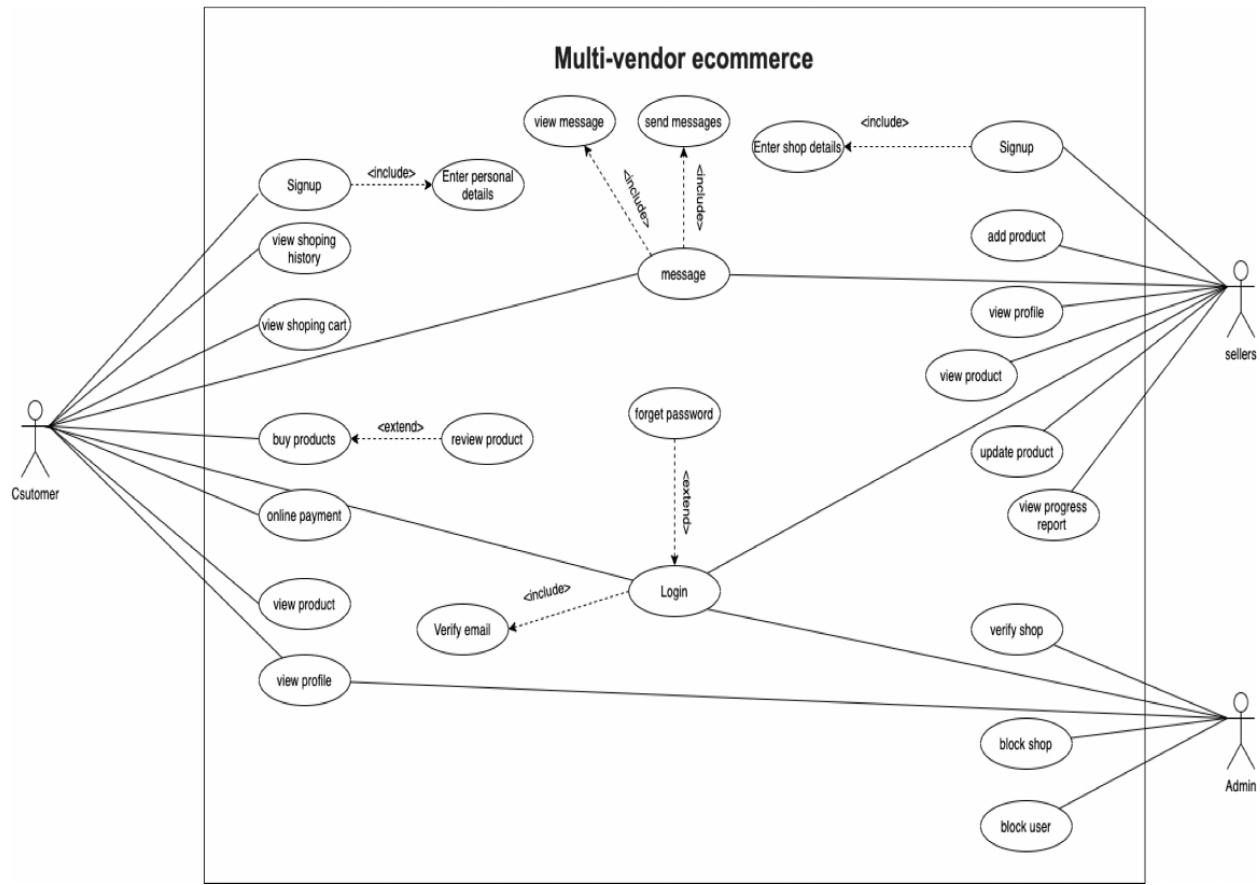


Figure 129 Use case

7.4.8 High Level Use Case of the System

1. Signup as customer into the system

Use Case: Customer signup

Actor: Customer

Descriptions: Customer need to sign up to access the system. Then customer needs to fill up their personal detail to use the system features

Table 33 High level use case 1

2. login into the system

Use Case: Login into the system

Actor: Customer, Seller, Admin

Descriptions: If customer wants to buy product then they have to login into the system.

For this they have to enter their email and password.

Table 34 High level use case 2

3. view products

Use Case: View product

Actor: Customer

Descriptions: Customer can view all product on the site.

Table 35 High level use case 3

4. buy products

Use Case: Buy product

Actor: Customer

Descriptions: customer can buy any product they like by paying money.

Table 36 High level use case 4

5. view shopping history

Use Case: View Shopping History

Actor: Customer

Descriptions: Customer can browse their shopping history like what they. have bought and when they have bought

Table 37 High level use case 5

6. view shopping cart

Use Case: View shopping cart

Actor: Customer

Descriptions: customer can view the things they had added in the shopping cart

Table 38 High level use case 6

7. Payment

Use Case: Payment

Actor: Customer

Descriptions: after adding items into cart or by clicking on the buy now option payment option can be seen and after filling the required fields people can choose to pay online or to pay cash on delivery.

Table 39 High level use case 7

8. message

Use Case: Message

Actor: Customer, Seller

Descriptions: customer can send message to seller and seller can receive message and respond to it.

Table 40 High level use case 8

9. view shop profile

Use Case: view profile

Actor: Seller

Descriptions: Seller can see their shop profile like shop name shop phone number and other details about their shop

Table 41 High level use case 9

10. signup as seller into the system

Use Case: signup

Actor: Seller

Descriptions: Seller need to sign up in order to use the services provided by the system.

To be eligible to sell thing on the system seller must fill their shop details and wait for the admin approval.

Table 42 High level use case 10

11. Add product

Use Case: Add product

Actor: Seller

Descriptions: Seller can add product to the shop which can be visible to the customer.

Table 43 High level use case 11

12. View shop product

Use Case: View product

Actor: Seller

Descriptions: Seller can view their product which they have added on the shop

Table 44 High level use case 12

13. Update product

Use Case: Update product

Actor: Seller

Descriptions: seller also can update the product details if anything went wrong or if they want to change something or add something on their product.

Table 45 High level use case 13

14. View shop progress report

Use Case: View progress report

Actor: Seller

Descriptions

Table 46 High level use case 14

15. Verify shop

Use Case: Verify shop

Actor: Admin

Descriptions: Admin have to verify shop by checking the details provided by the shop owner and have to verify shop either they are eligible to sell product in the market.

Table 47 High level use case 15

16. Block shop

Use Case: Block Shop

Actor: Admin

Descriptions: Admin have full authority to block shop if the shop owner causes any misconduct.

Table 48 High level use case 16

17. Block user

Use Case: Block user

Actor: Admin

Descriptions: Admin also have ability to block user for their misconduct.

Table 49 High level use case 17

7.4.8 WIREFRAME

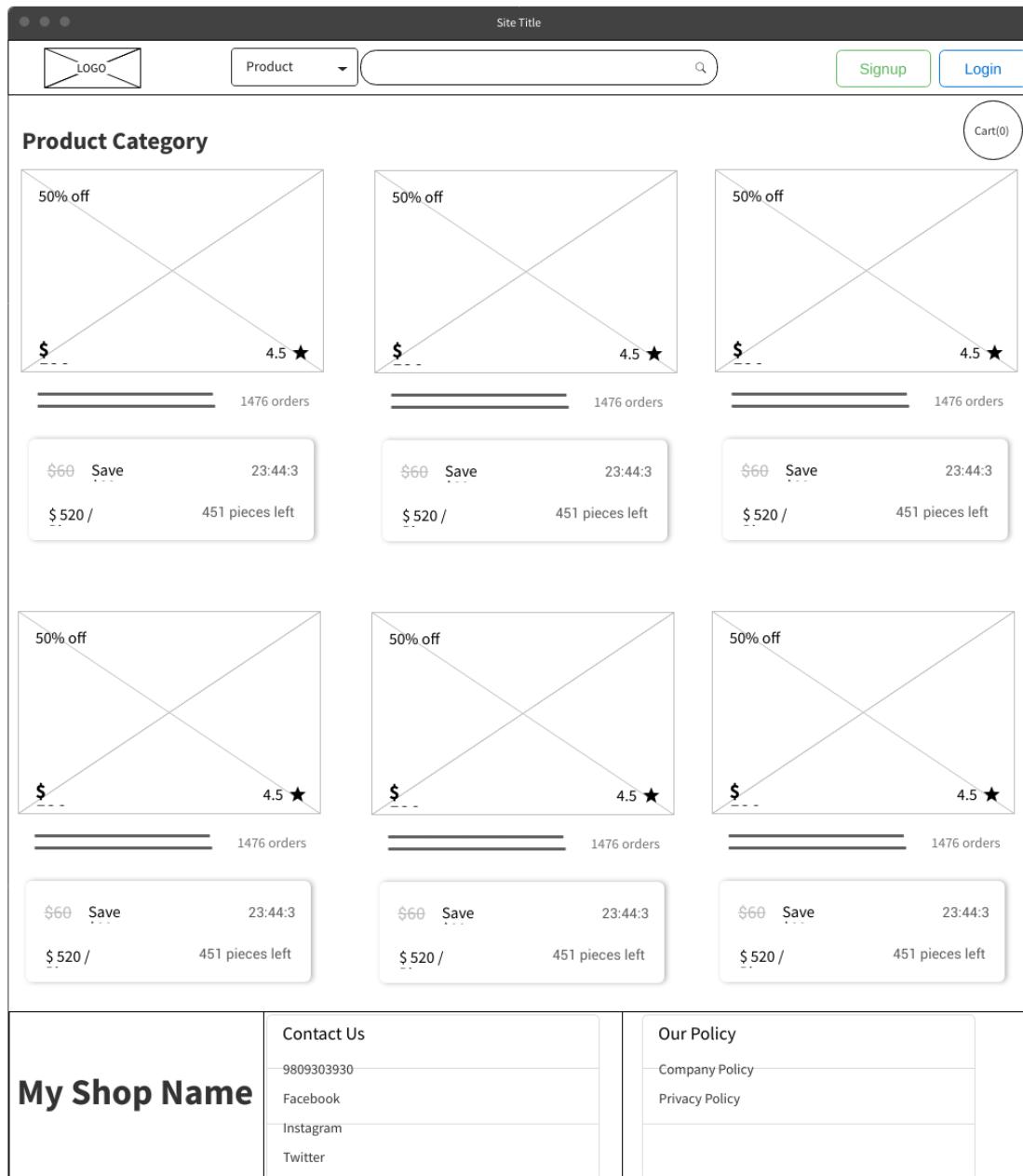


Figure 130 Wireframe i

Site Title

LOGO

Product

Category Name

Product Name

\$ 1245 \$12645 FLAT 24% off

BANK OFFERS EXTRA 5% OFF ON STATE BANK CREDIT AND DEBIT CARDS

Product Delivery

Enter Delivery Pincode

Color

Green Red Blue white

Size

XS SM M L XL XXL

Add to Cart

Warranty

Seller

Product Description

Similar Category Item

50% off

\$ ~~60~~ Save

1476 orders

23:44:3

\$ 520 / 451 pieces left

50% off

\$ ~~60~~ Save

1476 orders

23:44:3

\$ 520 / 451 pieces left

50% off

\$ ~~60~~ Save

1476 orders

23:44:3

\$ 520 / 451 pieces left

My Shop Name	Contact Us 9809303930 Facebook Instagram	Our Policy Company Policy Privacy Policy
---------------------	--	---

Figure

131 Wireframe

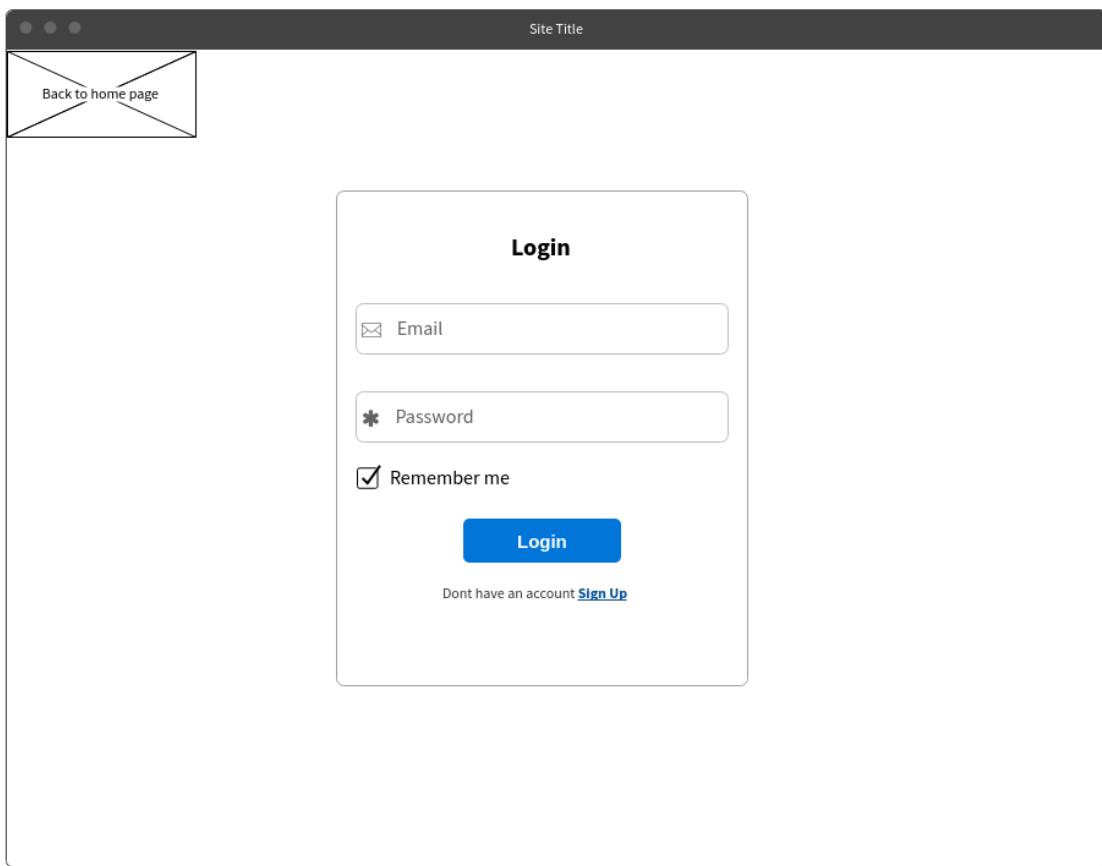


Figure 132 Wireframe login

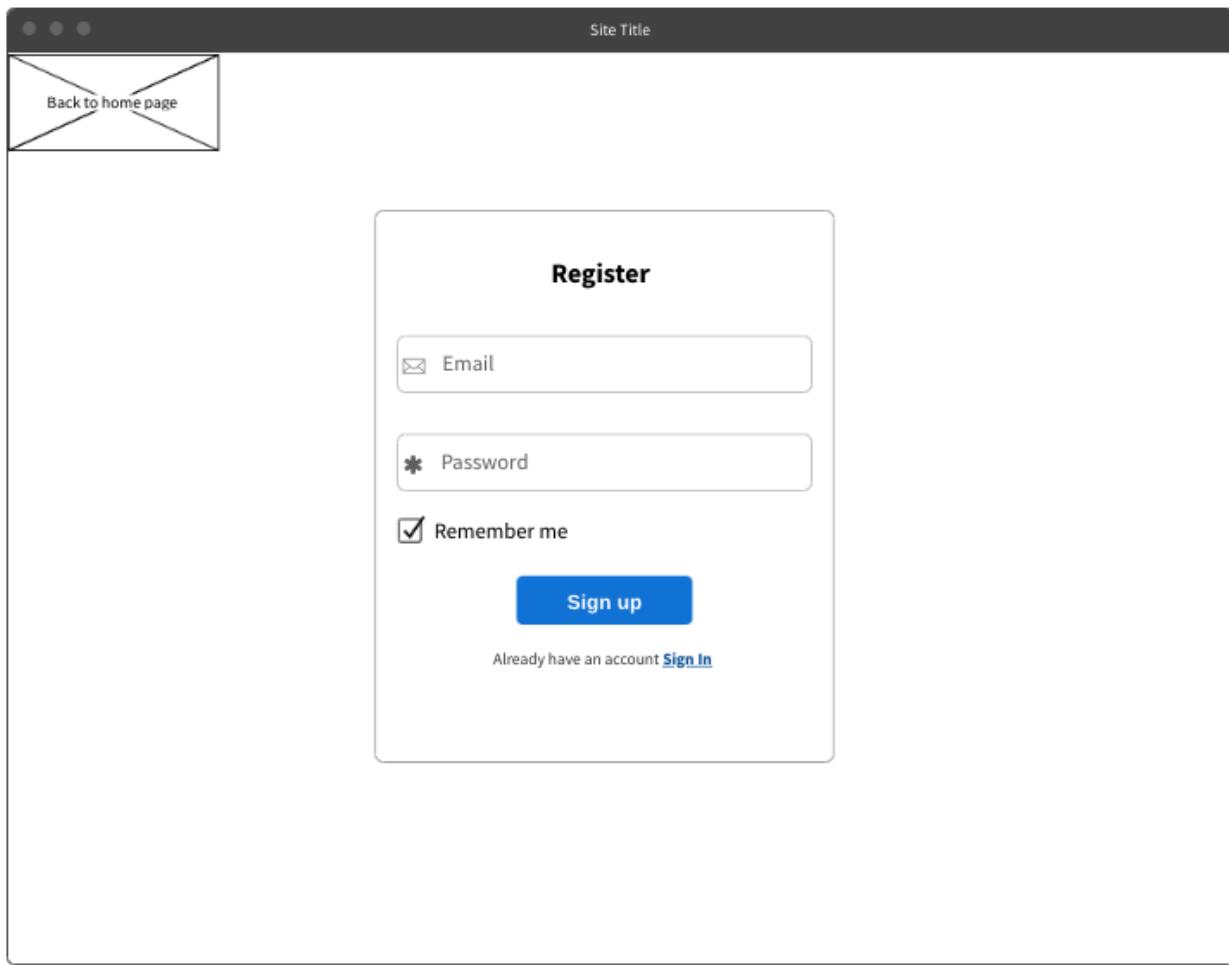


Figure 133 Wireframe register

A wireframe of a user details form. At the top left is a 'Back to home page' button. The top right corner has a placeholder 'Site Title'. The main content area is titled 'Your Details' and contains six input fields: 'Name', 'phone number', 'username', 'your address', 'gender', and 'Date of birth'. A blue 'Submit' button is located at the bottom of the form.

Figure 134 Wireframe user details

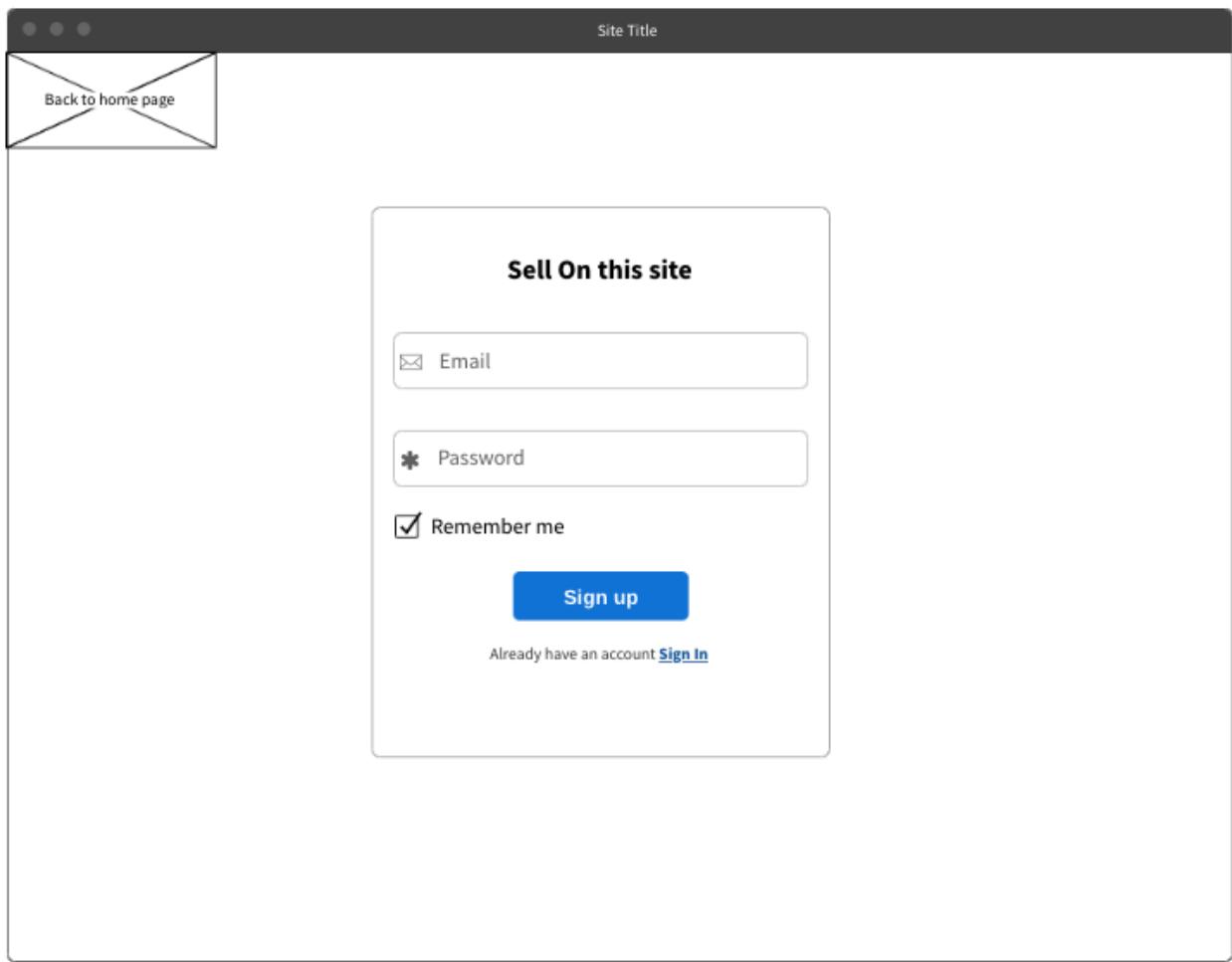
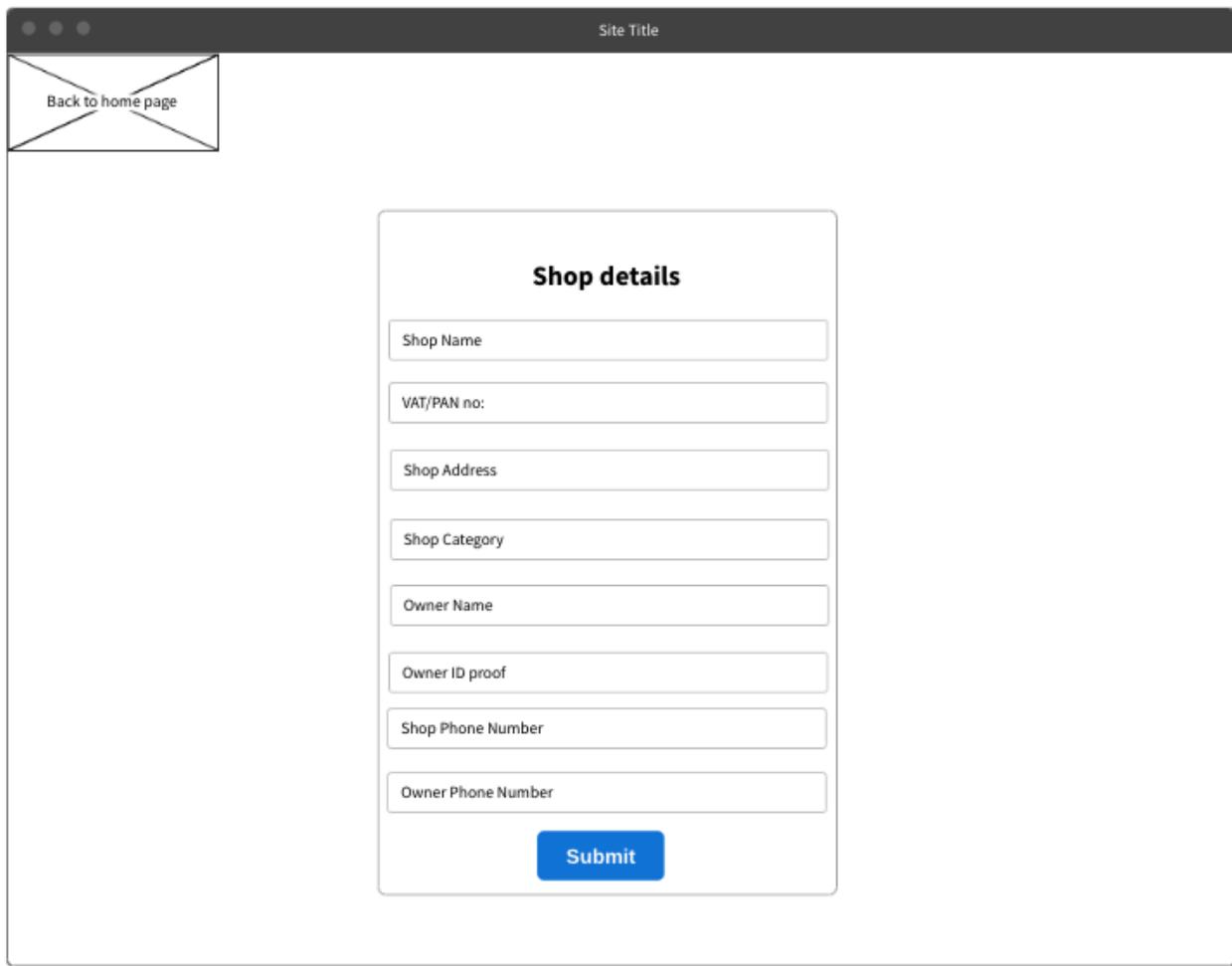


Figure 135 wireframe shop register



A wireframe of a web page titled "Site Title". In the top-left corner, there is a "Back to home page" button with a large "X" over it. The main content area is a form titled "Shop details" containing the following fields:

- Shop Name
- VAT/PAN no:
- Shop Address
- Shop Category
- Owner Name
- Owner ID proof
- Shop Phone Number
- Owner Phone Number

At the bottom of the form is a blue "Submit" button.

Figure 136 Wireframe shop details

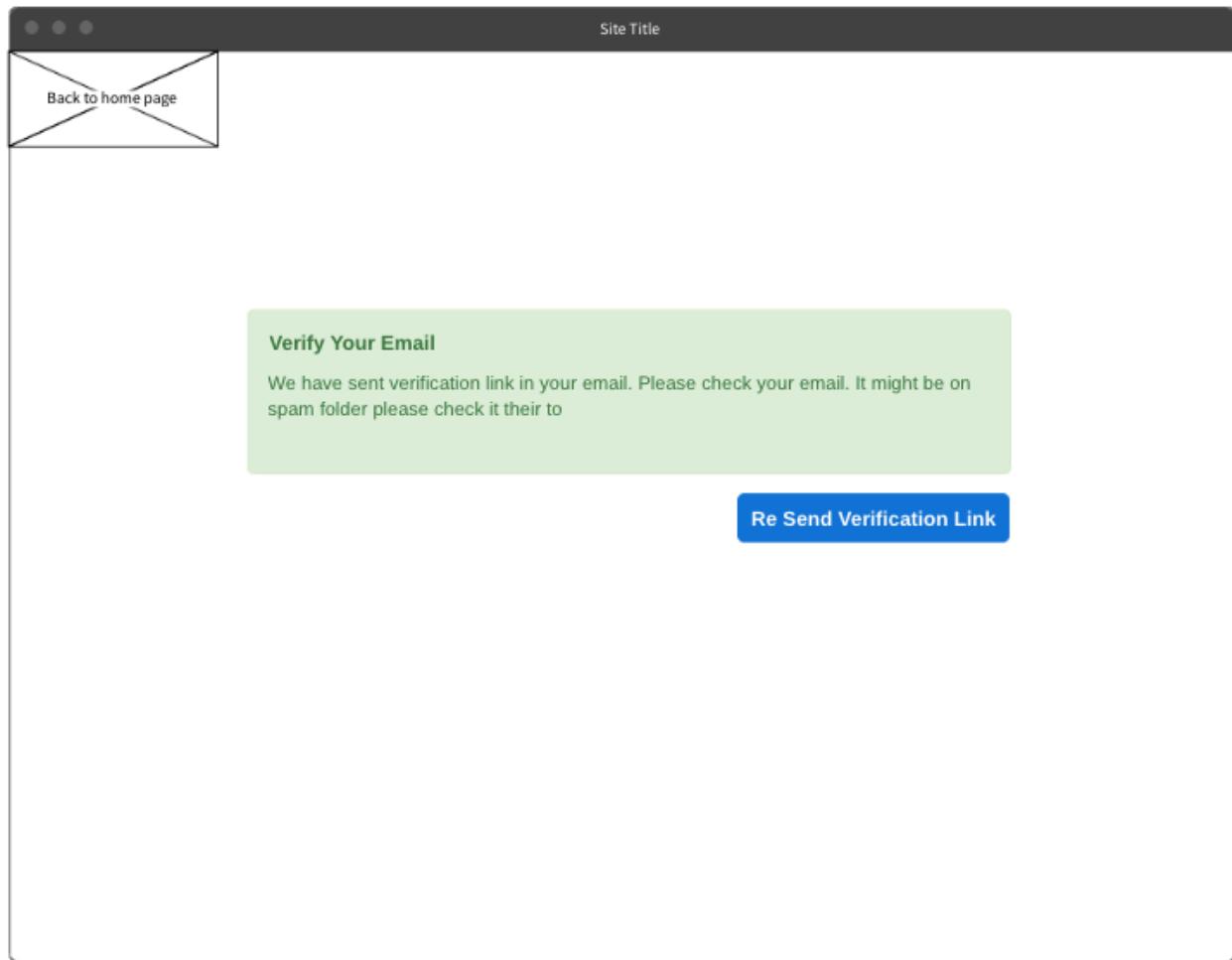


Figure 137 email verification

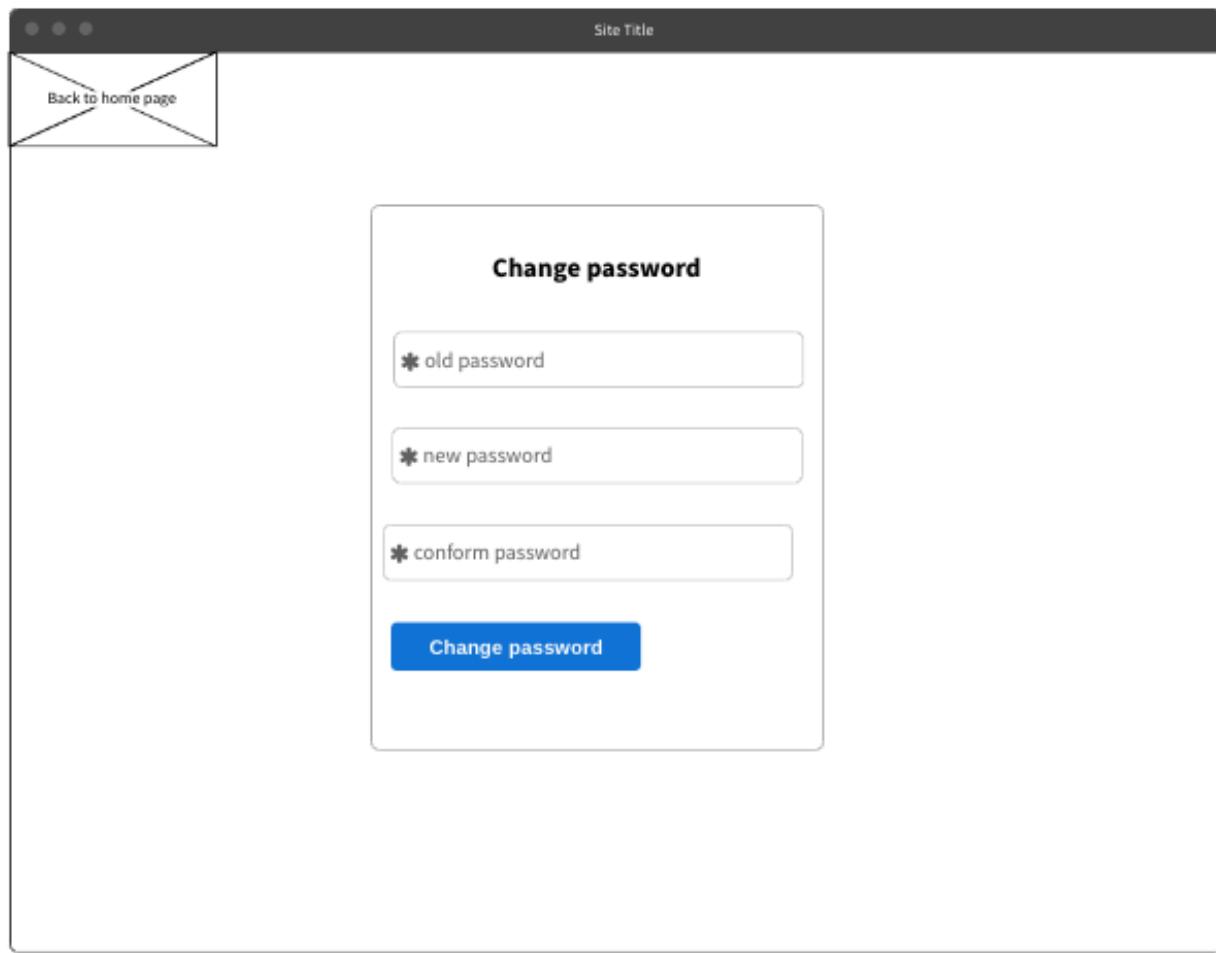


Figure 138 Wireframe change password



Figure 139 Wireframe password reset link

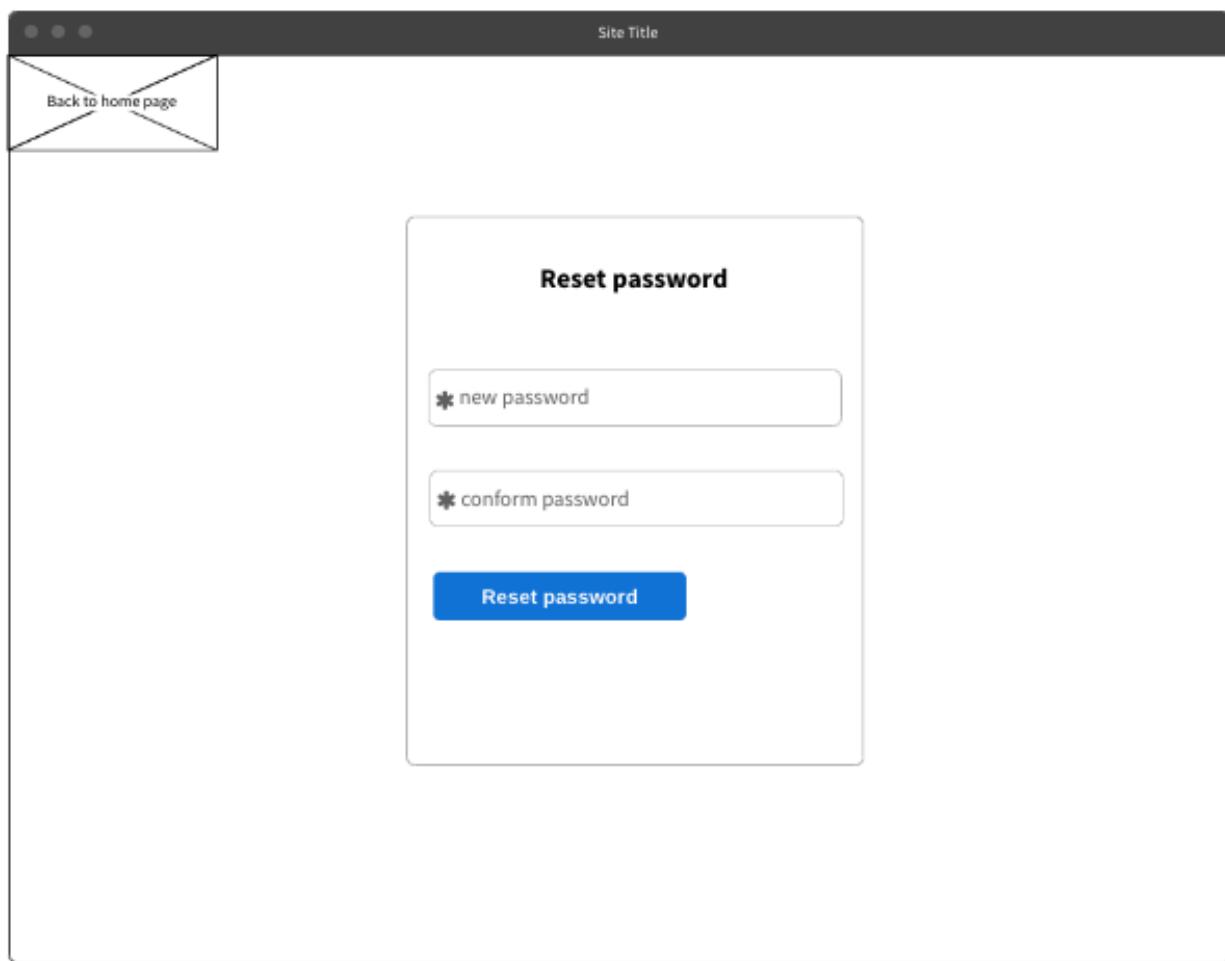


Figure 140 Wireframe reset

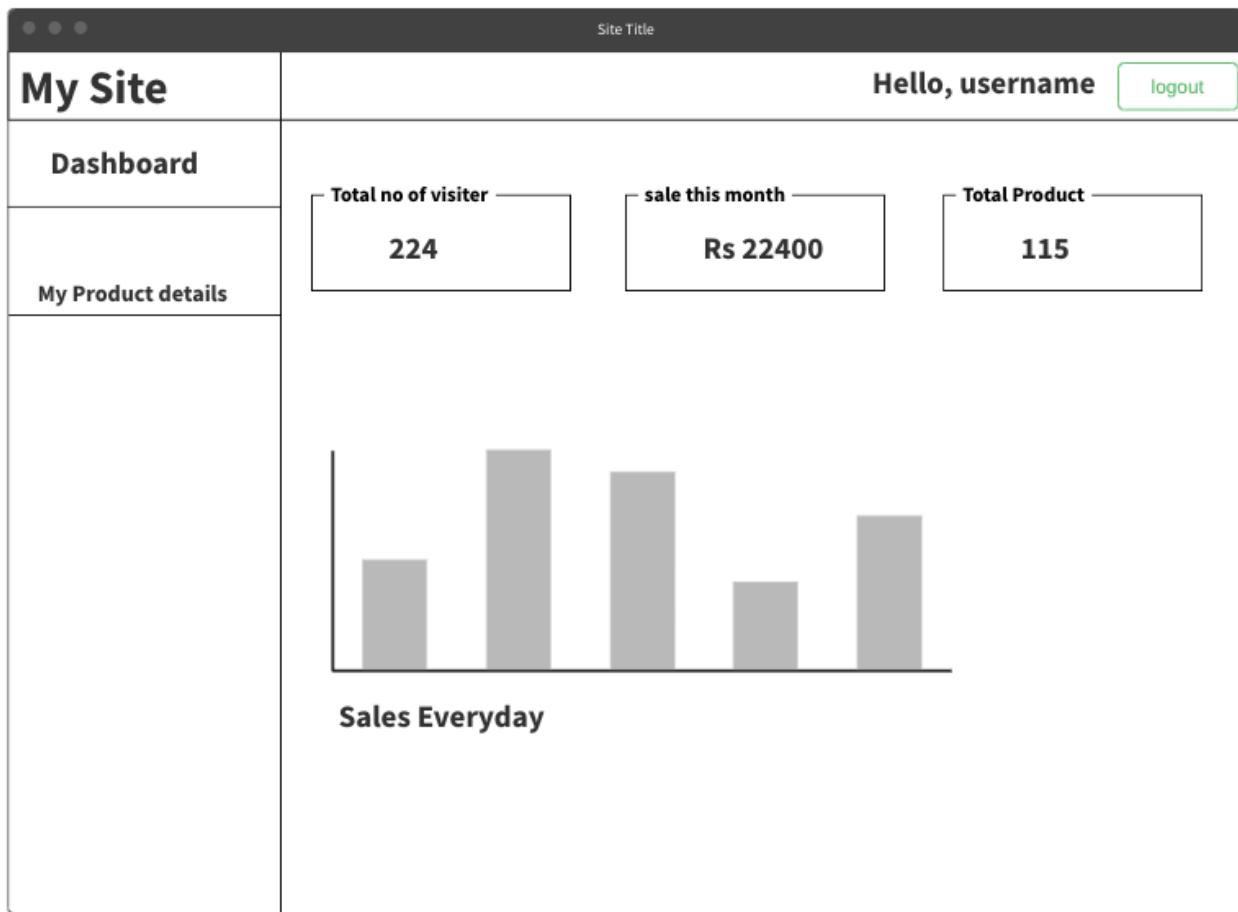


Figure 141 Wireframe dashboard 1

My Site		Hello, username logout																																																																							
Dashboard																																																																									
My Product details		<table border="1"> <thead> <tr> <th>Sn</th><th>Product Name</th><th>Price</th><th>Stock</th><th>views</th><th colspan="2">Action</th></tr> </thead> <tbody> <tr><td>1</td><td>Tshirt</td><td>RS 1000</td><td>2</td><td>3</td><td>Edit</td><td>Delete</td></tr> <tr><td>1</td><td>Tshirt</td><td>RS 1000</td><td>2</td><td>4</td><td>Edit</td><td>Delete</td></tr> <tr><td>1</td><td>Tshirt</td><td>RS 1000</td><td>2</td><td>5</td><td>Edit</td><td>Delete</td></tr> <tr><td>1</td><td>Tshirt</td><td>RS 1000</td><td>2</td><td>8</td><td>Edit</td><td>Delete</td></tr> <tr><td>1</td><td>Tshirt</td><td>RS 1000</td><td>2</td><td>7</td><td>Edit</td><td>Delete</td></tr> <tr><td>1</td><td>Tshirt</td><td>RS 1000</td><td>2</td><td>6</td><td>Edit</td><td>Delete</td></tr> <tr><td>1</td><td>Tshirt</td><td>RS 1000</td><td>2</td><td>6</td><td>Edit</td><td>Delete</td></tr> <tr><td>1</td><td>Tshirt</td><td>RS 1000</td><td>2</td><td>5</td><td>Edit</td><td>Delete</td></tr> <tr><td>1</td><td>Tshirt</td><td>RS 1000</td><td>2</td><td>5</td><td>Edit</td><td>Delete</td></tr> </tbody> </table>		Sn	Product Name	Price	Stock	views	Action		1	Tshirt	RS 1000	2	3	Edit	Delete	1	Tshirt	RS 1000	2	4	Edit	Delete	1	Tshirt	RS 1000	2	5	Edit	Delete	1	Tshirt	RS 1000	2	8	Edit	Delete	1	Tshirt	RS 1000	2	7	Edit	Delete	1	Tshirt	RS 1000	2	6	Edit	Delete	1	Tshirt	RS 1000	2	6	Edit	Delete	1	Tshirt	RS 1000	2	5	Edit	Delete	1	Tshirt	RS 1000	2	5	Edit	Delete
Sn	Product Name	Price	Stock	views	Action																																																																				
1	Tshirt	RS 1000	2	3	Edit	Delete																																																																			
1	Tshirt	RS 1000	2	4	Edit	Delete																																																																			
1	Tshirt	RS 1000	2	5	Edit	Delete																																																																			
1	Tshirt	RS 1000	2	8	Edit	Delete																																																																			
1	Tshirt	RS 1000	2	7	Edit	Delete																																																																			
1	Tshirt	RS 1000	2	6	Edit	Delete																																																																			
1	Tshirt	RS 1000	2	6	Edit	Delete																																																																			
1	Tshirt	RS 1000	2	5	Edit	Delete																																																																			
1	Tshirt	RS 1000	2	5	Edit	Delete																																																																			
						«	1	2	3																																																																

Figure 142 Wireframe dashboard2

The screenshot shows a web-based dashboard titled "My Site". The top navigation bar includes "Site Title", "Hello, username", and "logout". A sidebar on the left lists "Dashboard" and "My Product details". The main content area displays a table of product details. A modal dialog box is overlaid on the page, asking "Are You sure you want to delete" and providing a message: "once deleted item can be bought back so be wise while performing this action". The table data is as follows:

Sn	Product Name	Category	Price	Stock	Action
1	Tshirt		RS 1000	2	<button>Edit</button> <button>Delete</button>
1	Tshirt		RS 1000	2	<button>Edit</button> <button>Delete</button>
1	Tshirt		RS 1000	2	<button>Edit</button> <button>Delete</button>
1	Tshirt		RS 1000	2	<button>Edit</button> <button>Delete</button>
1	Tshirt		RS 1000	2	<button>Edit</button> <button>Delete</button>
1	Tshirt		RS 1000	2	<button>Edit</button> <button>Delete</button>
1	Tshirt		RS 1000	2	<button>Edit</button> <button>Delete</button>
1	Tshirt		RS 1000	2	<button>Edit</button> <button>Delete</button>

Pagination at the bottom right shows pages 1, 2, and 3.

Figure 143 dashboard 3

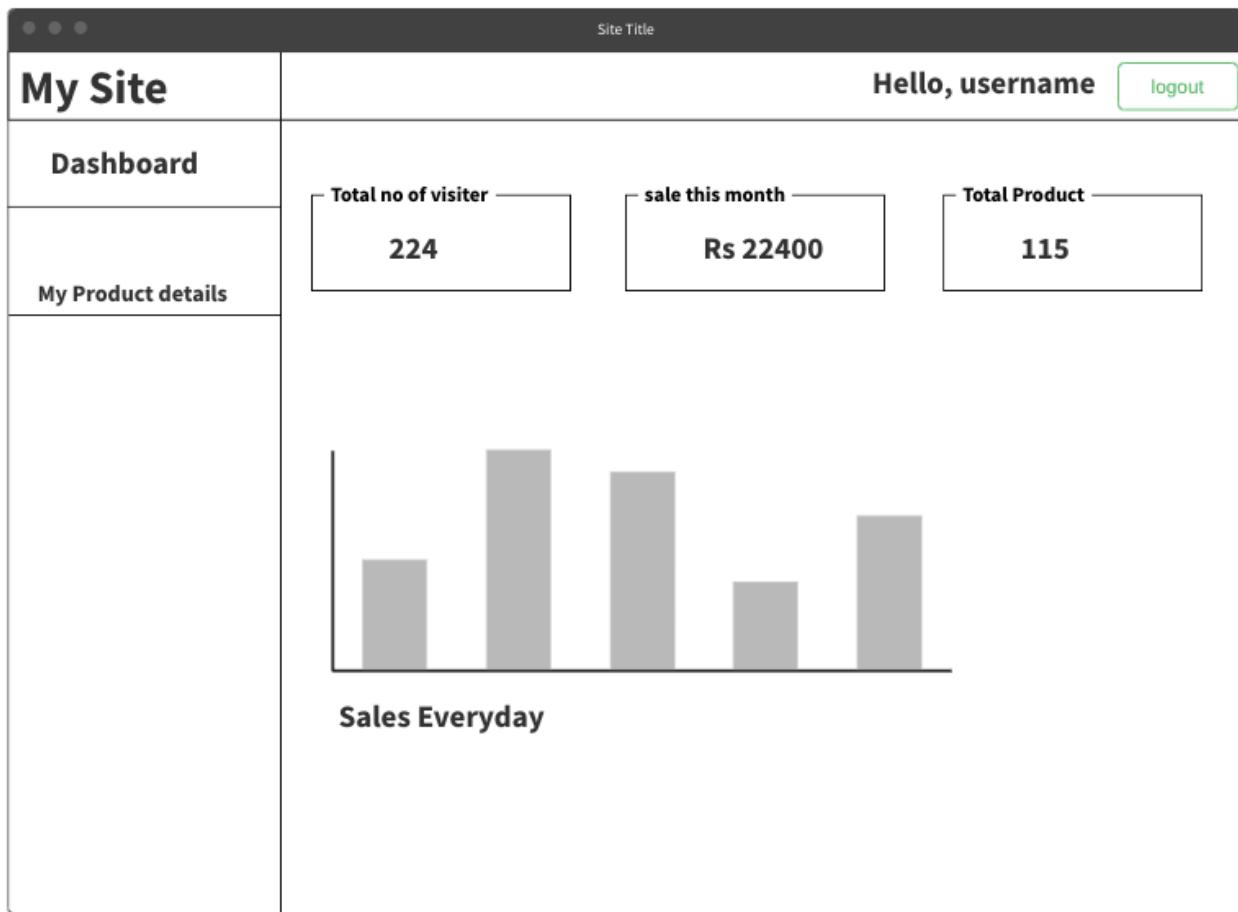


Figure 144 Wireframe dashboard 4

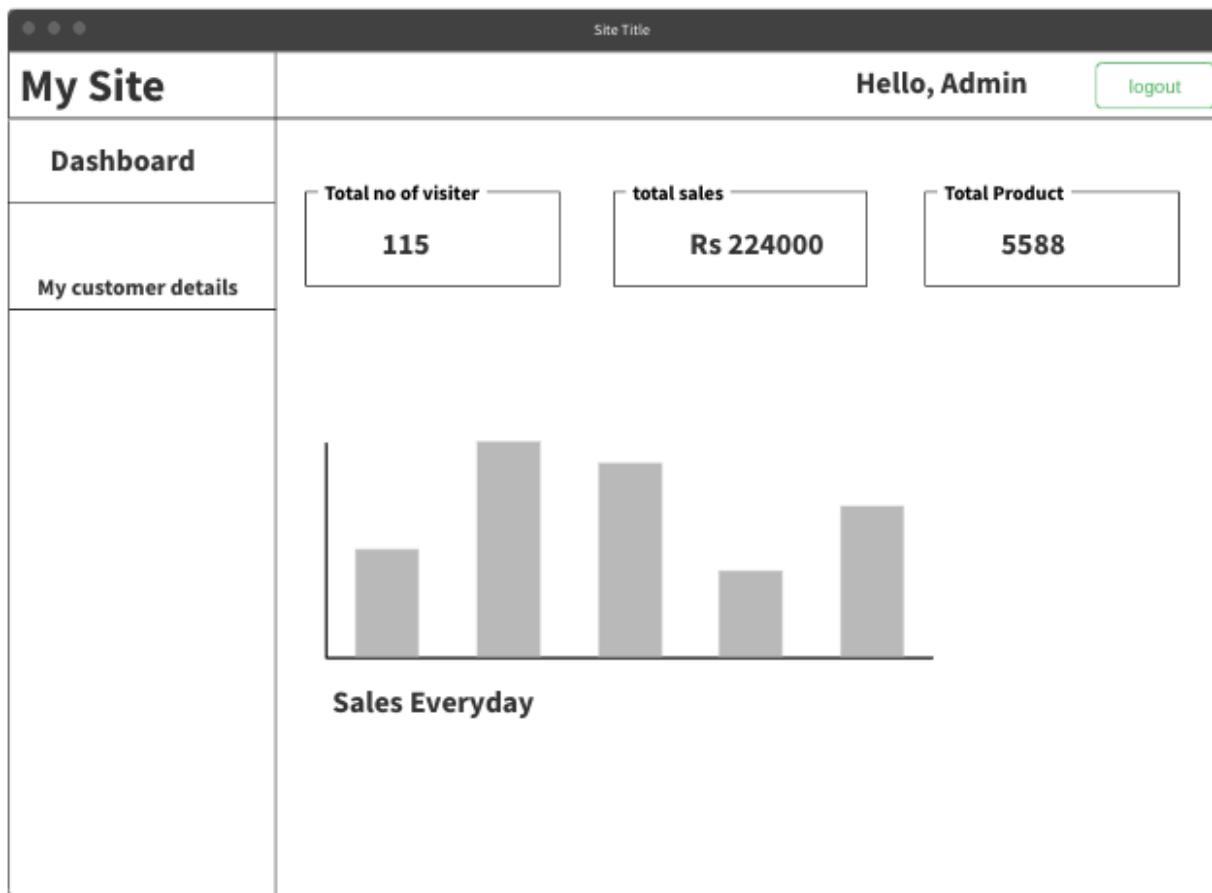


Figure 145 Wireframe dashboard 5

7.5 APPENDIX E: SCREENSHOTS OF THE SYSTEM

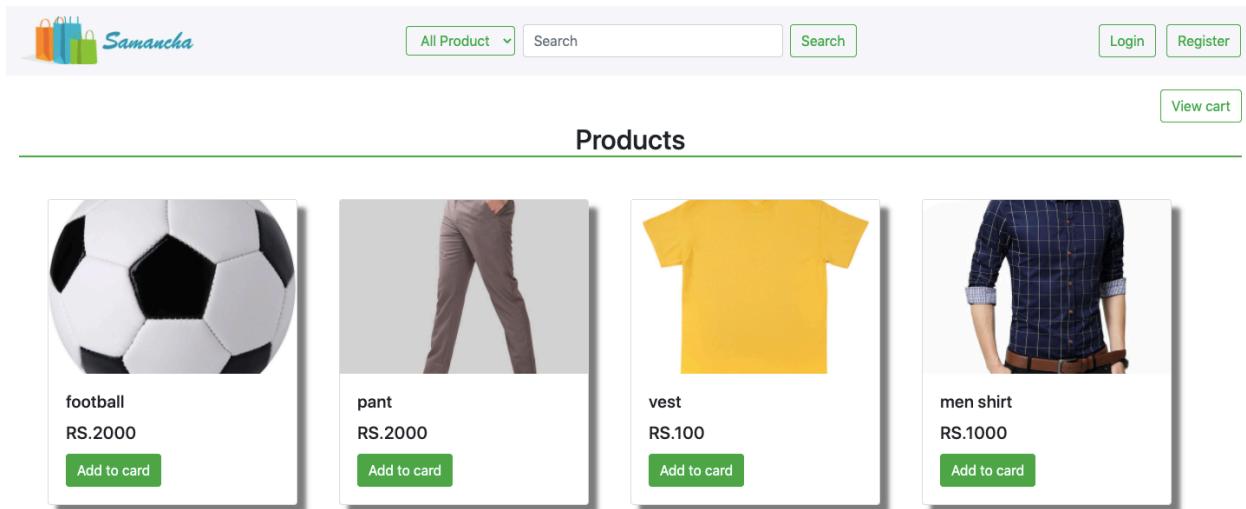


Figure 146 screenshot of the system 1

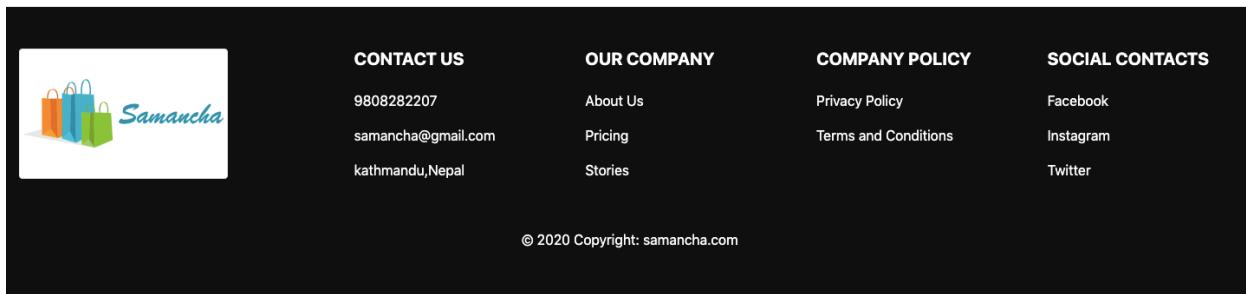


Figure 147 screenshot of the system 2

My Cart

Show 10 entries Search:

Product Name	Product Photo	Product Price	Action
football		2000	<button>Remove From Cart</button>
pant		2000	<button>Remove From Cart</button>

Showing 1 to 2 of 2 entries

Total:4000
Check Out

Figure 148 screenshot of the system 3

My Order Details

Show 10 entries Search:

Product Name	Product Photo	Product Price	Status
football		2000	On the Way
pant		2000	In review

Showing 1 to 2 of 2 entries

Figure 149 4

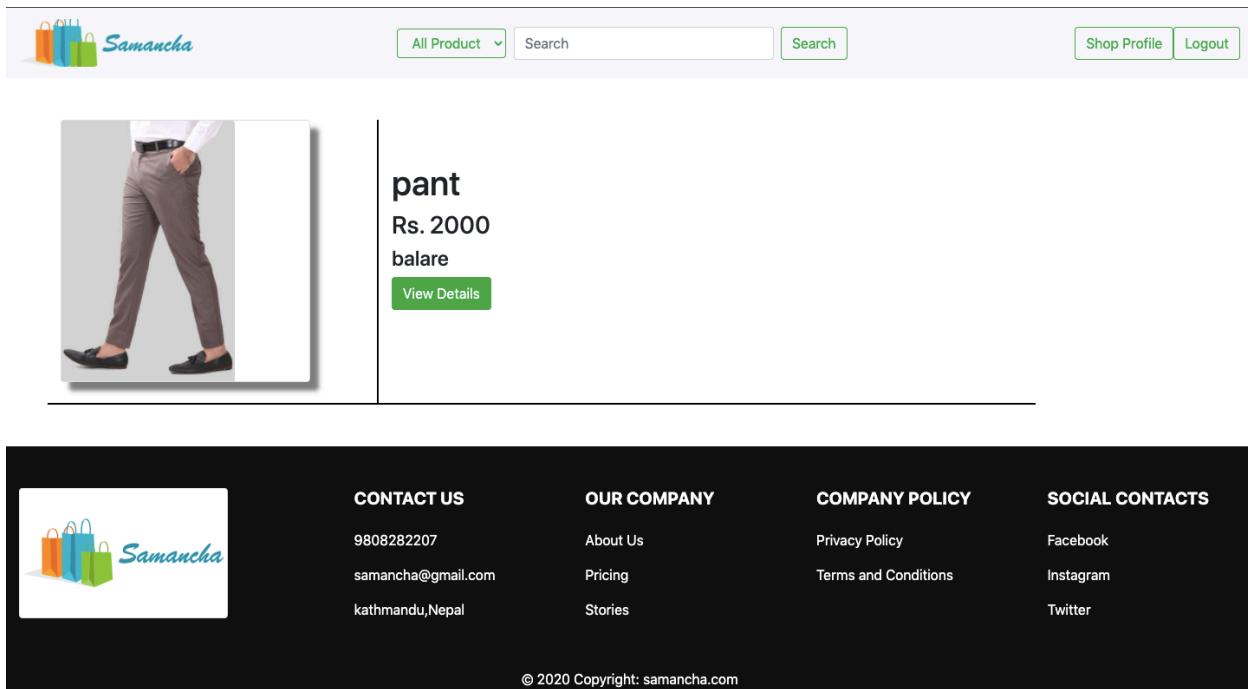


Figure 150 screenshot of the system 5

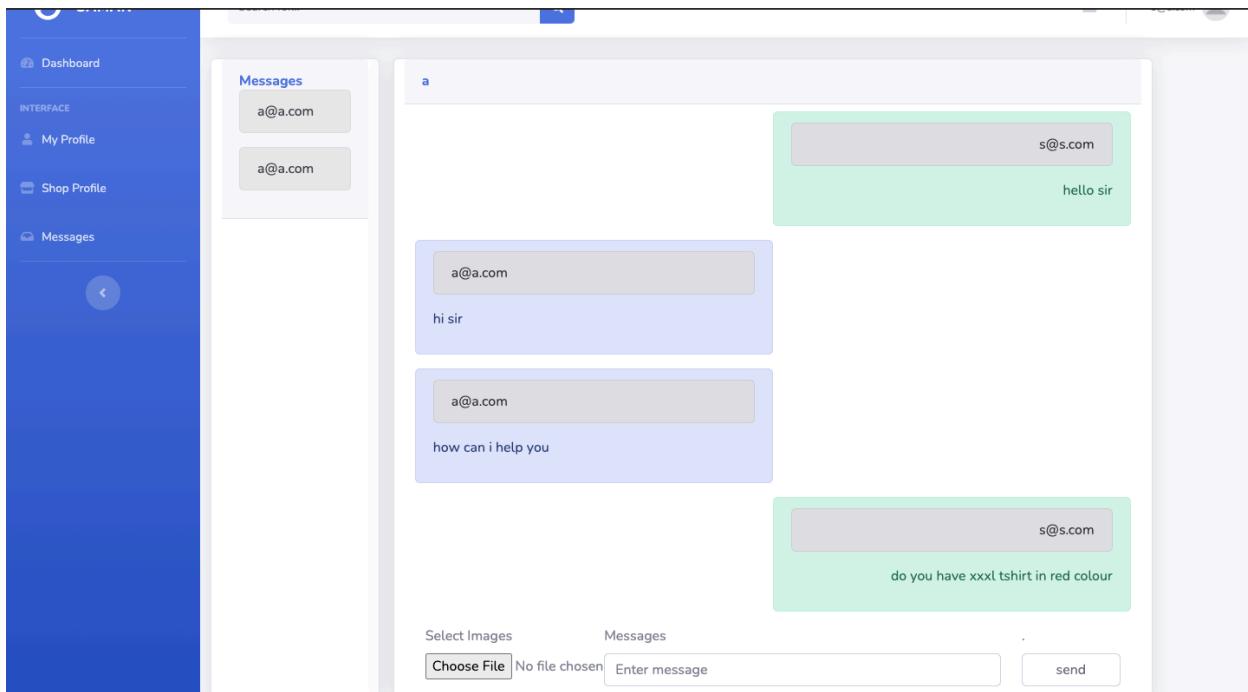


Figure 151 screenshot of the system 6

Shop Profile

Show 10 entries

Shop Name	Shop User Name	Shop Category	Shop Verification	Action
pujanrajrai	pujanraj	food	True	view Shop Details
sulav	sulav	sulav	False	view Shop Details
Shop Name	Shop User Name	Shop Category	Shop Verification	Action

Showing 1 to 2 of 2 entries

Copyright © Your Website 2020

Figure 153 screenshot of the system 7

Shop Profile

Show 10 entries

Product Name	Stock	Price	Image
men shirt	88	1000	
vest	77	100	
Product Name	Stock	Price	Image

Showing 1 to 2 of 2 entries

Copyright © Your Website 2020

Figure 152 screenshot of the system 8

The screenshot shows a web-based application interface for managing orders. On the left, a sidebar menu includes 'Dashboard', 'INTERFACE', 'My Profile', 'Shop Profile', and 'Messages'. The main content area is titled 'My Order Details' and displays a table of two entries:

Product Name	Product Photo	Product Price	Customer Name	Customer Address	Status
football		2000	a	a	<button>Deliver item</button>
pant		2000	a	a	<button>Item in review send item</button>

Below the table, there are buttons for 'Previous' and 'Next'.

Figure 154 screenshot of the system 9

The screenshot shows a messaging interface. The sidebar menu is identical to Figure 154. The main content area is titled 'Messages' and shows a conversation between 'a@a.com' and 's@s.com' with the following messages:

- a@a.com: hello sir
- a@a.com: hi sir
- a@a.com: how can i help you
- s@s.com: do you have xxl tshirt in red colour

At the bottom, there are buttons for 'Select Images' and 'Messages'.

Figure 155 screenshot of the system 10

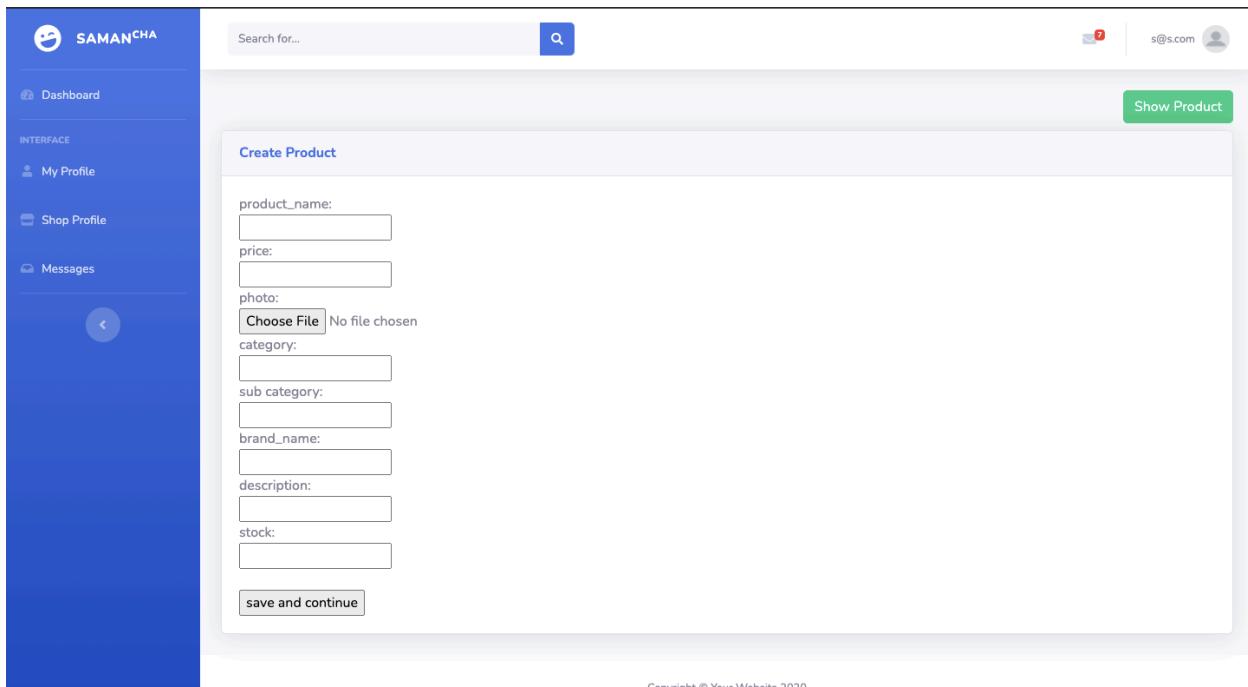


Figure 156 screenshot of the system 11

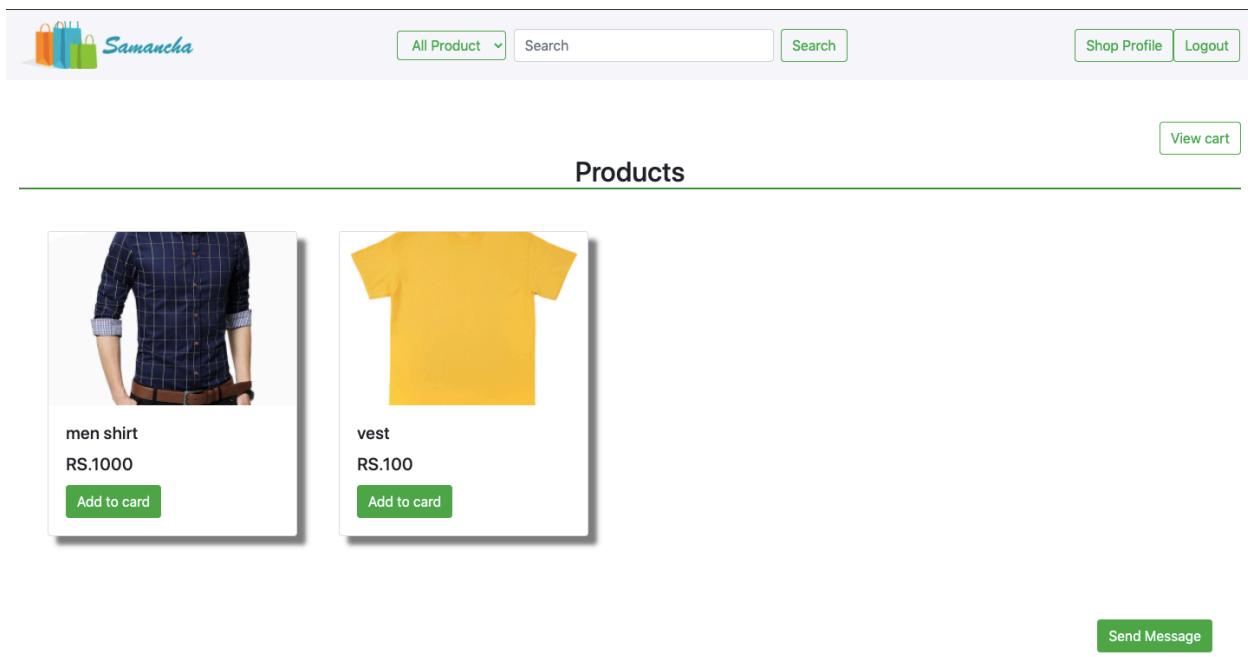


Figure 157 screenshot of the system 12

7.6: APPENDIX F: USER FEEDBACK

7.6.1 USER FEEDBACK FORM

User Feedback form

Form description

name *

Short answer text

email *

Short answer text

Figure 158 USER FEEDBACK FORM 1

e-commerce Feedback form(1:unsatisfactory, 5:Perfect)

	1	2	3	4	5
Design	<input type="radio"/>				
Simplicity	<input type="radio"/>				
User Friendliness	<input type="radio"/>				
Security	<input type="radio"/>				
privacy	<input type="radio"/>				
speed	<input type="radio"/>				
future of my we...	<input type="radio"/>				

Figure 159 USER FEEDBACK FORM 2

7.6.2 SAMPLE OF FILLED USER FEEDBACK FORMS

User Feedback form

* Required

name *
pujan raj rai

email *
pujanr0809@gmail.com

Figure 160 SAMPLE OF FILLED USER FEEDBACK FORMS 1

e-commerce Feedback form(1:unsatisfactory, 5:Perfect)

	1	2	3	4	5
Design	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Simplicity	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
User Friendliness	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Security	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
privacy	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
speed	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
future of my website	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>

[Clear selection](#)

[Submit](#)

Figure 161 SAMPLE OF FILLED USER FEEDBACK FORMS 2

7.7 APPENDIX G: FUTURE WORK

7.7.1 READINGS FOR FUTURE WORK

The first phase of the development is completed but there are still room for improvement in a lot of place. To integrate as many payment systems as possible to the system I will read the documentation provided by the payment gateway and purchase their merchant account. To add notification, I will be using Django message system which is used to send short message. To implement it I will be reading Django official documentation. I will be learning different algorithm of machine learning from different source available in internet. Then I will implement recommendation system with the help of that. In future I will be developing mobile apps for this project so I have to create APIs. So, I will be learning Django rest framework to make apis.