

**Module Code & Module Title****CS6PO5NT Final Year Project****Assessment Weightage & Type****40% Individual Coursework****Year and Semester****2022 Autumn, Year long****Student Name: Rishi Raj Shrestha****College ID: 19031820****Assignment Submission Date: 27<sup>th</sup> April 2022****Academic Supervisors****Internal Supervisor: Prateek Kokh Shrestha****External Supervisor: Prabodh Tuladhar**

*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**

Firstly, I would like to express my gratitude to London Metropolitan University and Itahari International College for arranging the Final Year Project Module (CS6P05NTI). I am very thankful for being eligible to complete this project.

While working in this project work, I have been helped by my respected supervisor, who heartily deserves my greatest gratitude. I would like to show my gratitude to my supervisor Mr. Prateek Kokh Shrestha for taking out his precious time and helping me to complete this complete this project. I am very thankful to him for creating the best learning environment, guiding me in each step and helping me by providing the information in different sectors. I am very fortunate to work under his guideline.

I would like to thank all the individual who have directly or indirectly contributed in completing the project within the limited time frame.

## **Abstract**

The project's main goal of providing an online store where customers can place orders and book counsellors has been achieved, as well as objectives such as testing and client review using various APIs. A complete web app named "Guardian" has been developed successfully, incorporating all the features and functionality outlined previously. Understanding the risk of physical contact during covid time and the demanding need of people to have online presence of doctors, pharmacists, a project named 'Guardian' is introduced.

For pharmacists, counsellors, and their parents, the projects serve as an internet hub. The customer can order medicine by uploading their prescription sheet. It also makes it easier for them to choose, order, and pay online or offline. It also provides patients (clients) with information about the medications they are taking. For the development of the system database like PostgreSQL and SQLite has been used with VS code as a code editor. The backend is done using python programming language using Django framework. The projects are well compared with other similar projects and the features has been critically evaluated. Similarly different methodology was implemented like waterfall, extreme programming but Scrum has been selected. Hence, overall development of the system is based on the scrum methodology as it satisfies the need and requirements of the projects to be developed. The project's testing included white-box testing, black-box testing, and system testing, all of which were done properly with proper test cases. In the same way, a survey of users has been conducted, which includes a pre-survey and post-survey form. The project has been updated because of the user feedback and analysis.

Overall, the system development and learning experience have been fantastic. It helped me a lot to increase and boost my research and critical thinking by doing all the components of the system on my own research and plan.

---

## Table of Contents

1.	Introduction .....	1
1.1.	Project Description .....	1
1.2.	Current Scenario .....	2
1.3.	Problem Domain .....	3
1.4.	Project as a Solution .....	4
1.5.	Aims and Objectives .....	4
1.6.	Structure of the Report.....	5
1.6.1.	Background .....	5
1.6.2.	Development .....	5
1.6.3.	Testing and Analysis .....	5
1.6.4.	Conclusion.....	5
2.	Background.....	6
2.1.	End Users .....	6
2.2.	Understanding the Solution .....	6
2.2.1.	Overview of the system .....	6
2.2.2.	Technical terms and definition .....	6
2.2.3.	Database .....	7
2.3.	Similar Projects .....	8
2.3.1.	Dhani Pharmacy .....	8
2.3.2.	Netmeds.com .....	9
2.3.3.	Nepmeds .....	11
2.4.	Comparisons .....	13
3.	Development.....	14
3.1.	Considered Methodologies.....	14

3.1.1. Waterfall Methodology .....	14
3.1.2. Extreme Programming.....	16
3.2. Selected Methodology.....	17
3.2.1. Scrum .....	17
3.3. Phases of Methodology.....	19
3.3.1. Initiate.....	19
3.3.2. Planning.....	19
3.3.3. Design .....	19
3.3.4. Implementation .....	19
3.3.5. Testing.....	20
3.3.6. Review and Retrospect.....	20
3.3.7. Deployment and Documentation .....	20
3.4. Development - Scrum Cycle.....	20
3.4.1. Planning.....	20
3.4.2. Sprint 1 .....	21
3.4.3. Sprint 2 .....	21
3.4.4. Sprint 3 .....	22
3.4.5. Sprint 4 .....	22
3.4.6. Sprint 5 .....	23
3.4.7. Sprint 6 .....	23
3.4.8. Sprint 7 .....	24
3.4.9. Sprint 8 .....	24
3.4.10. Deployment .....	25
3.5. Survey Result.....	25
3.5.1. Pre-Survey Results.....	25

---

3.5.2. Post-Survey Results .....	29
3.6. Requirement Analysis .....	31
3.6.1. Features Requirement.....	31
3.6.2. Hardware Requirement.....	32
3.6.3. Software Requirement.....	32
3.6.4. Programming Language Requirement.....	33
3.6.5. Web Application Framework Requirement .....	33
3.6.6. Libraries Requirement .....	34
3.7. Design.....	35
3.7.1. Planning Phase .....	35
3.7.2. Sprint 1 .....	36
3.7.3. Sprint 2 .....	45
3.7.4. Sprint 3 .....	71
3.8. Implementation.....	75
3.8.1. Database design.....	75
3.8.2. System Architecture .....	78
3.8.3. Important Screenshots.....	80
4. Testing and Analysis .....	84
4.1. Test Plan .....	86
4.1.1. Testing Plan.....	86
4.2. Black Box Testing .....	88
4.2.1. Testing Login button .....	88
4.2.2. Testing logout button .....	91
4.2.3. Testing Cart .....	93
4.2.4. Testing Message in Cart page .....	97

---

4.2.5. Testing increment and decrement of Cart item.....	99
4.2.6. Testing Booking button .....	101
4.2.7. Testing Pharmacy detail .....	103
4.2.8. Testing Counsellor detail .....	105
4.2.9. Testing Medicine detail .....	106
4.2.10. Testing Blog Page .....	108
4.2.11. Testing Payment Button .....	109
4.3. White Box testing .....	111
4.3.1. Testing User registration.....	111
4.3.2. Testing Pharmacy Registration.....	115
4.3.3. Testing Counsellor Registration.....	120
4.3.4. Testing User Login .....	123
4.3.5. Testing Forget Password.....	124
4.3.6. Testing Reset Password.....	126
4.3.7. Testing Admin Login .....	128
4.3.8. Testing City Page .....	130
4.3.9. Testing Pharmacy Page .....	131
4.3.10. Testing View user details .....	132
4.3.11. Testing view user document.....	134
4.3.12. Testing approve user.....	136
4.3.13. Testing add Medicine .....	139
4.3.14. Testing add Blog .....	141
4.3.15. Testing Payment .....	144
4.3.16. Validation Testing.....	147
4.4. Integration Testing .....	151

---

4.4.1.	Testing Create Operation of the medicine.....	151
4.4.2.	Testing Create Operation of the Blog .....	154
4.4.3.	Testing Read Operation of the Medicine .....	157
4.4.4.	Testing Read Operation of the Blog .....	158
4.4.1.	Testing Delete Operation of the Medicine.....	160
4.4.2.	Testing Delete Operation of the Blog.....	162
4.5.	System Testing .....	164
4.5.1.	Run the application in different browsers.....	164
4.5.2.	Run the application in online terminal for deployment.....	166
4.6.	Critical Analysis.....	167
5.	Conclusion .....	168
5.1.	Issues.....	169
5.1.1.	Legal Issues .....	169
5.1.2.	Social Issues .....	169
5.1.3.	Ethical Issues .....	170
5.2.	Advantages .....	170
5.3.	Limitations.....	171
5.4.	Future Work .....	171
6.	References.....	172
7.	Appendix .....	174
7.1.	Pre-Survey .....	174
7.1.1.	Pre-Survey Form .....	174
7.1.2.	Sample of Filled Pre-Survey Forms.....	180
7.1.3.	Pre-Survey Result .....	186
7.2.	Post-survey .....	191

---

7.2.1.	Post Survey Form .....	191
7.2.2.	Sample of filed Post Survey Form .....	193
7.2.3.	Post Survey Result .....	195
7.3.	Sample Codes.....	198
7.3.1.	Code of User interface.....	198
7.3.2.	Code for the Automation Script.....	232
7.4.	Design.....	234
7.4.1.	Gantt Chart .....	234
7.4.2.	Work Breakdown Structure.....	235
7.4.3.	Use Case.....	236
7.4.4.	Wireframe .....	237
7.5.	Screenshots of the System .....	244
7.6.	Future Work .....	255

## Table of Figures

Figure 1: Notice posted by Department of Drug Administration.	2
Figure 2: Health care product page of Dhani Pharmacy.....	8
Figure 3: Health Article page of Dhani Pharmacy. ....	9
Figure 4: Home page of Netmeds.com.....	10
Figure 5: Blog page of Netmeds.com .....	10
Figure 6: Home page of NepMeds .....	11
Figure 7: Uploading prescription feature on NetMeds .....	12
Figure 8: Health related articles. ....	12
Figure 9: Page to contact specialist Doctors. ....	13
Figure 10: Waterfall Methodology Model.....	15
Figure 11: Extreme Programming workflow. ....	16
Figure 12: Scrum Cycle.....	18
Figure 13: Pie Chart showing the need of online pharmacy.....	25
Figure 14:Pie Chart showing the need of online counsellor/peoples. ....	26
Figure 15: Pie Chart showing the need of counsellor during lockdown. ....	26
Figure 16: Bar graph showing the response thought on aged people. ....	27
Figure 17: Bar graph showing the response opinion during uncertain medical emergency. .....	27
Figure 18: Pie Chart showing the percentage of difficulties faced by response on understanding the prescription.....	28
Figure 19: Pie Chart showing the interest of response on online platform for pharmacy. .....	28
Figure 20: Pie Chart showing the interest of response on online platform for counsellor. .....	29
Figure 21: Post Survey result 1.....	29
Figure 22: Post Survey result 2.....	30
Figure 23: Post survey result 3.....	30
Figure 24: Gantt Chat for the project.....	35
Figure 25: Project Work break down .....	35
Figure 26: Mind Map of the project.....	36

Figure 27: Screenshot of home page wireframe.	37
Figure 28: Screenshot of login page wireframe.	38
Figure 29: Screenshot of pharmacy page wireframe.	38
Figure 30: Screenshot of counsellor page wireframe	39
Figure 31: Screenshot of counsellor detail page wireframe	40
Figure 32: Screenshot of search page wireframe	41
Figure 33: Screenshot of medicine detail page wireframe	42
Figure 34: Screenshot of blog page wireframe	43
Figure 35: Screenshot of read blog page wireframe	44
Figure 36: Official Logo of Application	45
Figure 37: Use case diagram for user registration.	46
Figure 38: Collaboration diagram for user registration.	47
Figure 39: Sequence diagram for register user.	47
Figure 40: Use case diagram for Pharmacy registration	48
Figure 41: Collaboration diagram for pharmacy registration	48
Figure 42: Sequence diagram for register pharmacy.	49
Figure 43: Use case diagram for counsellor registration	50
Figure 44: Collaboration diagram for counsellor registration	50
Figure 45: Sequence diagram for register counsellor.	51
Figure 46: Use case diagram for user login	52
Figure 47: Collaboration diagram for user login.	52
Figure 48: Sequence diagram for user login.	53
Figure 49: Use case diagram for admin login	54
Figure 50: Collaboration diagram for admin login	54
Figure 51: Sequence diagram for admin login	55
Figure 52: Use case diagram for add medicine	56
Figure 53: Collaboration diagram for add medicine	56
Figure 54: Sequence diagram for add medicine	57
Figure 55: Use case diagram for add blog	58
Figure 56: Collaboration diagram for add blog	58
Figure 57: Sequence diagram for add blog	59

---

Figure 58: Use case diagram for add to cart .....	60
Figure 59: Collaboration diagram for add to cart .....	60
Figure 60: Sequence diagram for add to cart .....	61
Figure 61: Use case diagram for make payment.....	62
Figure 62: Collaboration diagram for making payment.....	62
Figure 63: Sequence diagram for make payment. ....	63
Figure 64: Use case diagram for add review.....	64
Figure 65: Collaboration diagram for add review.....	64
Figure 66: Sequence diagram for add review.....	65
Figure 67: Use case diagram for book session/meeting.....	66
Figure 68: Collaboration diagram for booking meeting/session.....	66
Figure 69: Sequence diagram for booking meeting/session.....	66
Figure 70: Use case diagram for upload prescription.....	67
Figure 71: Collaboration diagram for upload prescription.....	68
Figure 72: Sequence diagram for upload prescription.....	68
Figure 73: Use case diagram for add cities. ....	69
Figure 74: Collaboration diagram for add cities.....	70
Figure 75: Sequence diagram for add cities.....	70
Figure 76: Screenshot of homepage. ....	71
Figure 77: Screenshot of login page. ....	71
Figure 78: Screenshot of user registration page. ....	72
Figure 79: Screenshot of pharmacy registration page.....	72
Figure 80: Screenshot of counsellor registration page. ....	73
Figure 81: Screenshot of city page.....	73
Figure 82: Screenshot of shop page. ....	74
Figure 83: Screenshot of booking page.....	74
Figure 84: Entity Relationship Diagram (Drawn with help of draw.io).....	75
Figure 85: Entity Relationship Diagram (generated by Db visualizer) .....	76
Figure 86: System Architecture of the project. ....	79
Figure 87: Custom user Model .....	80
Figure 88: Testing account object. ....	81

---

Figure 89: Context Processor for Cart item .....	81
Figure 90: Cart Views.....	82
Figure 91: Booking View with Calendar API being Implemented. ....	83
Figure 92: Types of Software testing.....	84
Figure 93: Flow of testing strategies.....	86
Figure 94: Screenshot of Homepage before user login. ....	88
Figure 95: Screenshot of homepage after user login. ....	89
Figure 96: Screenshot of initial code. ....	89
Figure 97: Screenshot of after editing the code.....	90
Figure 98: Screenshot of homepage after updating the code.....	91
Figure 99: Screenshot of home page. ....	92
Figure 100: Screenshot of login page.....	92
Figure 101: Screenshot of product detail page.....	93
Figure 102: Screenshot of cart page. ....	94
Figure 103: Screenshot of store page. ....	94
Figure 104: Screenshot of code for cart session. ....	95
Figure 105: Screenshot of cart page after updating the code.....	96
Figure 106: Screenshot of store page after updating the code.....	97
Figure 107: Screenshot of empty cart page. ....	98
Figure 108: Screenshot of code added on cart page. ....	98
Figure 109: Screenshot of cart page with zero item on cart. ....	99
Figure 110: Screenshot of cart page. (1).....	100
Figure 111: Screenshot of cart page. (2).....	100
Figure 112: Screenshot of cart page. (3).....	100
Figure 113: Screenshot of counsellor detail page. ....	101
Figure 114: Screenshot of booking form page. ....	102
Figure 115: Screenshot of city page.....	103
Figure 116: Screenshot of pharmacy page. ....	104
Figure 117: Screenshot of pharmacy detail page.....	104
Figure 118: Screenshot of counsellor page.....	105
Figure 119: Screenshot of counsellor detail page. ....	106

---

Figure 120: Screenshot of store page.....	107
Figure 121: Screenshot of medicine detail page.....	107
Figure 122: Screenshot of blog page.....	108
Figure 123: Screenshot of read blog page.....	109
Figure 124: Screenshot of cart page.....	110
Figure 125: Screenshot of PayPal login.....	110
Figure 126: Screenshot of user registration page.....	111
Figure 127: Screenshot of user registration page with filled form.....	112
Figure 128: Screenshot of alert message.....	113
Figure 129: Screenshot of all the user's account.....	113
Figure 130: Screenshot of email verification send to user.....	114
Figure 131: Screenshot of user activation.....	114
Figure 132: Screenshot of login page.....	115
Figure 133: Screenshot of user registration page.....	116
Figure 134: Screenshot of admin page viewing users' details.....	117
Figure 135: Screenshot of pharmacy registration page.....	117
Figure 136: Screenshot of pharmacy registration page with filled data.....	118
Figure 137: Screenshot of admin page viewing pharmacy registered.....	118
Figure 138: Screenshot of admin viewing pharmacy document.....	119
Figure 139: Screenshot of pharmacy page.....	119
Figure 140: Screenshot of user registration page.....	120
Figure 141: Screenshot of admin page viewing users' details.....	121
Figure 142: Screenshot of counsellor registration page.....	121
Figure 143: Screenshot of counsellor registration page with filled data.....	122
Figure 144: Screenshot of counsellor page.....	122
Figure 145: Screenshot of login page.....	123
Figure 146: Screenshot of home page.....	124
Figure 147: Screenshot of login page.....	125
Figure 148: Screenshot of forget password page.....	125
Figure 149: Screenshot of email verification send to user.....	126
Figure 150: Screenshot of reset password page.....	127

Figure 151: Screenshot of sign page.....	127
Figure 152: Screenshot of admin login page.....	128
Figure 153: Screenshot of admin login page with filled data .....	129
Figure 154: Screenshot of admin dashboard. ....	129
Figure 155: Screenshot of cities added on admin. ....	130
Figure 156: Screenshot of cities page.....	131
Figure 157: Screenshot of admin page showing registered pharmacy.....	132
Figure 158: Screenshot of pharmacy viewed by user. ....	132
Figure 159: Screenshot of admin login page.....	133
Figure 160: Screenshot of admin viewing user details. ....	133
Figure 161: Screenshot of admin updating user details. ....	134
Figure 162: Screenshot of registered counsellor. ....	135
Figure 163: Screenshot of admin checking counsellor detail. ....	135
Figure 164: Screenshot of admin checking counsellor document. ....	136
Figure 165: Screenshot of admin viewing user details. ....	137
Figure 166: Screenshot of admin approving user.....	137
Figure 167: Screenshot of admin approving user.....	138
Figure 168: Screenshot of success after approving user. ....	138
Figure 169: Screenshot of pharmacist adding product. ....	139
Figure 170: Screenshot of pharmacist viewing recently added product. ....	140
Figure 171: Screenshot of admin viewing product. ....	140
Figure 172: Screenshot of user viewing product. ....	141
Figure 173: Screenshot of counsellor adding blog. ....	142
Figure 174: Screenshot of recently added blog viewed as a counsellor.....	143
Figure 175: Screenshot of admin viewing blog.....	143
Figure 176: Screenshot of user viewing blog. ....	144
Figure 177: Screenshot of PayPal login. ....	145
Figure 178: Screenshot of payment made with PayPal.....	146
Figure 179: Screenshot of message alert after successful payment. ....	147
Figure 180: Screenshot of login page with different credential.....	148
Figure 181: Screenshot of unsuccessful login.....	149

---

Figure 182: Screenshot of registration user with similar email address.....	150
Figure 183: Screenshot of adding product by pharmacist. ....	152
Figure 184: Screenshot of products added by pharmacist. ....	152
Figure 185: Screenshot of admin viewing product. ....	153
Figure 186: Screenshot of store page with products. ....	153
Figure 187: Screenshot of blog being added by counsellor.....	155
Figure 188: Screenshot of added blogs by counsellor. ....	155
Figure 189: Screenshot of admin viewing blog.....	156
Figure 190: Screenshot of user viewing all blogs. ....	156
Figure 191: Screenshot of admin viewing product/medicine. ....	157
Figure 192: Screenshot of user viewing medicine. ....	158
Figure 193: Screenshot of admin viewing blog.....	159
Figure 194: Screenshot of user viewing blogs.....	159
Figure 195: Screenshot of store page. ....	160
Figure 196: Screenshot of product being removed from the store. ....	161
Figure 197: Screenshot of store page after removing products.....	161
Figure 198: Screenshot of admin page showing products.....	161
Figure 199: Screenshot of blog page. ....	162
Figure 200: Screenshot of blog added by a counsellor. ....	163
Figure 201: Screenshot of empty table representing zero blog. ....	163
Figure 202: Screenshot of blog page. ....	163
Figure 203: Screenshot of admin viewing blogs.....	164
Figure 204: Starting the application.....	164
Figure 205: Home page viewed on chrome.....	165
Figure 206: Home page viewed on Firefox.....	165
Figure 207: PythonAnywhere dashboard. ....	166
Figure 208: Homepage viewed after deployment.....	167

---

## Table of Tables

Table 1: Time and days allocated for planning phase.....	20
Table 2: Time and days allocated for sprint 1.....	21
Table 3: Time and days allocated for sprint 2.....	21
Table 4: Time and days allocated for sprint 3.....	22
Table 5: Time and days allocated for sprint 4.....	22
Table 6: Time and days allocated for sprint 5.....	23
Table 7: Time and days allocated for sprint 6.....	23
Table 8: Time and days allocated for sprint 7.....	24
Table 9: Time and days allocated for sprint 8.....	24
Table 10: Time and days allocated for deployment.....	25
Table 11: List of Hardware Requirements.....	32
Table 12: List of Software Requirement .....	32
Table 13: List of Language Requirements.....	33
Table 14: List of Framework Requirements.....	33
Table 15: List of required libraries.....	34
Table 16: High level Use Case Diagram for user registration.....	46
Table 17: High level Use Case Diagram for pharmacy registration.....	48
Table 18: High level Use Case Diagram for counsellor registration.....	49
Table 19: High level Use Case Diagram for user login.....	51
Table 20: High level Use Case Diagram for admin login.....	53
Table 21: High level Use Case Diagram for add medicine.....	55
Table 22: High level Use Case Diagram for add Blog.....	57
Table 23: High level Use Case Diagram for add to cart .....	59
Table 24: High level Use Case Diagram for make payment.....	61
Table 25: High level Use Case Diagram for add review.....	63
Table 26: High level Use Case Diagram for book session.....	65
Table 27: High level Use Case Diagram for upload prescription.....	67
Table 28: High level Use Case Diagram for Add Cities.....	69
Table 29: Testing on login button 1 .....	88
Table 30: Testing login button 2 .....	90

Table 31: Testing logout button.....	91
Table 32: Testing cart 1.....	93
Table 33: Testing cart 2.....	96
Table 34: Testing cart message 1.....	97
Table 35: Testing cart message 2.....	98
Table 36: Testing cart item increment and decrement. ....	99
Table 37: Testing booking button.....	101
Table 38: Testing pharmacy detail.....	103
Table 39: Counsellor detail.....	105
Table 40: Testing medicine detail.....	106
Table 41: Testing blog page.....	108
Table 42: Testing payment button.....	109
Table 43: Whitebox testing for user registration. ....	111
Table 44: Whitebox testing for pharmacy registration. ....	115
Table 45: Whitebox testing for counsellor registration.....	120
Table 46: Whitebox testing for user login. ....	123
Table 47: Whitebox testing for forget password. ....	124
Table 48: Whitebox testing for reset password. ....	126
Table 49: Whitebox testing for admin login. ....	128
Table 50: Whitebox testing for city page. ....	130
Table 51: Whitebox testing for pharmacy page. ....	131
Table 52: Whitebox testing for viewing user details. ....	132
Table 53: Whitebox testing for viewing user document. ....	134
Table 54: Whitebox testing for user approval. ....	136
Table 55: Whitebox testing for add medicine. ....	139
Table 56: Whitebox testing for add blog. ....	141
Table 57: Whitebox testing for payment. ....	144
Table 58: Validation testing for user login. ....	147
Table 59: Validation testing for user registration. ....	149
Table 60: Integration testing to create operation of medicine. ....	151
Table 61: Integration testing to create operation of blog. ....	154

---

Table 62: Integration testing to read operation of medicine.....	157
Table 63: Integration testing to read operation of blog. ....	158
Table 64: Integration testing to remove operation of medicine.....	160
Table 65: Integration testing to remove operation of blog. ....	162
Table 66: System testing 1.....	164
Table 67: System Testing 2.....	166

---

## 1. Introduction

### 1.1. Project Description

The recent unpredicted pandemic has caused a drastic change in the world. It has jolted the economy, lifestyle and health care system worldwide. Lockdown and mobility restrictions caused inconvenience in various sectors. The challenges and threats caused by COVID-19 in health care system has highly affected the effective delivery of essential health service. The pandemic has highly affected the present world causing the shortage of medicine and other health related products and services. This global outbreak has highlighted a strong need for sustainable and resilient investment in health care system.

Amidst the global mayhem caused by the pandemic; digital technologies have emerged as a key element online social resilience. Since the introduction of internet and online applications, there has been a high revolution in the market, including the health sector. With everything being digitalized, it has been a lot easy for the patients to know about their illness, diseases, and its diagnosis. The demanding need of people to have physicians, pharmacists and counsellor online is meet by digital platforms. The best way to reduce the risk of future pandemics is reducing physical proximity. With the same thought of reduce the future risk, a project named 'Guardian' is introduced. The main aim of this project is to make physical and mental health services accessible in all the areas of Nepal. The physical health services are provided by creating an online store where customers (patients) can place their order of different medicines and health related products and the mental health service is provided by the counselling therapy from a counsellor.

## 1.2. Current Scenario

Although the covid outbreak highlighted the strong need of sustainable and resilient investment in health care system the government of Nepal have put banned on selling and buying an online medicines and drugs.

According to the article posted by KTM2DAY on April 6<sup>th</sup>, 2021, the Department of drug administration of Nepal has put an immediate ban on sale of online drugs. The department has also banned pharmacies selling drugs online through various apps and websites. (News Team, 2021)

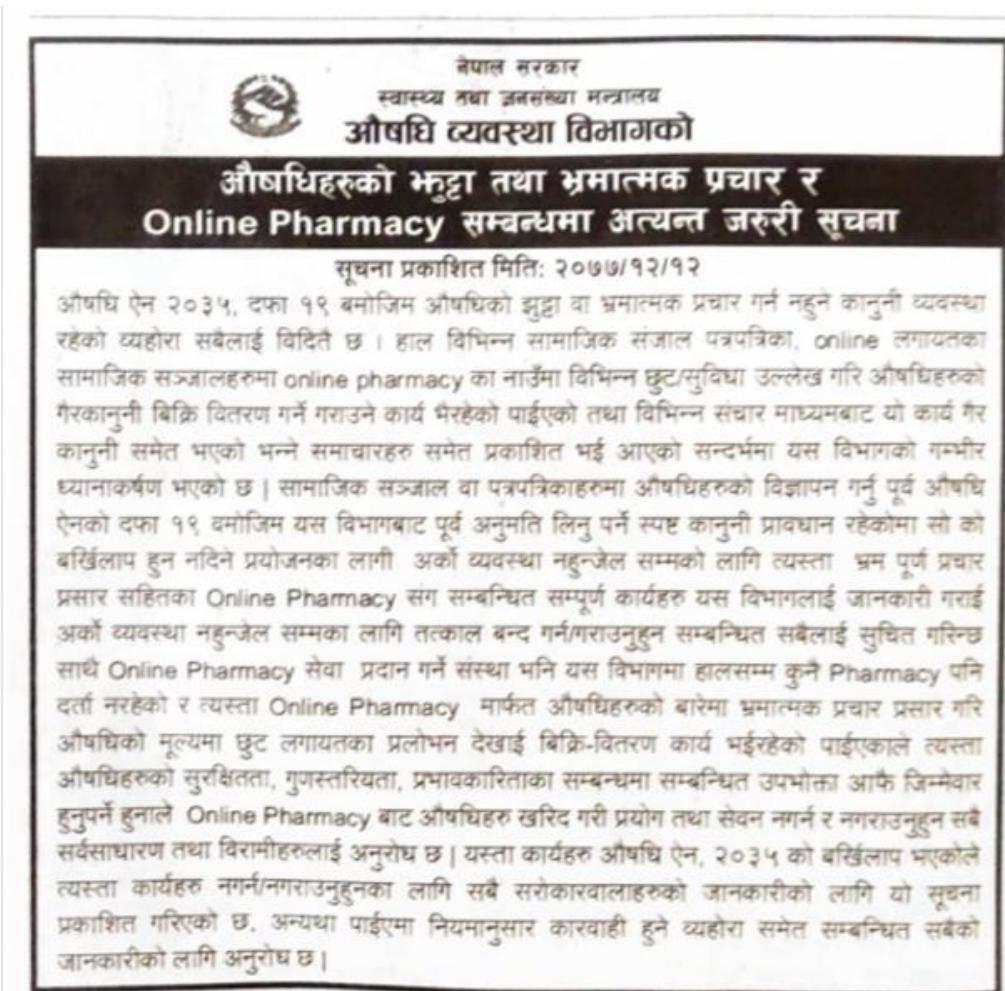


Figure 1: Notice posted by Department of Drug Administration.

### 1.3. Problem Domain

With the number of growing COVID-19 cases and unexpected lockdown within the entire country, it had been most difficult and challenging situation to access the medical health services and medicine. With the strict rules and regulation of less physical proximity and distance maintenance, people were not able to get out of home for any means and purpose. This became a critical problem for the patients and pharmacies to sell their products. The distance gap between patients and pharmacies is hence covered by Online platforms.

Likewise, being in home whole day, staying in quarantine and spending time on digital gadgets had made children, youth, and adult mental health ill. With all these scenarios and acculsive use gadgets, many people get prone to mental sickness. Although social distancing is the most effective way to contain the outspread of this virus, this is not easy to implement for healthcare professionals who require direct contact with COVID-19 patients and puts them under a high risk of being infected themselves. Frontline healthcare professionals were particularly vulnerable during this pandemic owing to their commitment to contain the disease. Many healthcare professionals worked around 17 h, including long tele-counselling shifts each day. Further, healthcare professionals faced acute shortage of masks, hand gloves, and personal protective equipment (PPE) to protect themselves from COVID-19 infection. Healthcare professionals also suffered from insomnia, loneliness, sleep disorder, and mental depression as a result of the workload and related stress. They were experiencing anxiety attacks as well as frustration due to a lack of knowledge, environmental changes, and fear of infection both by themselves and by their family members. (WHO, 2022)

The other problem is that not all people are able to buy their medicines physically in a drug store or pharmacy and attend their therapy session in clinic. For instance, a disabled person or an aged person must be reliable to their caretaker or their family to buy them their medicines and drop them for their therapy session. Additionally, the prescription provided by the doctors are unable to read or understand by patients themselves which creates a lot of difficulties for patients to research and buy medicines without prescription.

#### **1.4. Project as a Solution**

Since the Internet has high potentials to overcome this problem, a web application called "Guardian" cart of health and wealth is brought up for the above problem statement. The Guardian serves as an internet hub for pharmacist, counsellors, and their patients, allowing pharmacists and counsellors to expand their businesses by offering drugs and health advice to their clients. Customers can use the as an online medical store to order, study and take online therapy classes prescribed by their doctors. They can upload their prescription sheet or manually order the medicines. This accessibility and mobility connect clients with pharmacies and counsellor in their area that provide therapy classes. It also helps them to select, order and pay online or offline. Similarly, this reduces the reliance of the elderly or disabled on their caretaker or family members and also helping them to release some stress and convenient to buy things online. Also, it informs patients (clients) about their medicines they are taking. This gives them reference about their medicine intake

#### **1.5. Aims and Objectives**

The main aim of this project is to provides physical and mental health services in all the areas of Nepal. The physical health services are provided by creating and online store where customers can place their orders and the mental health service is provided by the counsellors.

The Objective of the project are listed below:

- Identifying and evaluating existing customers' needs and requirements.
- Confirming the methodology for development.
- Requirements gathering and confirming technical specification.
- Design the website, use case diagram, database design, wireframes, sequence diagram, User Interface Design etc.
- Development of the website.
- Testing and Client review.
- Creating the documentation for the website.
- Using different API to learn and implement the tools and technology needed to finalize the application

## **1.6. Structure of the Report**

### **1.6.1. Background**

The report's background part provides a better understanding of the project by describing and defining the project requirements, application, and end-users. It covers the technical jargon associated with the project used to construct the solution, as well as incorporating and comparing a more comprehensive overview of similar projects and other technical components of the application.

### **1.6.2. Development**

The report's development section describes the many types of methodology, as well as the methodology used for project development. It describes the project development phases and analyses the actions that take place during various periods.

### **1.6.3. Testing and Analysis**

The testing and analysis component of the report covers the testing plan for the application, which includes all types of testing to ensure that the system functions correctly.

### **1.6.4. Conclusion**

The report's conclusion section contains information about the application's ethical, social, and legal issues. It also describes the application's limitations as well as future work to be incorporated in the system.

## 2. Background

### 2.1. End Users

This project proposes a web platform that is used by Pharmacist, counsellors and their customers. This application will be used for buying/selling medicines, connecting with required counsellors, reading health related blogs, knowing the prescriptions and many other reasons. There isn't any single client for this application.

To understand the need of this application survey was carried out among pharmacists, counsellors, health workers and normal peoples. With the help of this survey various forms of problems were collected from them. In this way most of the requirements were gathered for the project which could solve most of their problems

### 2.2. Understanding the Solution

#### 2.2.1. Overview of the system

The system being developed is a Web-based application named “Guardian”. The main objective of the application is to provide a common platform for users, pharmacist and counsellor to buy, sell medicines and perform therapy session. The requirements for the application are determined by the basic features and their needs and by the user feedbacks. According to the research and reviews, the application needs to be well organized and easy to operate. Even if a new user uses the application, he/she should be able to use the application without facing any difficulty in finding the desired content.

#### 2.2.2. Technical terms and definition

##### 2.2.2.1. Integrated development environment

An integrated development environment (IDE) is software for building applications that combines common developer tools into a single graphical user interface (GUI).

An IDE typically consists of:

- Source code editor: A text editor that can assist in writing software code with features such as syntax highlighting with visual cues, providing language specific auto-completion, and checking for bugs as code is being written.
- Local build automation: Utilities that automate simple, repeatable tasks as part of creating a local build of the software for use by the developer, like compiling computer source code into binary code, packaging binary code, and running automated tests.
- Debugger: A program for testing other programs that can graphically display the location of a bug in the original code.

#### **2.2.2.1.1. Visual Studio**

Visual Studio is an Integrated Development Environment (IDE) developed by Microsoft to develop GUI (Graphical User Interface), console, Web applications, web apps, mobile apps, cloud, and web services, etc. With the help of this IDE, you can create managed code as well as native code. It is not a language-specific IDE as you can use this to write code in C#, C++, VB (Visual Basic), Python, JavaScript, and many more languages. (Geeks for Geeks, 2019)

#### **2.2.3. Database**

A database is an organized collection of structured information, or data, typically stored electronically in a computer system. A database is usually controlled by a database management system (DBMS). (oracle, 2021)

##### **2.2.3.1. PostgreSQL**

PostgreSQL is a powerful, open-source object-relational database system that uses and extends the SQL language combined with many features that safely store and scale the most complicated data workloads. (PostgreSQL, 2021)

##### **2.2.3.2. SQLite**

SQLite is an in-process library that implements a self-contained, serverless, zero-configuration, transactional SQL database engine

## 2.3. Similar Projects

Research was carried out before the development of an application over different applications similar to this project. I have taken some design ideas and other architectures from them. Some of the similar projects were-

### 2.3.1. Dhani Pharmacy

Dhani Healthcare Limited is a primary healthcare and pharmacy services provider. Through its Health offering - Dhani Doctor, it provides holistic OPD solutions to patients with instant video consultations by in-house specialists and general practitioners available 24x7. Customers can also avail door-step delivery of prescribed medicines and in-home diagnostic tests. Alongside healthcare consultations, Dhani Health also includes a pharmacy offering that provides medicines and healthcare products. Available on both website and mobile app, medicines are delivered to customers' doorsteps across different cities.

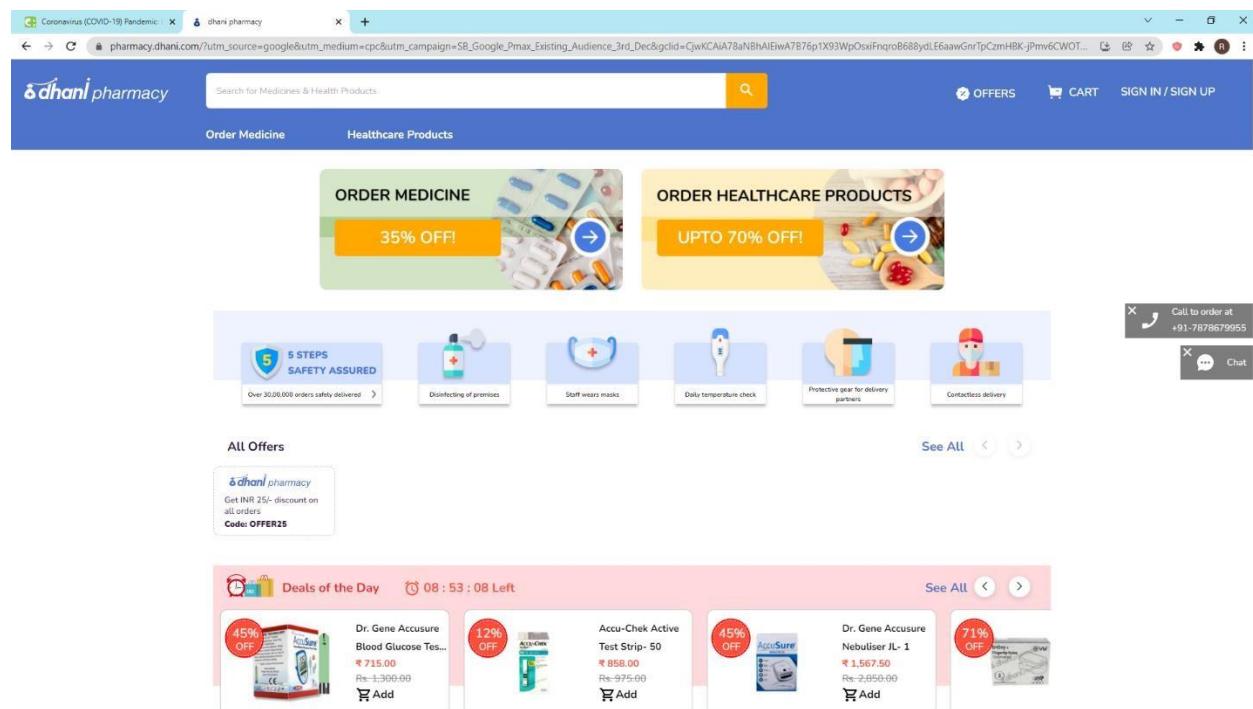
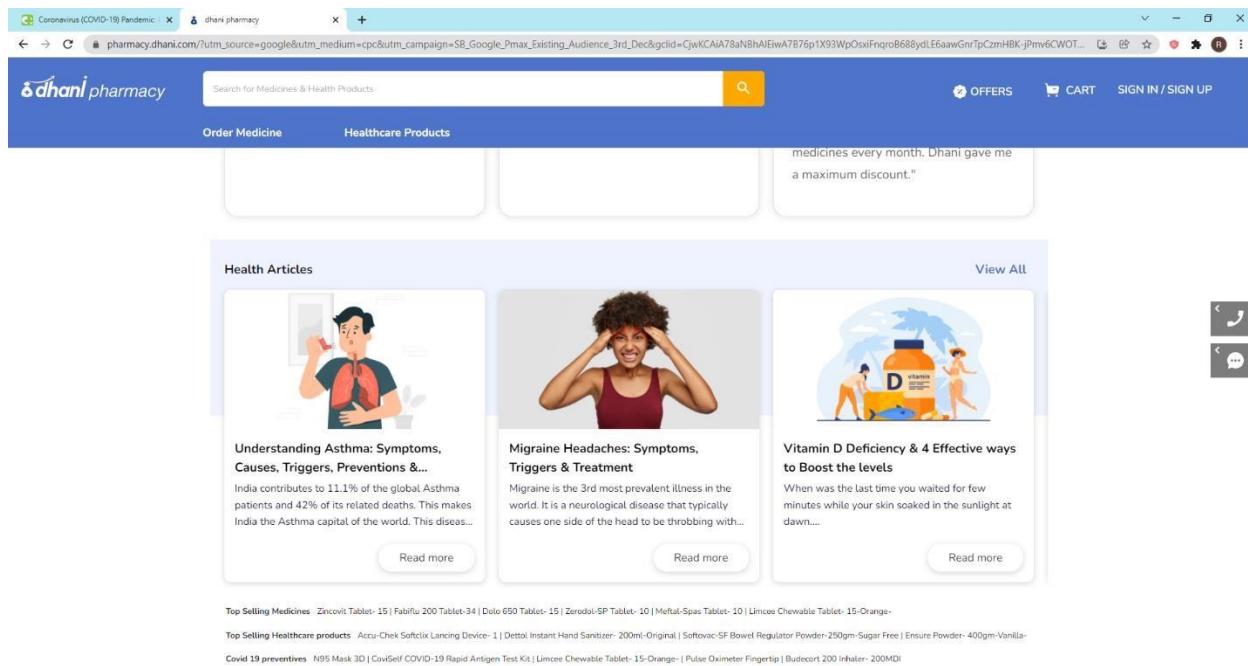


Figure 2: Health care product page of Dhani Pharmacy.

The positive aspect of this application is that it provides primary health care as well as provide pharmacy services. Similarly, they also have video call system to operate with

their patients. No negative aspects in this application, but I would have loved if they had carried out therapy to their patient with video conference as well.

It influences me to develop the idea to including video call schedule. And encouraged me to include health care products as well. (Dhani Pharmacy, 2021)



*Figure 3: Health Article page of Dhani Pharmacy.*

### 2.3.2. Netmeds.com

Netmeds.com is an online portal for medicines where one can upload the prescription, order medicines, and get them delivered at the doorstep. The company came into being in 2015 and since then it is trying to make all the modern medicines available and affordable for all with a hybrid inventory/marketplace model. Netmeds.com takes full responsibility for every order from receiving and verifying prescriptions, to taking payments and then finally sourcing the products.

As per my analysis, netmeds.com have very broad and popular application with positive aspects are it have high user trust, with different features like upload prescription, ordering medicines with the location tracker feature. (Netmeds, 2021)

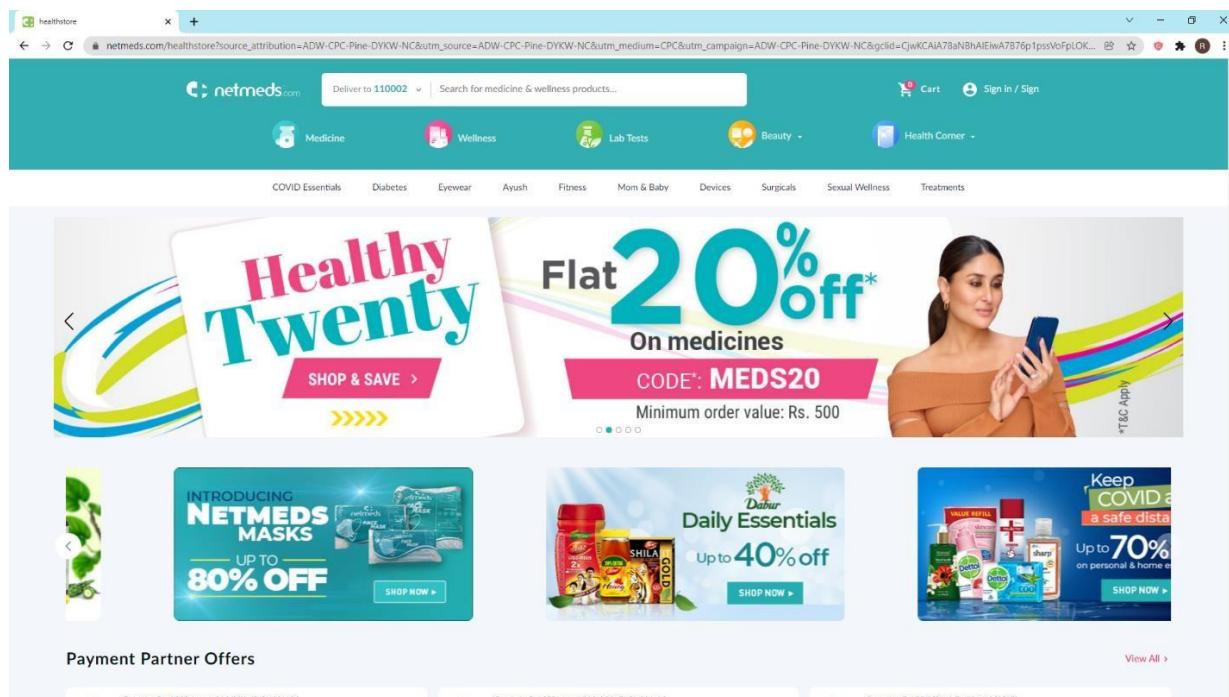


Figure 4: Home page of Netmeds.com

It might help me with the idea of uploading prescription and add health related news into the platform. It encourages me to use blog posts in my project.

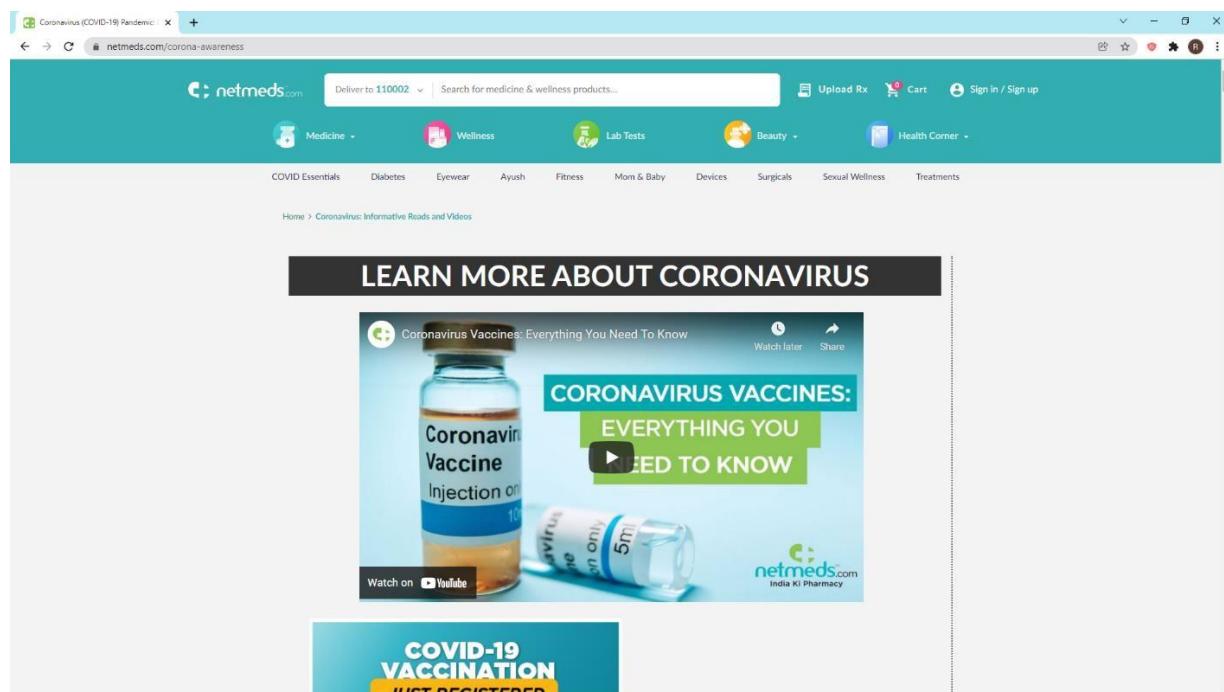


Figure 5: Blog page of Netmeds.com

### 2.3.3. Nepmeds

Nepmeds is a digital healthcare partner with an aim to providing a healing touch to you and your loved ones. Nepmeds help their patients in accessing health issues, getting in touch with the right doctor, booking diagnostic tests, vaccination, getting your required medicines, storing health records, staying fit or learning new ways to live happier. They provide a one-stop solution to a variety of medicines from different manufacturers offering a variety of categories including generic medicines, OTC products, wellness products, vitamins, diet/fitness supplements, Ayurveda products, pain relievers, diabetic care kits, baby/mother care products, beauty care products and surgical consumables.

The positive aspect of this application is that they have collaborated with 400 plus company to provide proper services to the customs. I found a lot of good things in this application, it provides doctor consultation, online laboratory, and wholesale item for small pharmacies. (Nepmeds, 2021)

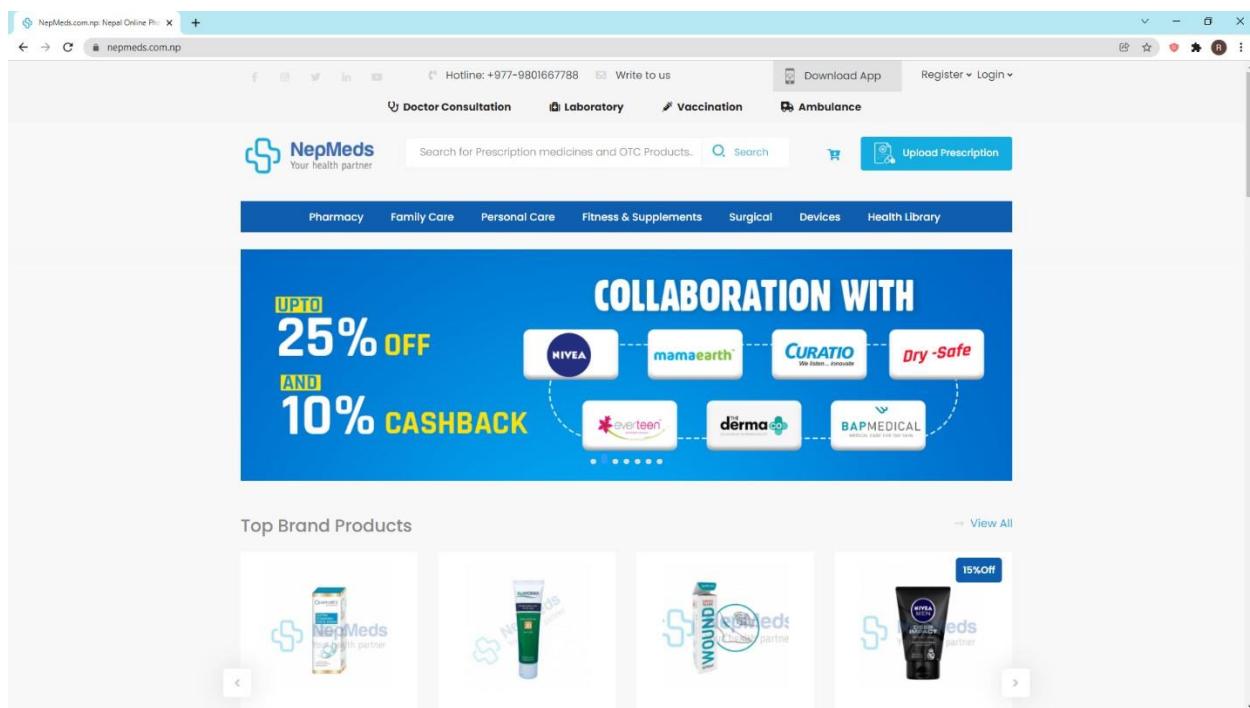


Figure 6: Home page of NepMeds

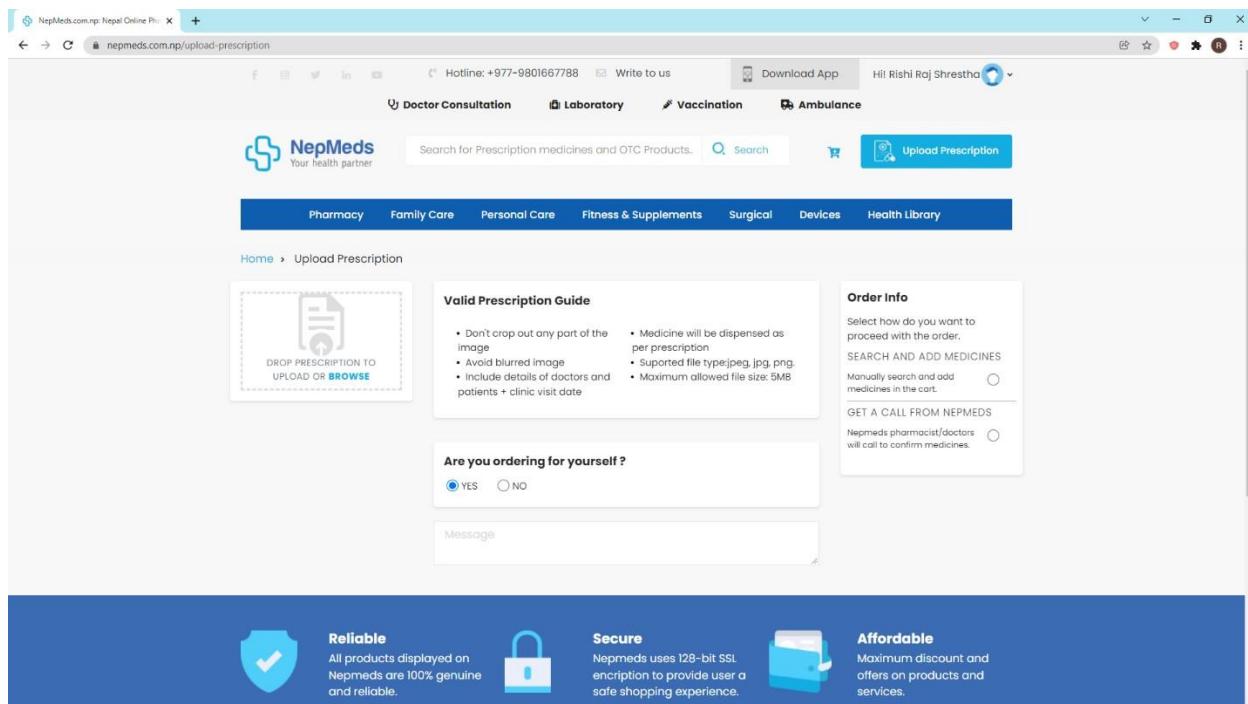


Figure 7: Uploading prescription feature on NetMeds

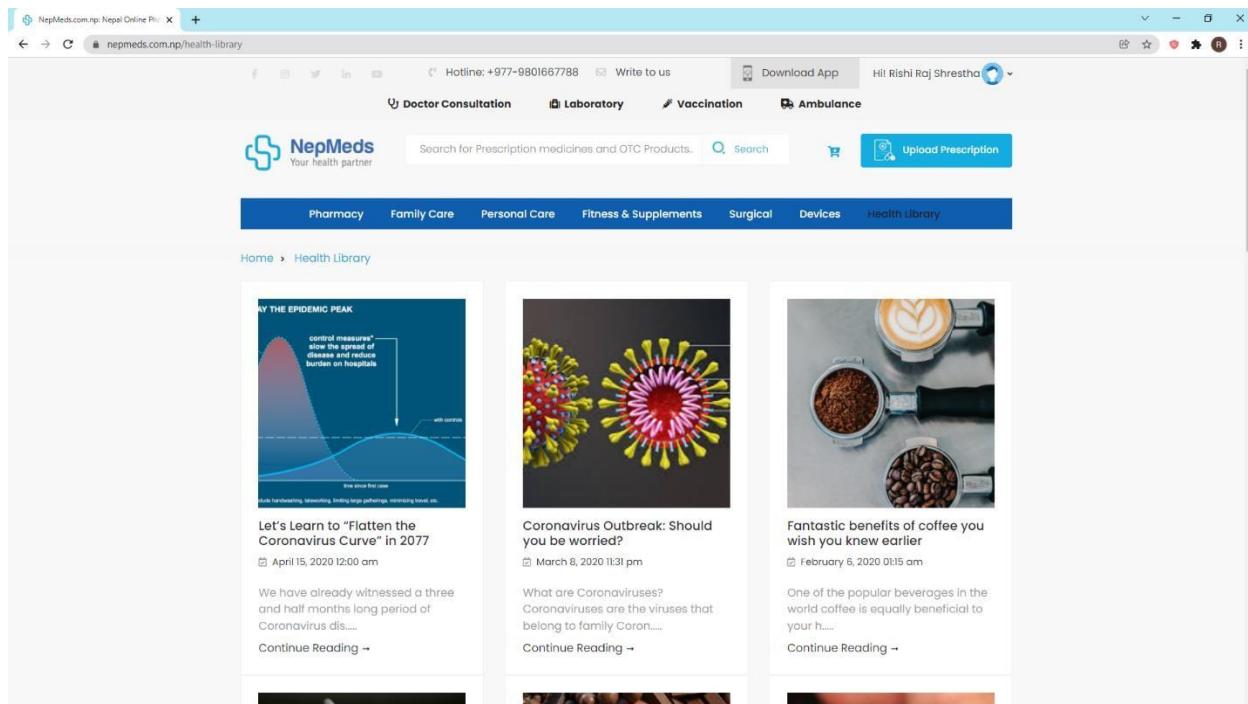


Figure 8: Health related articles.

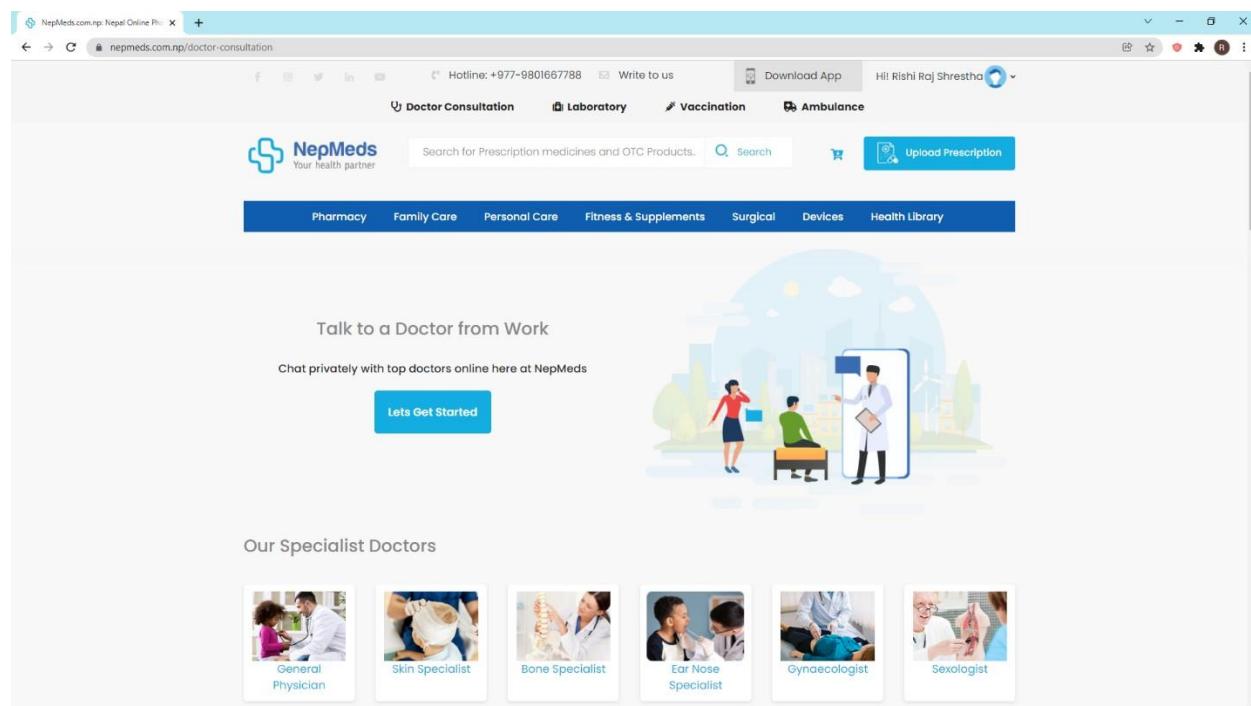


Figure 9: Page to contact specialist Doctors.

## 2.4. Comparisons

S/N	Name	Analysis	Feature taken or can consider in my project
1	Netmeds.com	They have high user trust, with different features like upload prescription, ordering medicines with the location tracker feature.	uploading prescription, Health related news
2	Dhani Pharmacy	They provide primary health care as well as provide pharmacy services. Similarly, they also have video call system to operate with their patients.	Adding category for health care products, Online OPD system.
3	Nepmeds	They have collaborated with 400 plus company to provide proper services to the customs. Similarly, it provides doctor consultation, online laboratory, and wholesale item for small pharmacies.	User interfaces, feature to submit prescriptions, Collaborating with small pharmacies.

### 3. Development

Every Project must select a proper and well-suited methodologies for its development. As a project methodology provides a clear project roadmap that specifies all of the processes necessary to properly complete a project. These project approaches specify the governance structure, process rules, testing activities, procedures, and deliverables. They approach project management in a precise, strict, and repeatable manner.

There are lots of project methodologies out of all those few of them were taken in considered and a single methodology was selected for this project. The Considered methodologies were Waterfall and Extreme Programming Methodologies, and Scrum was the selected methodology.

#### 3.1. Considered Methodologies

##### 3.1.1. Waterfall Methodology

Waterfall methodology is a sequential development process that flows like a waterfall through all phases of a project, with each phase completely wrapping up before the next phase begins.

The Waterfall methodology follows the adage to “measure twice, cut once.” The success of the Waterfall method depends on the amount and quality of the work done on the front end, documenting everything in advance, including the user interface, user stories, and all the features’ variations and outcomes. (Lewis, 2020)

The Waterfall methodology follows a chronological process and works based on fixed dates, requirements, and outcomes. The different phases of waterfall methodology are:

- i. Analysis and Requirement Gathering
- ii. Design
- iii. Development and Implementation
- iv. Testing and Verification
- v. Deployment and maintenance

## The Waterfall Method

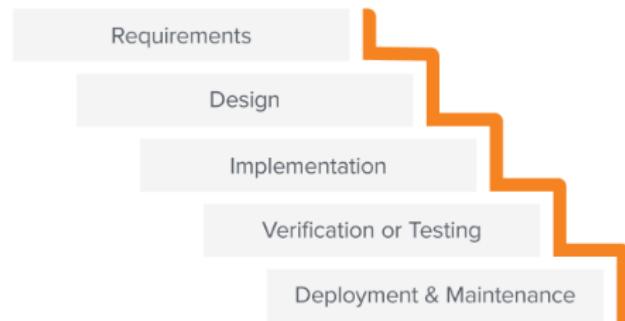


Figure 10: Waterfall Methodology Model

The Reasons for considering Waterfall model were:

- The Waterfall methodology is a straightforward, well-defined project management methodology with a proven track record.
- the requirements are clearly laid out from the beginning, each contributor knows what must be done when.
- Developers can catch design errors during the analysis and design stages, helping them to avoid writing faulty code during the implementation phase.
- With the structured approach, it is easier to measure progress according to clearly defined milestones.

The reason for rejecting waterfall model were: -

- Changes that come further in the process can be time-consuming, painful, and costly.
- Projects can take longer to deliver with this chronological approach than with an iterative one, such as the Agile method.
- Since the frontend requirement is not constant, opening the door to requests for changes and new features later in the process when they're harder to accommodate. (Lewis, 2020)

### 3.1.2. Extreme Programming

Extreme Programming (XP) is a framework of agile that aims to produce higher quality software, and higher quality of life for the development team. XP is the most specific of the agile frameworks regarding appropriate engineering practices for software development. (Agilealliance.org, 2021)

XP is a lightweight, efficient, low-risk, flexible, predictable, scientific, and fun way to develop a software. It also provides specific core practices where –

- Each practice is simple and self-complete.
- Combination of practices produces more complex and emergent behaviour.

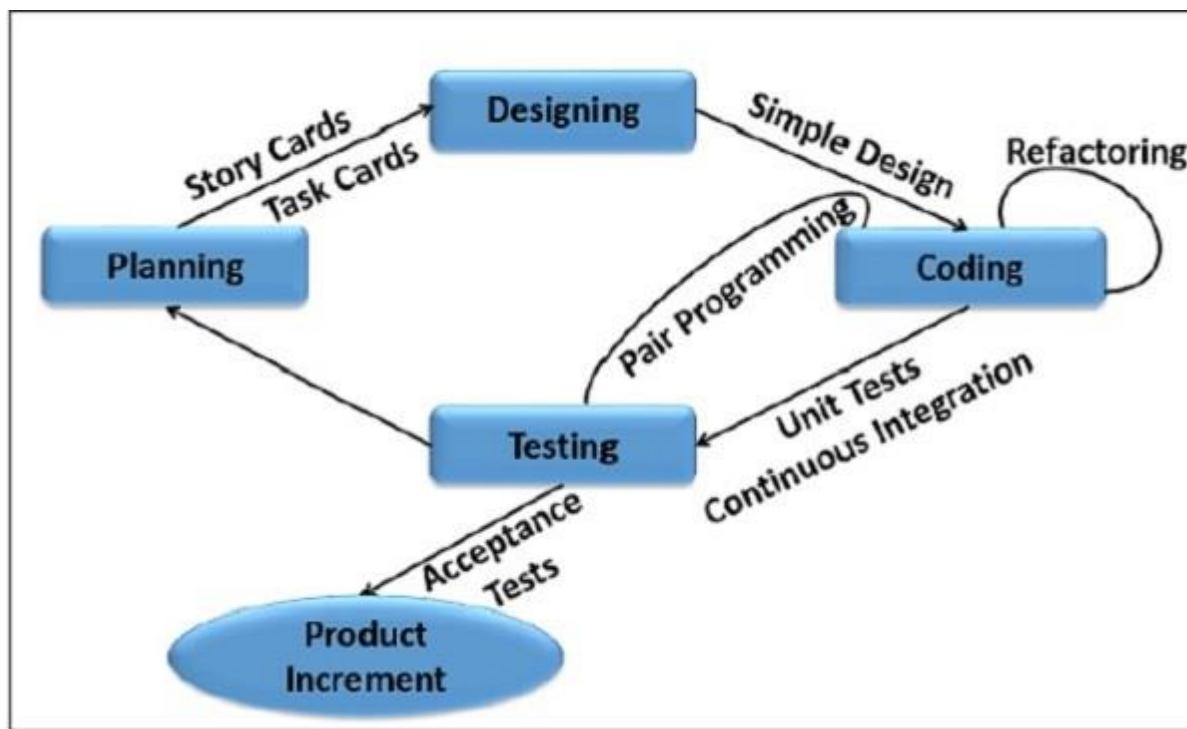


Figure 11: Extreme Programming workflow.

The Reasons for considering Extreme Programming were:

- Simplicity is one more advantage of Extreme Programming projects. The developers who prefer to use this methodology create extremely simple code that can be improved at any moment.

- The whole process in XP is visible and accountable. Developers commit what they will accomplish and show progress.
- Constant feedback is also the strong side. It is necessary to listen and make any changes needed in time.
- XP assists to create software faster thanks to the regular testing at the development stage.
- Extreme Programming contributes increasing employee satisfaction and retention.

The reason for rejecting Extreme Programming were:

- Extreme Programming is focused on the code rather than on design.
- This methodology does not measure code quality assurance. It may cause defects in the initial code.
- The constant changes cannot be documented properly. Thus, there are high risks of unexpected failures that cannot be tracked. (Agilealliance.org, 2021)

### **3.2. Selected Methodology**

#### **3.2.1. Scrum**

Scrum is adaptable, fast, flexible, and effective agile framework that is designed to deliver value to the customer throughout the development of the project. It allows a team to construct complex products in an abstract but nonprescriptive manner. It can assist the team in developing products that meet changing criteria, allowing the product to be improved to its full potential. It's light and simple to learn, but it's difficult to master. Scrum has three team roles, five events, and three artifacts.

Over the course of 30 calendar days, the scrum team generates a sprint. A sprint is equivalent to one product iteration. The product is created through iterative and incremental development. Each sprint begins with sprint planning, during which a development team works with the product owner to designate the item in the product backlog that needs to be developed. (Digite, 2021)

The sprint target is then established. Every sprint's sprint goal is a high-level and concise goal. A plan is developed for transforming the goal into a product increment. Collaboration between the scrum master and the product owner is maintained throughout the sprint, which determines the product increment of development at the conclusion of the sprint. Scrum review is carried out between scrum teams and stakeholders that determines future work based on the results of the review.

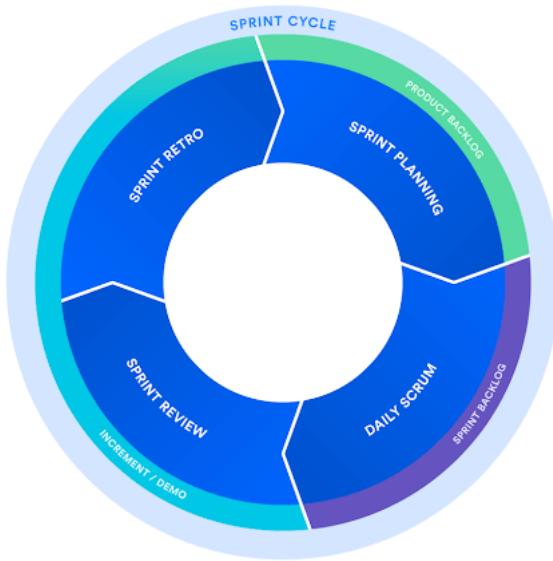


Figure 12: Scrum Cycle

Since the scrum framework also supports 'scrum for one'. This project also follows the 'Scrum for one' approach since it can adapt all components of Scrum but only considers a single individual rather than a team. Its primary purpose is to optimize production within the time constraints by structuring sprint tasks with one individual. Self-reflection and continuous analysis are frequently performed on a daily basis to understand what is going on with the project and how to enhance the workflow and the way things are done. (Digite, 2021)

The reason for selecting scrum for one was because of its simplicity, the rules, artifacts, events, and roles are easy to understand. Its semi-prescriptive approach actually helps remove the ambiguities in the development process,

while giving sufficient space for companies to introduce their individual flavour to it.

### **3.3. Phases of Methodology**

#### **3.3.1. Initiate**

During the initiate phase a project vision is developed and described to the stakeholders or end-users. To understand the complexity of the project research is done based on the problem domain and a survey is conducted among various individuals. The epic, which is a body of work that can later be broken down into specific tasks, will then be developed. After that, the epic would be fine-tuned, and a Product Backlog which contains all the main features of the system would be developed.

#### **3.3.2. Planning**

During this phase, the specifications that had been gathered would be checked and approved. In addition, the project's budget/cost as well as the resources needed would be estimated. Similarly, a task list would be made and estimate how much effort it would require to completing each task on the list in the allotted time.

Finally, a Sprint Backlog would be formed, which would contain all of the tasks that needed to be completed during the sprint.

#### **3.3.3. Design**

During the design process, different required designs required for the project like wireframe, use case Entity relationship diagram would be worked on. This stage will assist in progressing to the implementation stage.

#### **3.3.4. Implementation**

Different task mentioned on the sprint backlog would be carried out in this phase. This phase will make sprint deliveries. This phase takes the longest time during the sprint so a progress track will be made to track of the progress and activities and reflect them when it is necessary.

### **3.3.5. Testing**

During this phase, the task carried out during all the implementation phase is checked. Different testing like functional testing and unit testing are carried out so that the final project would have minimum errors and will meet the aimed target.

### **3.3.6. Review and Retrospect**

During this phase, A demonstrate is done on sprint deliveries and validate them against the project requirements. Self-reflection and continuous analysis are carried out to understand what is going on with the project and how to improve the workflow and the way things are done.

### **3.3.7. Deployment and Documentation**

The project would be deployed in the final phase. To assist a user in using the system, similarly a documentation is made based on the proposal of the project, problem solved, project aims and objectives, its final goals. (Digite, 2021)

## **3.4. Development - Scrum Cycle**

### **3.4.1. Planning**

During this phase, the specifications that was gathered was checked and approved. Similarly online survey was conducted between different types of users to collect their feedbacks and review regarding project aimed features and functions. In addition, that the proper technology and methodology was selected, and plan was made to perform the future tasks.

Finally, different diagram was formed, which would contain all of the tasks that needed to be completed during the sprint.

*Table 1: Time and days allocated for planning phase.*

S/N	Task	Estimated Time
1	Project Planning	11 days

The above table shows the time allocation for the planning phase.

### 3.4.2. Sprint 1

Since this was the first sprint cycle, according to the plan here project designing must be done. As a result, different diagrams were made like the project wireframe and the use case diagrams.

*Table 2: Time and days allocated for sprint 1*

S/N	Task	Estimated Time (Days)
1	Sprint Planning	2 days
2	Sprint Backlog	2 days
3	Designing	8 days
4	Review	2 days
5	Retrospective	2 days

The above table shows the time allocation for the sprint 1.

### 3.4.3. Sprint 2

During this sprint cycle the user interface was created. The user interfaced designed was based on the wireframe designed during the sprint 1. Similarly, Individual use case was taken and its was analysed and diagram like High level diagram, collaboration diagram and sequence diagram were created.

*Table 3: Time and days allocated for sprint 2*

S/N	Task	Estimated Time (Days)
1	Sprint Planning	3 days
2	Sprint Backlog	1 days
3	Frontend	23 days
4	Review	1 days
5	Retrospective	2 days

The above table shows the time allocation for the sprint 2.

### 3.4.4. Sprint 3

During this phase research was done to know more about the Django framework and its libraries and its ability to make dynamic website. Research was done on security aspects, its development, query sets and its deployment. With all those research and knowledge gained Python project was made and Django was imported, and a new Django project named FYP was created.

Likewise, the system was tested initially, and security key and authorized users were added and finally the project was pushed to GitHub.

*Table 4: Time and days allocated for sprint 3*

S/N	Task	Estimated Time (Days)
1	Sprint Planning	2 days
2	Sprint Backlog	1 days
3	Implement Django	23 days
4	Review	1 days
5	Retrospective	2 days

The above table shows the time allocation for the sprint 3

### 3.4.5. Sprint 4

As the Django takes templates and static file works differently all the UI created on sprint 2 was imported into the system. With the help of Django documentation location of static files and template files were managed and the UI was reorganized.

*Table 5: Time and days allocated for sprint 4*

S/N	Task	Estimated Time (Days)
1	Sprint Planning	3 days
2	Sprint Backlog	2 days
3	Import Frontend (Template)	23 days
4	Testing	7 days
5	Review and Retrospective	2 days + 2 days

The above table shows the time allocation for the sprint 4.

### 3.4.6. Sprint 5

There are multiple ways to create user authentication on Django. During this sprint the user authentication and authorization was handled.

*Table 6: Time and days allocated for sprint 5*

S/N	Task	Estimated Time (Days)
1	Sprint Planning	2 days
2	Sprint Backlog	1 days
3	User Authentication	8 days
4	Testing	3 days
5	Review	1 days
6	Retrospective	1 days

The above table shows the time allocation for the sprint 5.

### 3.4.7. Sprint 6

During this sprint research was done regarding the stable database for the Django project during the development phase and deployment phase. As per the result two database was setup SQLite for development and PostgreSQL for deployment.

The models created on sprint 5 for users' authentication was also managed.

*Table 7: Time and days allocated for sprint 6*

S/N	Task	Estimated Time (Days)
1	Sprint Planning	1 days
2	Sprint Backlog	1 days
3	Database Construction	11 days
4	Testing	2 days
5	Review	1 days
6	Retrospective	2 days

The above table shows the time allocation for the sprint 6.

### 3.4.8. Sprint 7

As the database was constructed all the models on the database was not properly managed and well maintained. During this sprint, the models (also known as relation) were connected to each other which were necessary. And the testing was done if the data from the frontend is stored onto the database or not.

*Table 8: Time and days allocated for sprint 7*

S/N	Task	Estimated Time (Days)
1	Sprint Planning	1 days
2	Sprint Backlog	1 days
3	Mapping Database	1 days
4	Testing	10 days
5	Review	1 days
6	Retrospective	1 days

The above table shows the time allocation for the sprint 7.

### 3.4.9. Sprint 8

During this sprint different testing was conducted. The testing conducted is given on testing section of this document.

*Table 9: Time and days allocated for sprint 8*

S/N	Task	Estimated Time (Days)
1	Sprint Planning	1 days
2	Sprint Backlog	1 days
3	Testing	4 days
4	Review	1 days
5	Retrospective	2 days

The above table shows the time allocation for the sprint 8

### 3.4.10. Deployment

During this phase the popular hosting site for any python project was used i.e., python anywhere to host the application.

*Table 10: Time and days allocated for deployment.*

S/N	Task	Estimated Time
1	Website Deployment	7 days

The above table shows the time allocation for the deployment phase.

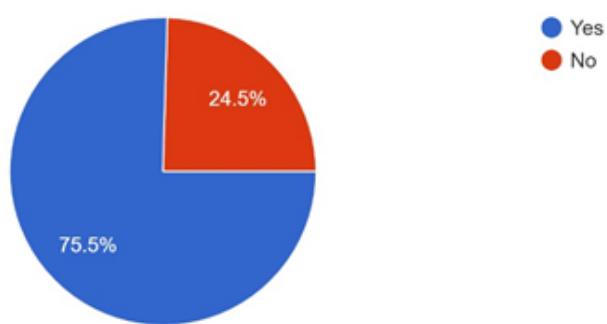
## 3.5. Survey Result

### 3.5.1. Pre-Survey Results

A survey was conducted among pharmacists, counsellors, health workers, and ordinary people to determine the necessity for this application. This survey was used to collect various types of problems from them. In this manner, the majority of the project's requirements were obtained, which could solve the majority of their problems. The following is a summary of the problems and their accompanying responses from the survey participants similarly the detail of survey is present on appendix.

Have you ever been in such a situation where you felt the need of online platform for/your pharmacy?

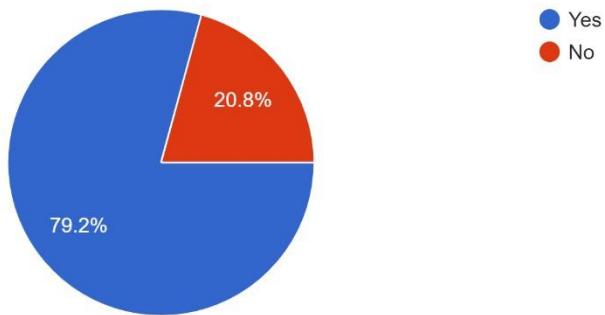
53 responses



*Figure 13: Pie Chart showing the need of online pharmacy.*

Have you ever been in such a situation where you felt the need of online platform to find out counsellor/Reach out to people(as counsellor)?

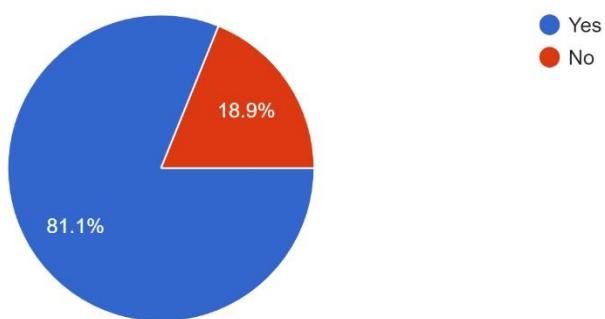
53 responses



*Figure 14: Pie Chart showing the need of online counsellor/people.*

When we were grounded because of covid lockdown do you think people needed someone to cheer them up like counsellors with whom they can ...ow they felt and release all their hidden words ?

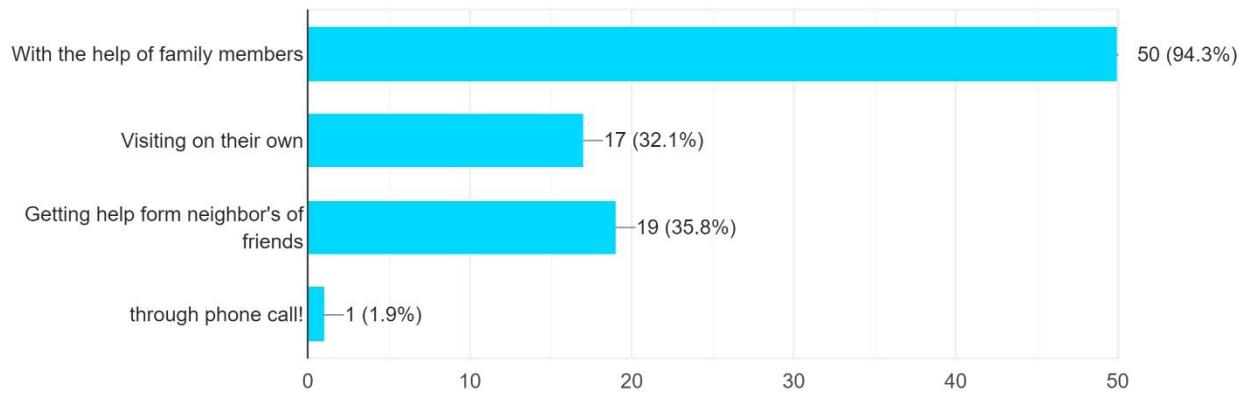
53 responses



*Figure 15: Pie Chart showing the need of counsellor during lockdown.*

Do you have any idea how aged people buy medicines or have a therapy sessions?

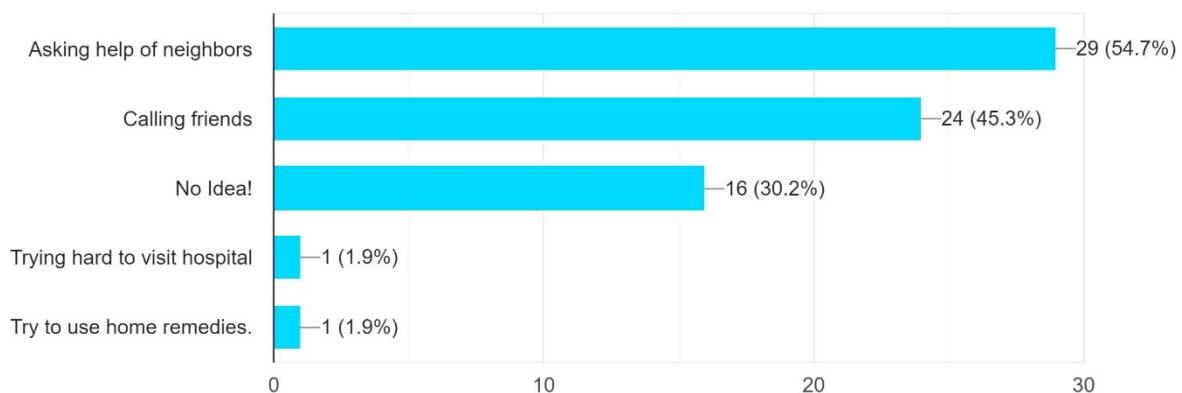
53 responses



*Figure 16: Bar graph showing the response thought on aged people.*

If you are home alone and you got sick then how will you buy medicines?

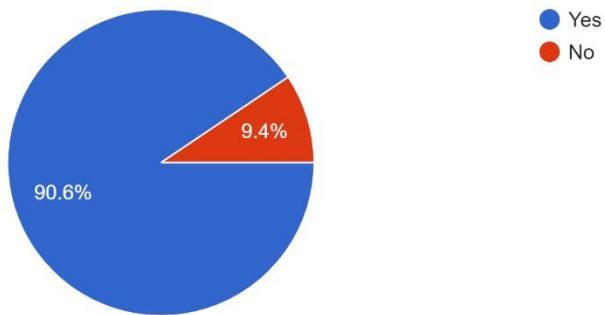
53 responses



*Figure 17: Bar graph showing the response opinion during uncertain medical emergency.*

Have you ever felt difficulties in understanding the prescriptions given by doctors and invoice given by pharmacist?

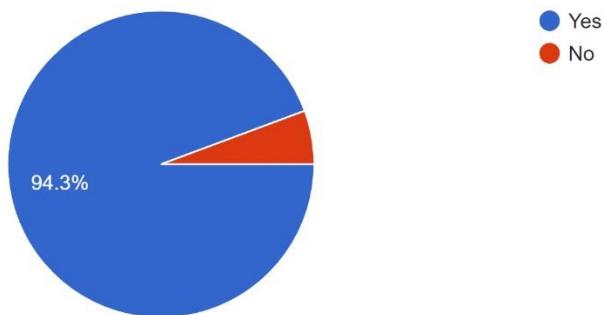
53 responses



*Figure 18: Pie Chart showing the percentage of difficulties faced by response on understanding the prescription.*

Do you think an online pharmacy with mentioned features would be better to solve the problems (mentioned above) ?

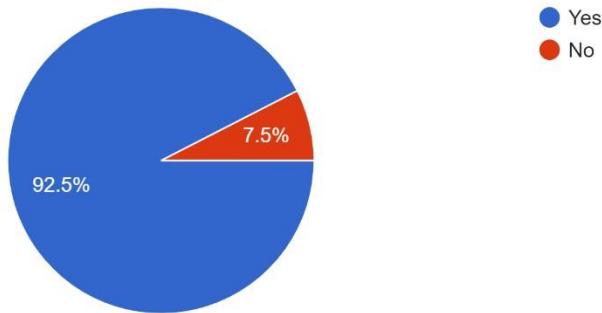
53 responses



*Figure 19: Pie Chart showing the interest of response on online platform for pharmacy.*

Do you think an online Counsellor with mentioned features would be better to solve the problems (mentioned above) ?

53 responses



*Figure 20: Pie Chart showing the interest of response on online platform for counsellor.*

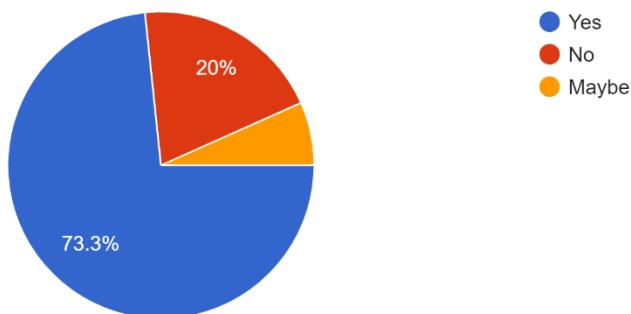
### 3.5.2. Post-Survey Results

The post survey was conducted after the project was completed and was deployed.

The result of post survey is shown below:

Do you find this application reliable and trustworthy?

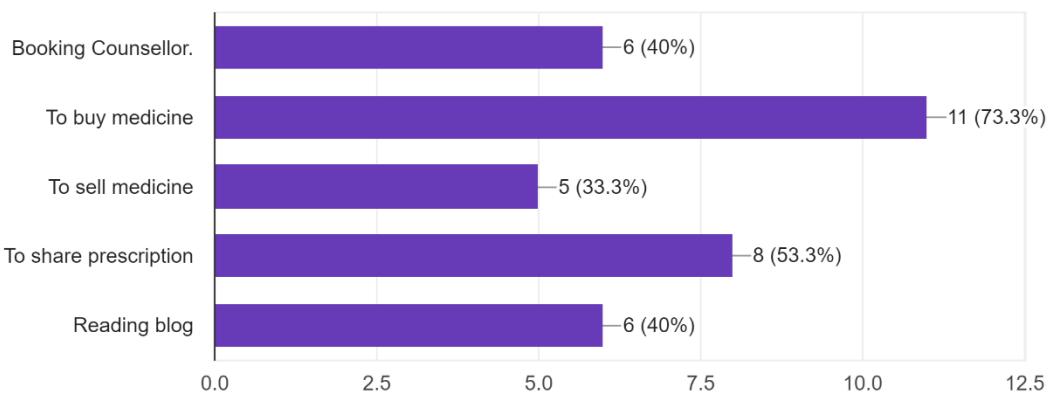
15 responses



*Figure 21: Post Survey result 1.*

For what purpose do you think you will use this application for?

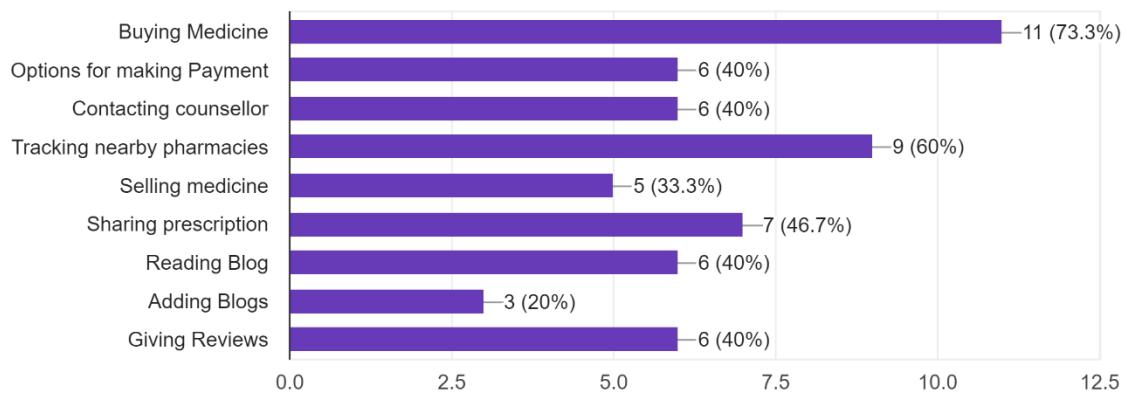
15 responses



*Figure 22: Post Survey result 2*

Select the feature you found useful in the application.

15 responses



*Figure 23: Post survey result 3*

### **3.6. Requirement Analysis**

#### **3.6.1. Features Requirement**

The functions and features listed in the application are based on the users' feedbacks and basic requirements. The features required are as follows:

##### **3.6.1.1. Register**

The user of this application can be registered as a normal user or membership user.

##### **3.6.1.2. Login**

If the user is registered, they can directly log in to the application.

##### **3.6.1.3. Buy**

The web application will have a buy. Using this feature users can get access to medicine easily.

##### **3.6.1.4. Book**

The web application will have a booking feature as well. Using this feature users can set meeting with counsellor

##### **3.6.1.5. Online Payment**

The users will also be provided with an online payment option. Using this feature the users can shop conveniently.

##### **3.6.1.6. Shop details**

The web application also contains a short description of the pharmacy along with its location and opening hours. Users can refer to this to find about the pharmacy details.

##### **3.6.1.7. Review**

A user can also give a review about a product they have bought. Other users can view these reviews to decide which books to rent or buy.

### 3.6.2. Hardware Requirement

*Table 11: List of Hardware Requirements.*

S/N	Name	Require for
1	Personal Computer/Laptop	Laptop computer, or notebook computer is a small, portable personal computer (PC) with a screen and alphanumeric keyboard.

### 3.6.3. Software Requirement

*Table 12: List of Software Requirement*

S/N	Name	Require for
1	Visual Studio	Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs, as well as websites, web apps, web services and mobile apps.
2	Chrome/Fire fox	Google Chrome is a cross-platform web browser developed by Google/ Firefox, is a free and open-source web browser developed by the Mozilla Foundation and its subsidiary, the Mozilla Corporation.
3	Postman	Postman is an API platform for building and using APIs. Postman simplifies each step of the API lifecycle and streamlines collaboration so you can create better APIs
4	PostgreSQL	PostgreSQL is a powerful, open-source object-relational database system that uses and extends the SQL language combined with many features that safely store and scale the most complicated data workloads.

5	DB Browser (SQLite)	SQLite is an in-process library that implements a self-contained, serverless, zero-configuration, transactional SQL database engine
6	Draw.io	Draw.io is an open-source technology stack for building diagramming applications, and the world's most widely used browser-based end-user.

### 3.6.4. Programming Language Requirement

Table 13: List of Language Requirements.

S/N	Name	Require for
1	Python	Python is an interpreted, object-oriented, high-level programming language with dynamic semantics.
2	JavaScript	JavaScript is a lightweight, interpreted, object-oriented language with first-class functions, and is best known as the scripting language for Web pages, but it's used in many non-browser environments as well.

### 3.6.5. Web Application Framework Requirement

Table 14: List of Framework Requirements.

S/N	Name	Require for
1	Django	Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design.
2	Hyper Text Markup Language	Hypertext Markup Language (HTML) is the basic scripting language used by web browsers to render pages on the world wide web.

3	Cascading Style Sheets	Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML.
---	------------------------	--

### 3.6.6. Libraries Requirement

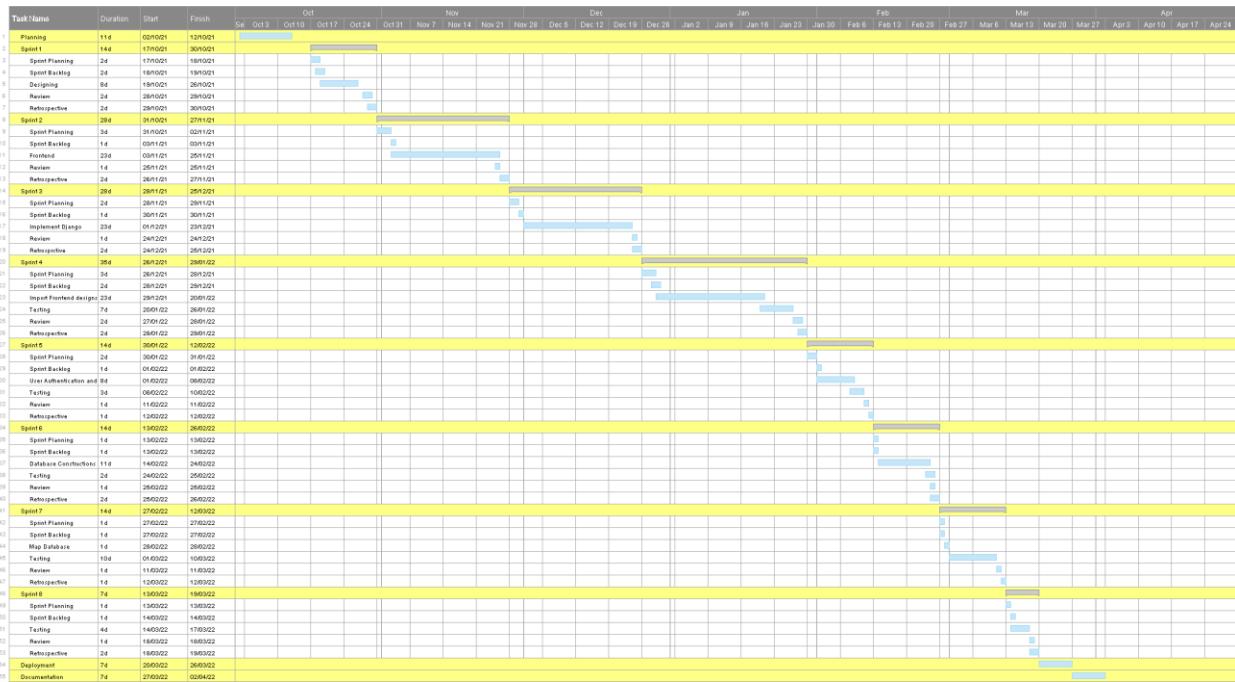
Table 15: List of required libraries.

S/N	Name	Require for
1	Environ	Django-environ allows you to utilize 12factor inspired environment variables to configure your Django application.
2	Google API	Google APIs are application programming interfaces (APIs) developed by Google which allow communication with Google Services and their integration to other services
3	Froala Editor	Froala Editor is a lightweight WYSIWYG HTML Editor written in JavaScript that enables rich text editing capabilities for an application.

### 3.7. Design

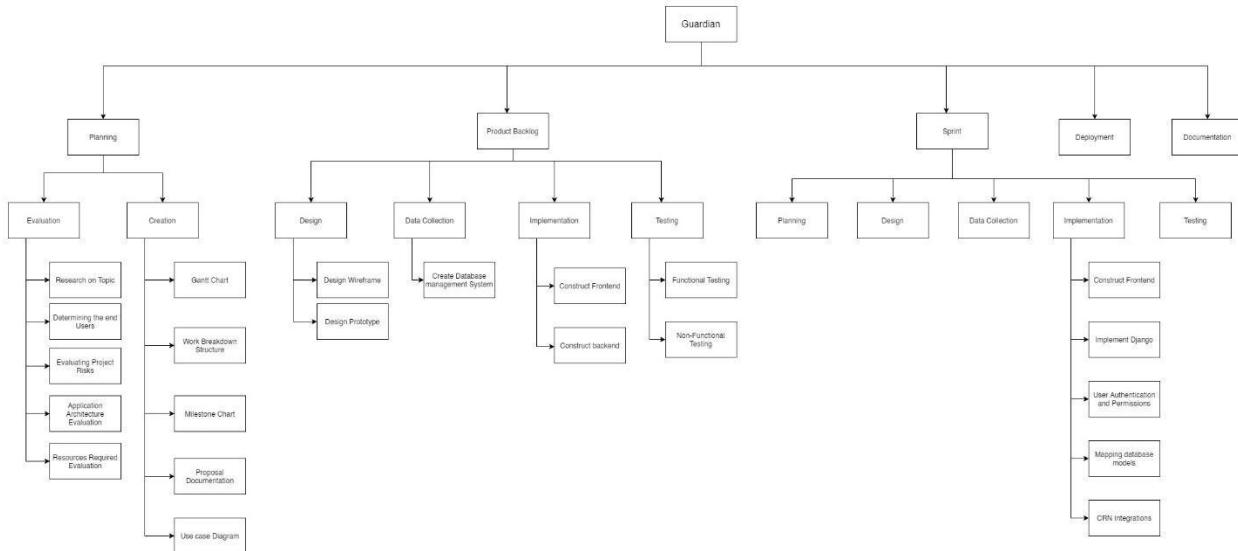
### **3.7.1. Planning Phase**

### 3.7.1.1. Gantt Chat



*Figure 24: Gantt Chat for the project*

### **3.7.1.2. Work breakdown**



*Figure 25: Project Work break down*

### 3.7.1.3. Mind Map

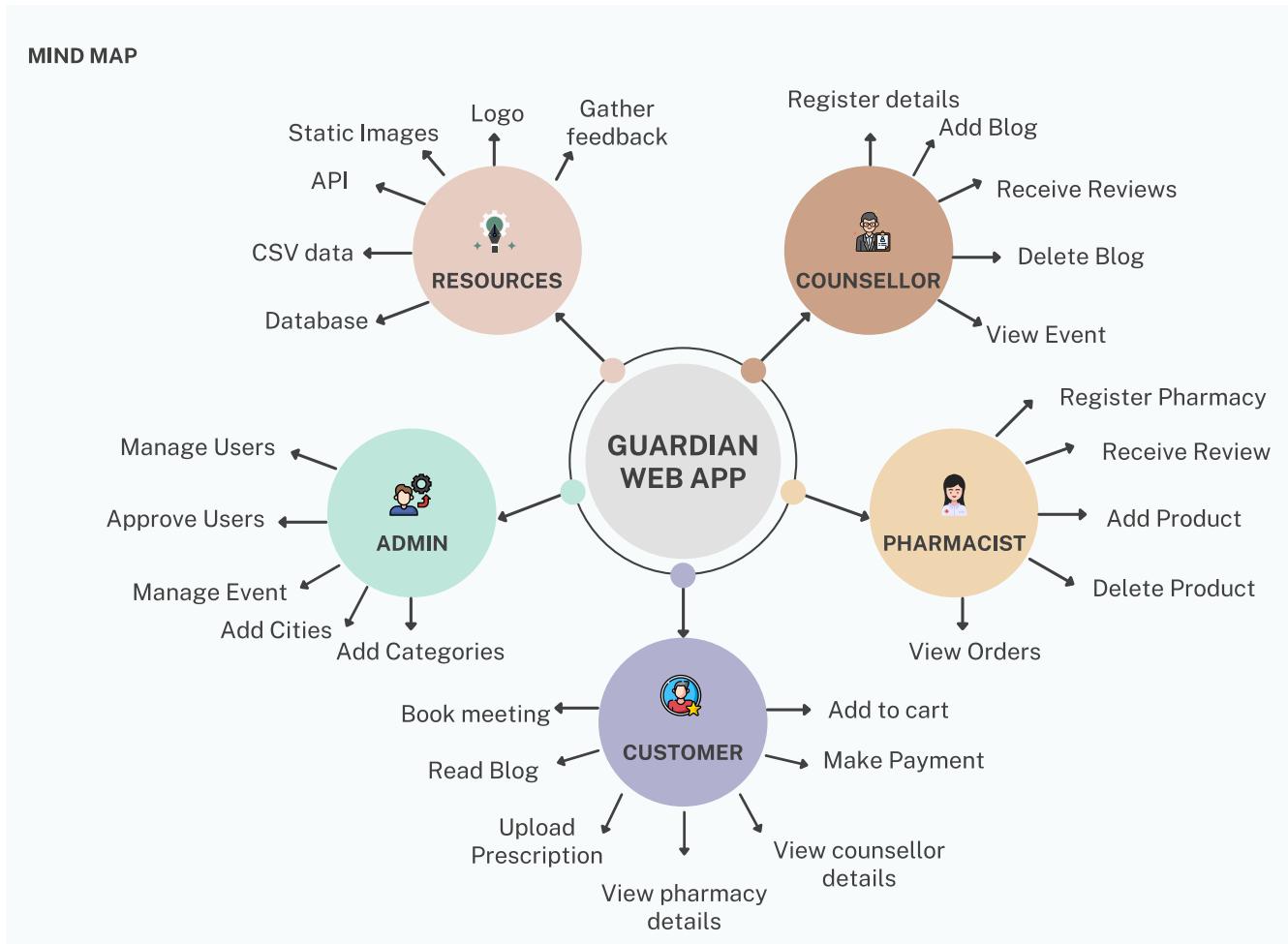


Figure 26: Mind Map of the project

### 3.7.2. Sprint 1

#### 3.7.2.1. Use Case Diagram

A use case diagram is used to summarize details of a system and their users within that system. It is shown as a graphic depiction of interactions among different elements in a system.

A use case diagram looks something like a flowchart, where intuitive symbols represent the system elements. The Initial use case designed for the project is attached below:

### 3.7.2.2. Wireframes

A wireframe is a layout of an application that demonstrates what interface elements will exist on key pages. It is a critical part of the interaction design process. The wireframe for this application is attached below:

#### 3.7.2.2.1. Wire frame for Home Page

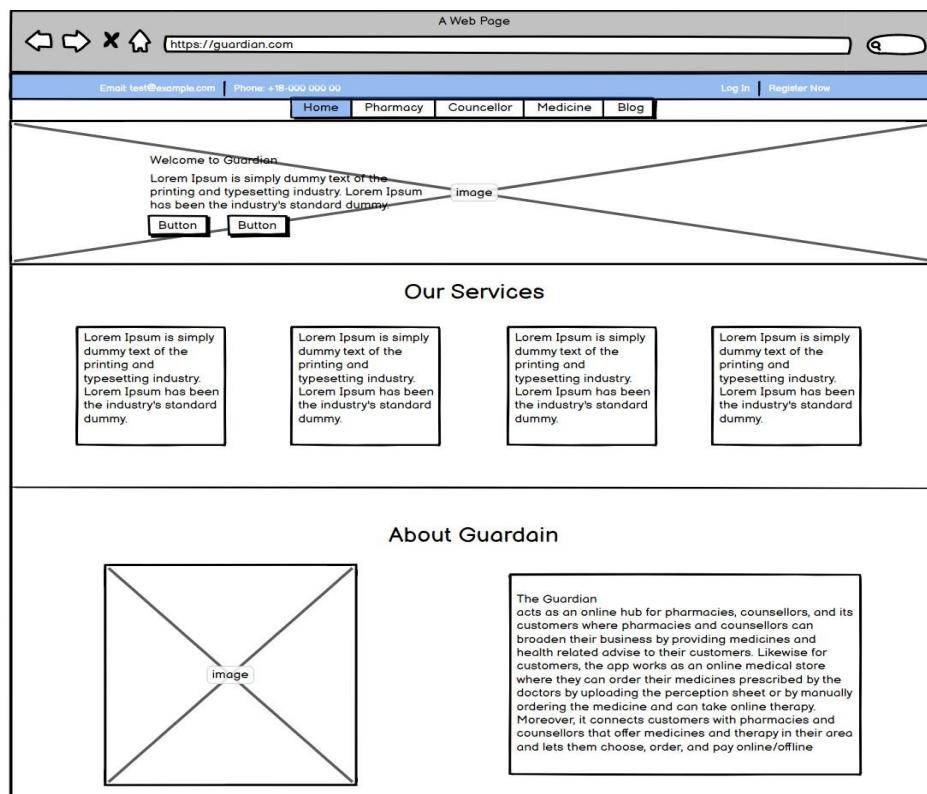


Figure 27: Screenshot of home page wireframe.

The image above is the wireframe for the home page. It was designed using online web application called Balsamiq.

### 3.7.2.2.2. Wire frame for Login Page

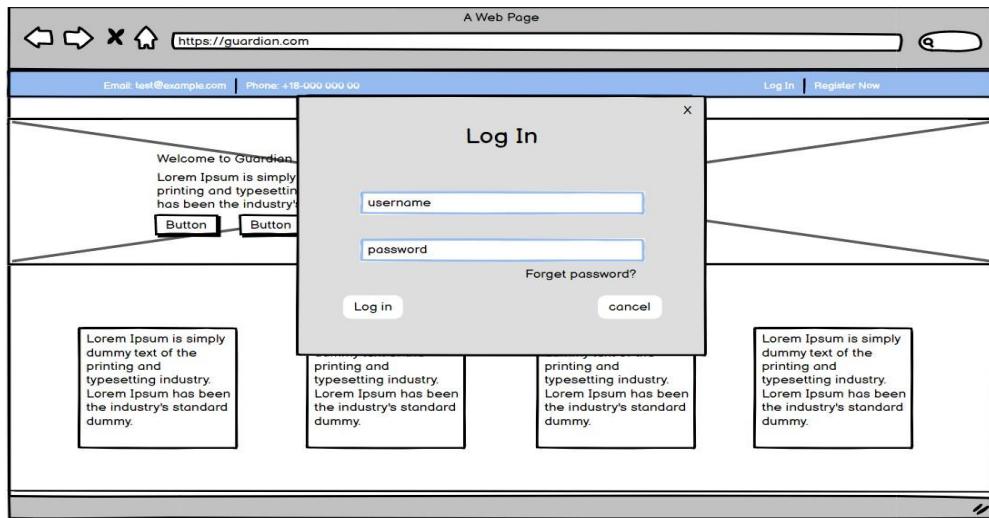


Figure 28: Screenshot of login page wireframe.

The image above is the wireframe for the login page. It was designed using online web application called Balsamiq.

### 3.7.2.2.3. Wire frame for Pharmacy Page

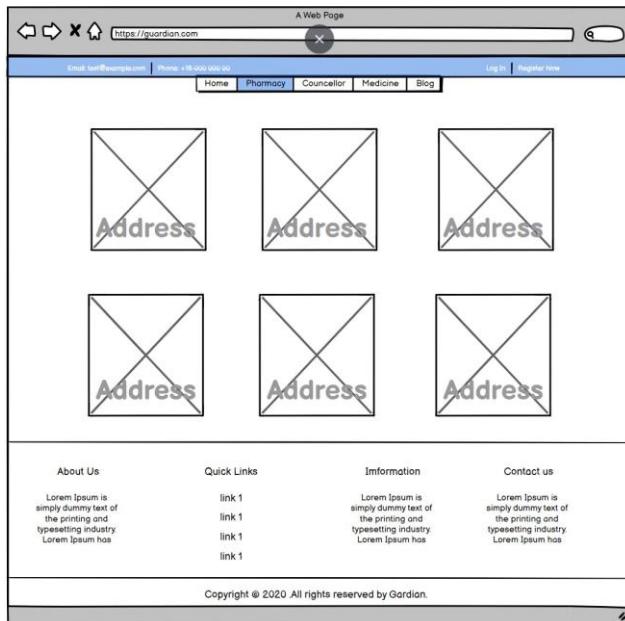
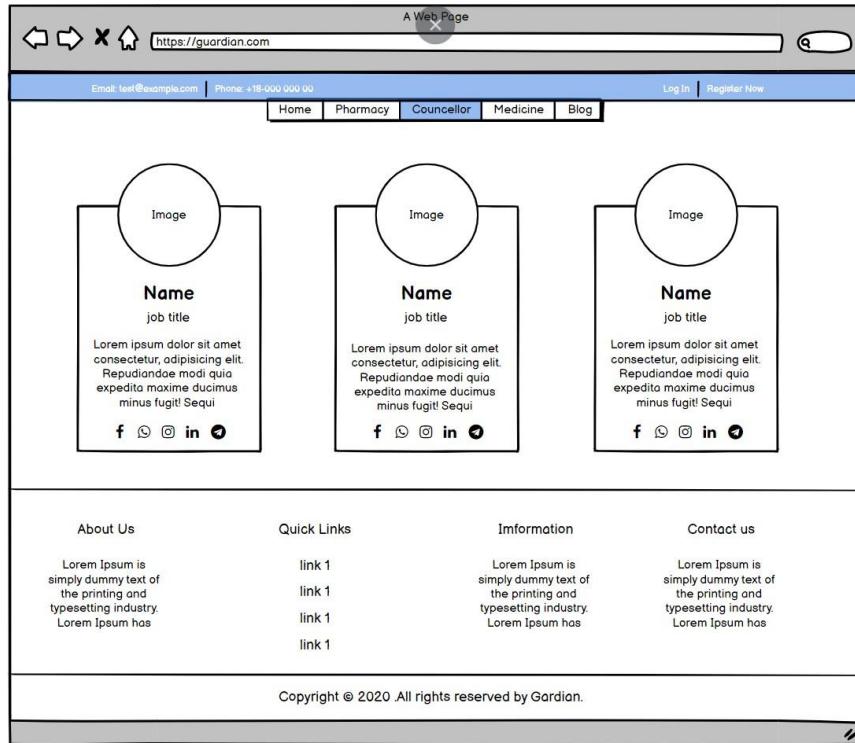


Figure 29: Screenshot of pharmacy page wireframe

The image above is the wireframe for the pharmacy page. It was designed using Balsamiq.

### 3.7.2.2.4. Wire frame for Counsellor Page



*Figure 30: Screenshot of counsellor page wireframe*

The image above is the wireframe for the counsellor page. It was designed using online web application called Balsamiq.

### 3.7.2.2.5. Wire frame for Detail Page

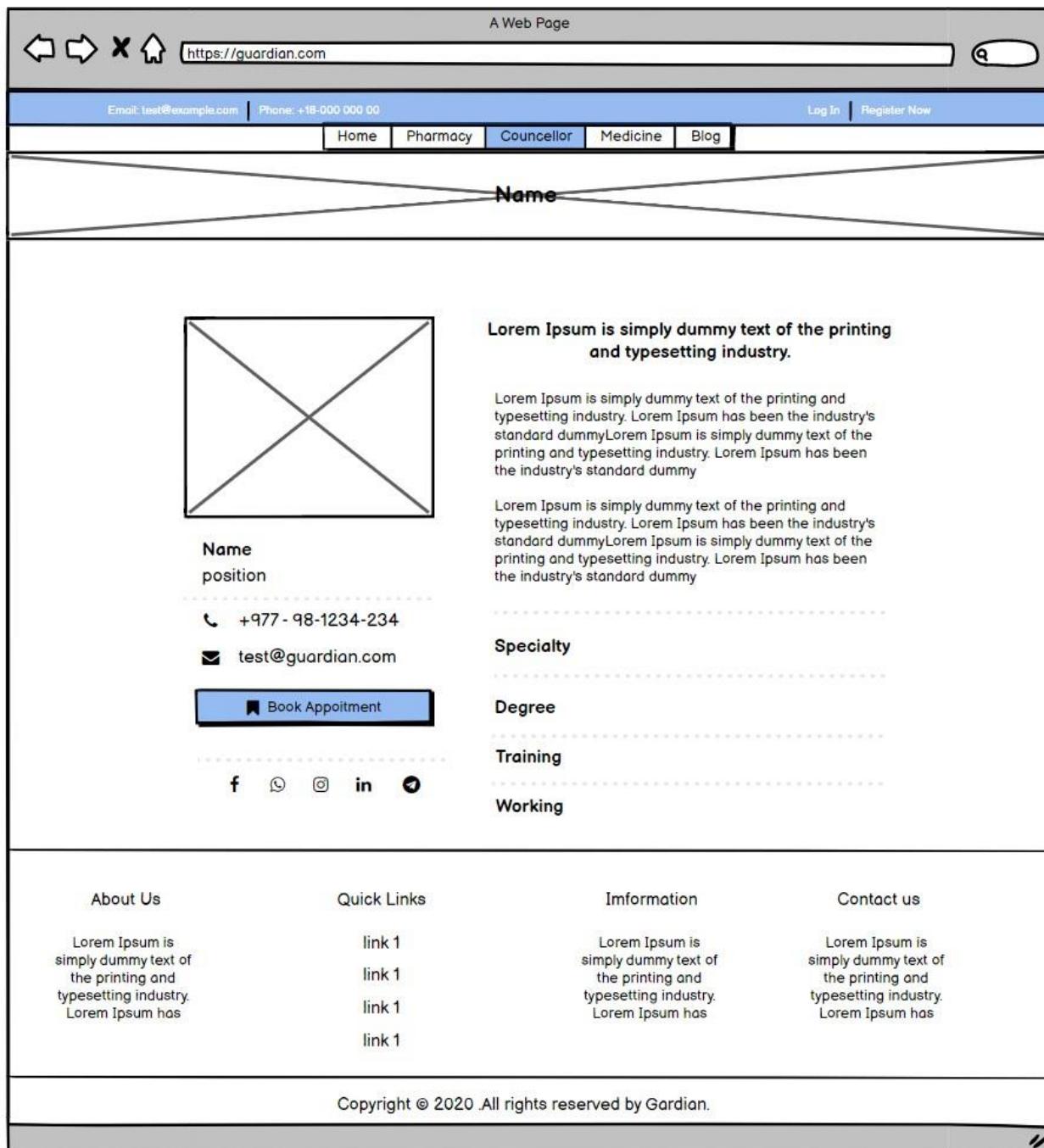


Figure 31: Screenshot of counsellor detail page wireframe

The image above is the wireframe for the counsellor detail page. It was designed using Balsamiq.

### 3.7.2.2.6. Wire frame for Search Medicine Page

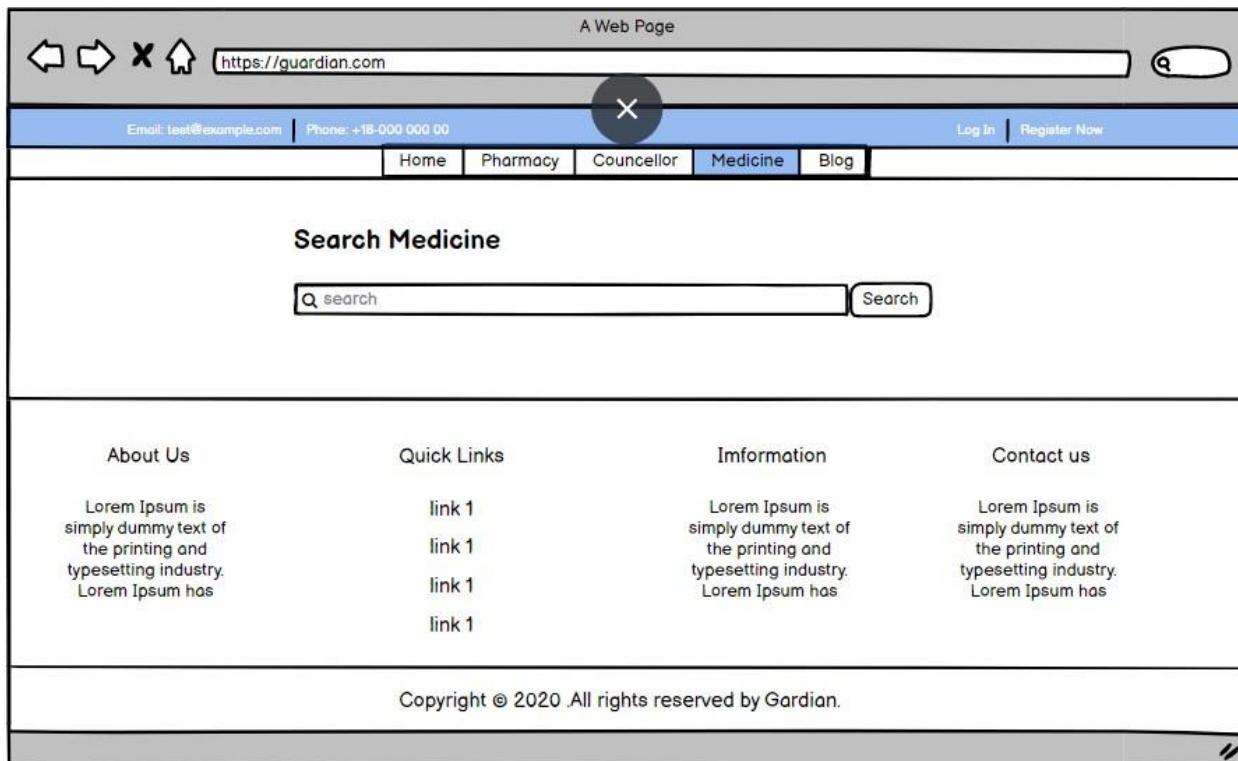


Figure 32: Screenshot of search page wireframe

The image above is the wireframe for the search page. It was designed using online web application called Balsamiq.

### 3.7.2.2.7. Wire frame for Detail page

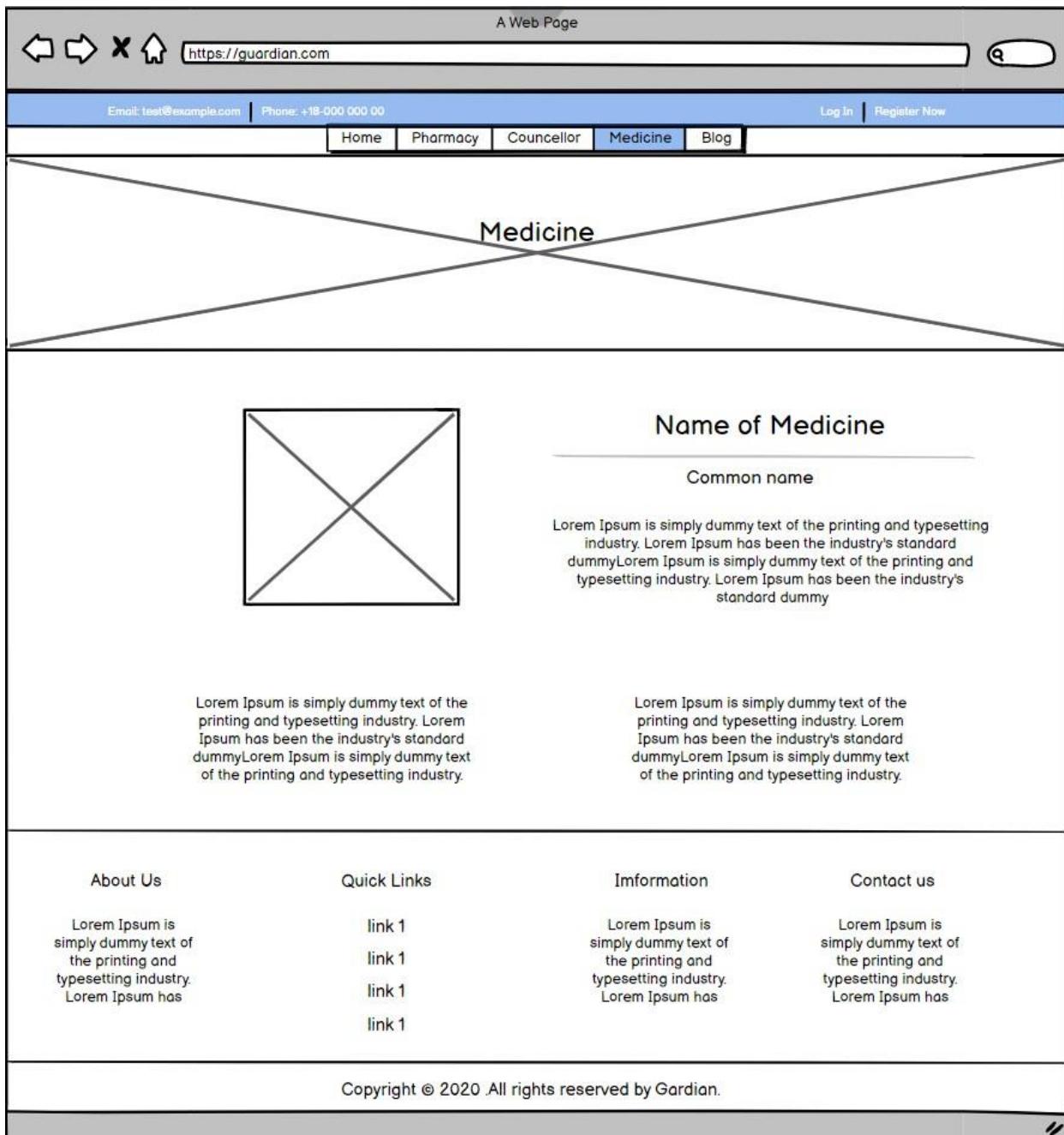


Figure 33: Screenshot of medicine detail page wireframe.

The image above is the wireframe for the medicine detail page. It was designed using Balsamiq.

### 3.7.2.2.8. Wire frame for Blog Page

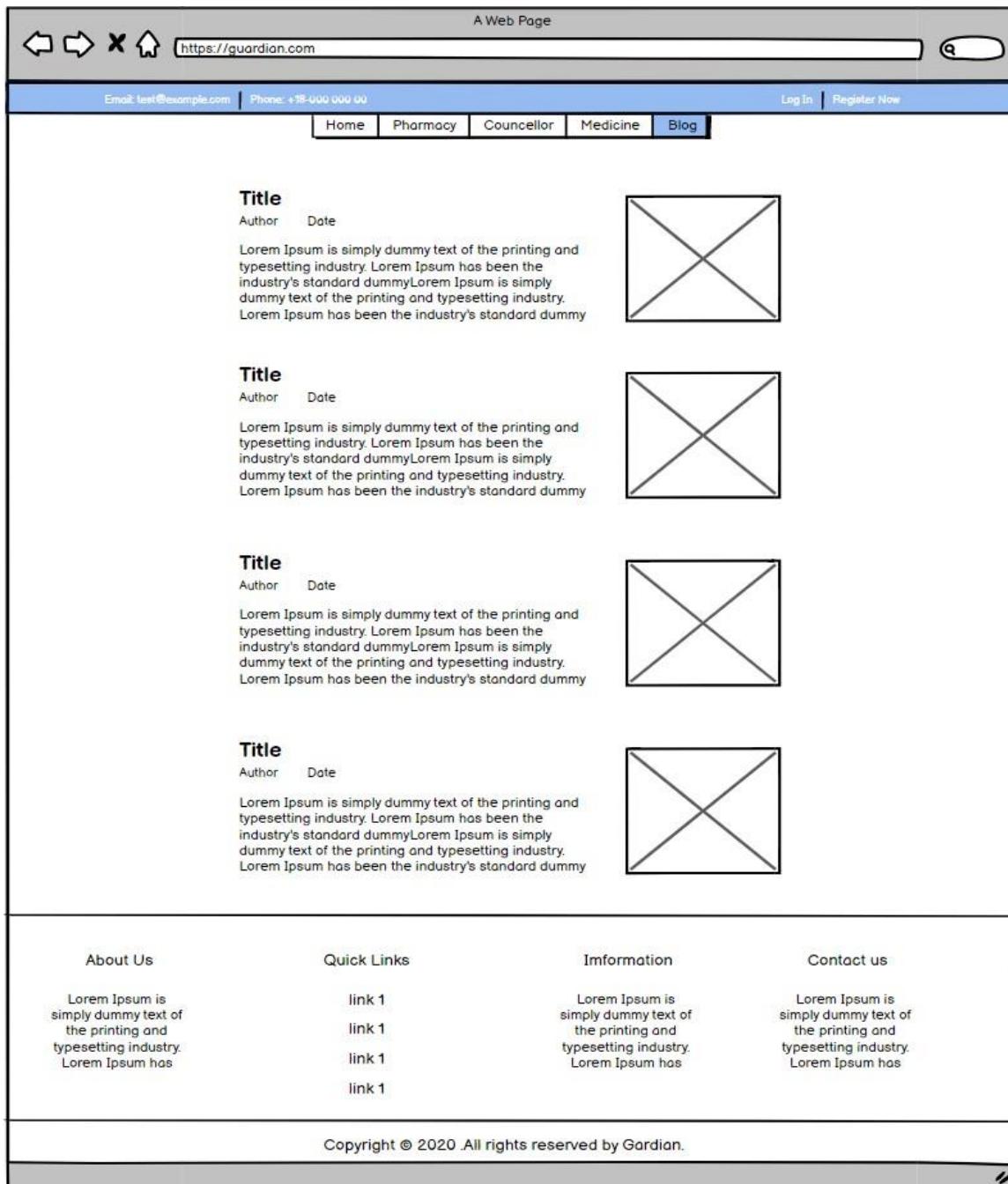
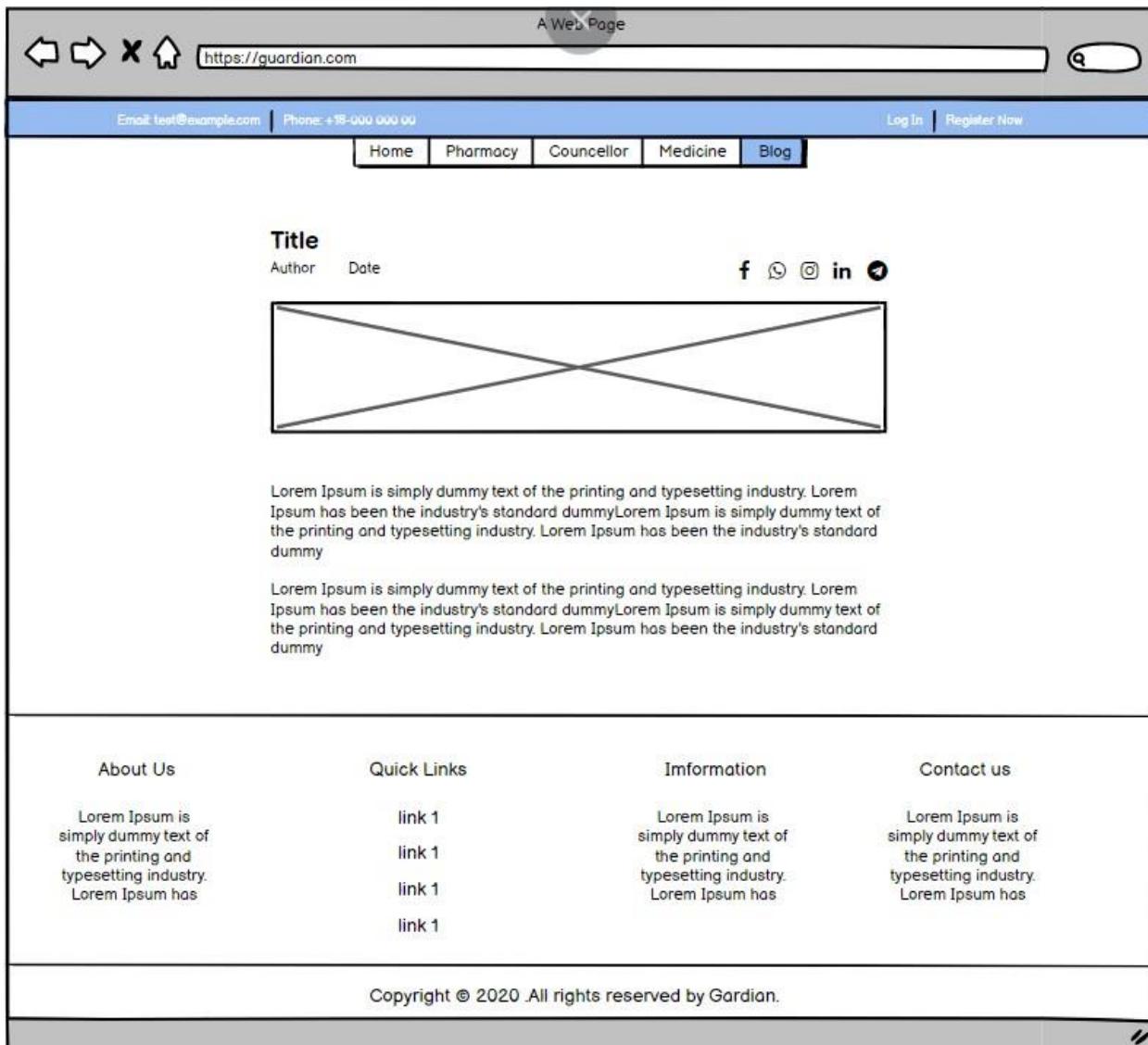


Figure 34: Screenshot of blog page wireframe

The image above is the wireframe for the list of blog page. It was designed using online web application called Balsamiq.

### 3.7.2.2.9. Wire frame for Read Blog Page



*Figure 35: Screenshot of read blog page wireframe*

The image above is the wireframe for the blog page. It was designed with the help of web app- Balsamiq.

### 3.7.2.3. Logo of the web application

The name for the application was decided after its main objective. As this application is for people's mental health and physical health, the logo had to be designed in a way that it had a green colour with a symbol of leaf as nature.

The logo for web application is shown below with its name:



*Figure 36: Official Logo of Application.*

### 3.7.3. Sprint 2

#### 3.7.3.1. Overall use case diagram

The image is attached in appendix.

#### 3.7.3.2. Design Process

The design phase of software development deals with transforming the customer requirements as described in the SRS documents into a form implementable using a programming language.

The software design process can be divided into the following three levels of phases of design:

- Interface Design
- Architectural Design
- Detailed Design

### 3.7.3.2.1. Register User

#### 3.7.3.2.1.1. High Level Use Case Diagram

Table 16: High level Use Case Diagram for user registration.

<b>Use case Name</b>	Register User
<b>Actor(s)</b>	User, Counsellor, Pharmacist
<b>Description</b>	Here, a user can either register as a normal user or as a pharmacist or a counsellor. A user will be registered through their email, and they need to verify their email address in order to proceed. If a user is pharmacist or a counsellor, they need to register their individual pharmacy and counsellor office.

#### 3.7.3.2.1.2. Use case Diagram

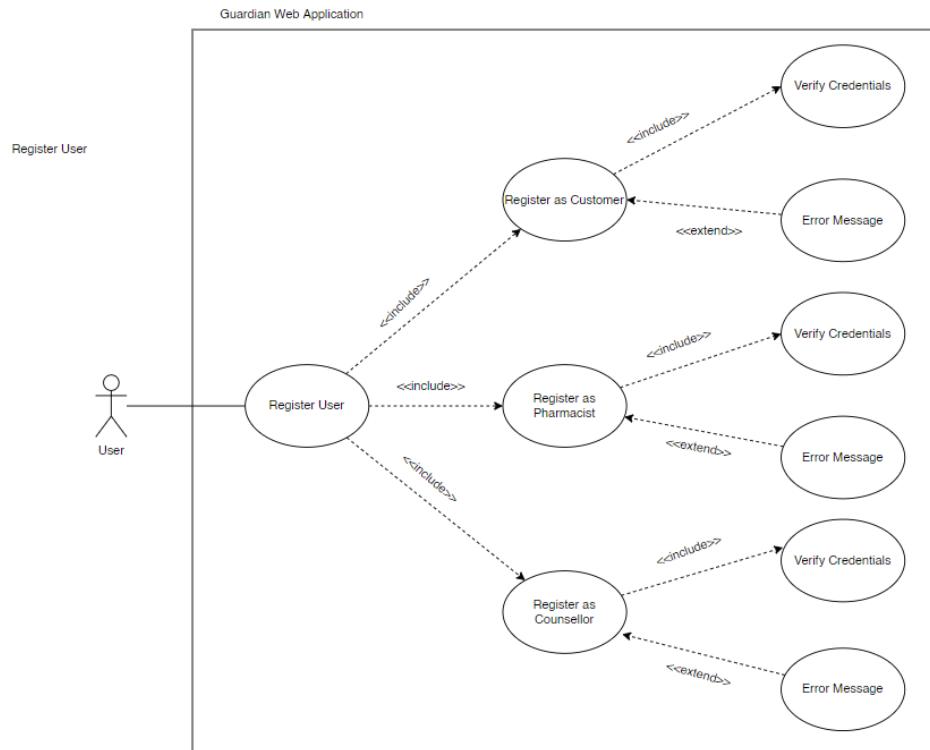


Figure 37: Use case diagram for user registration.

### 3.7.3.2.1.3. Collaboration Diagram

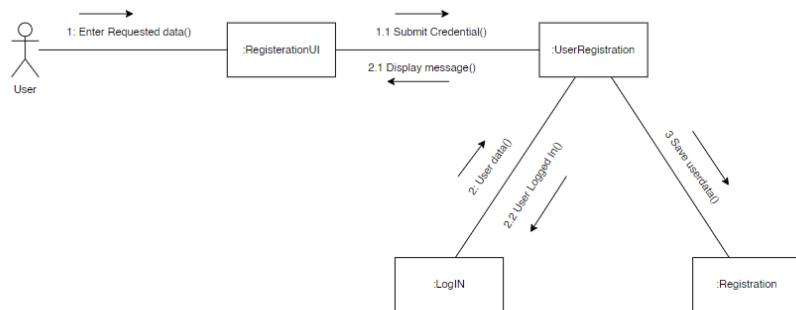


Figure 38: Collaboration diagram for user registration.

### 3.7.3.2.1.4. Sequence Diagram

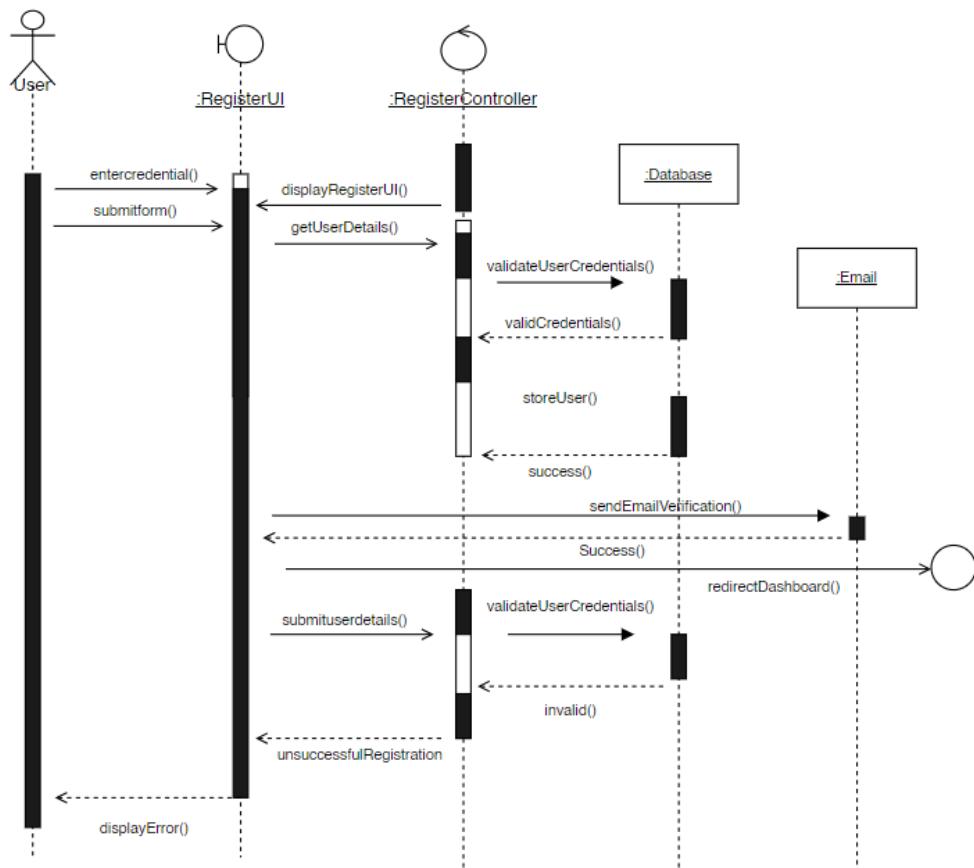


Figure 39: Sequence diagram for register user.

### 3.7.3.2.2. Register Pharmacy

#### 3.7.3.2.2.1. High Level Use Case Diagram

Table 17: High level Use Case Diagram for pharmacy registration.

<b>Use case Name</b>	Register Pharmacy
<b>Actor(s)</b>	Pharmacist
<b>Description</b>	Here, a pharmacist can register their pharmacy. The registered pharmacy must be legit and legally registered as a pharmacy according to the government of Nepal.

#### 3.7.3.2.2.2. Use case Diagram

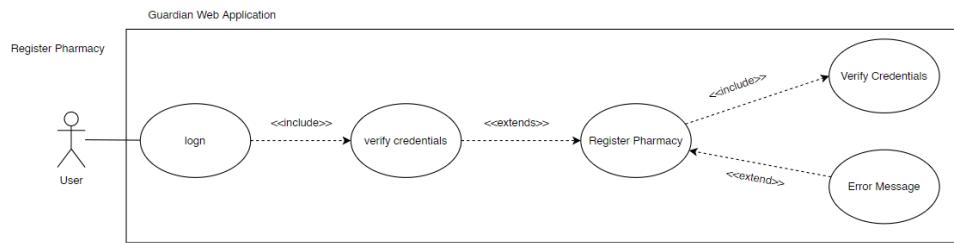


Figure 40: Use case diagram for Pharmacy registration

#### 3.7.3.2.2.3. Collaboration Diagram

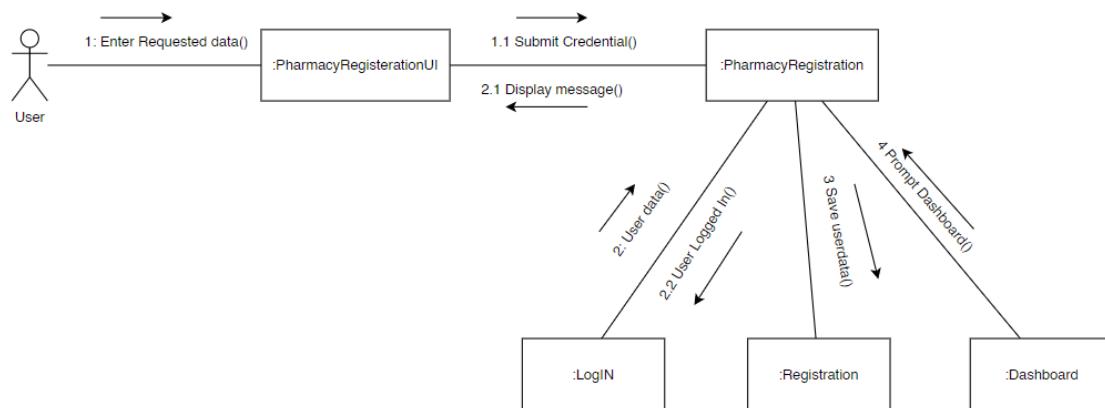


Figure 41: Collaboration diagram for pharmacy registration.

### 3.7.3.2.2.4. Sequence Diagram

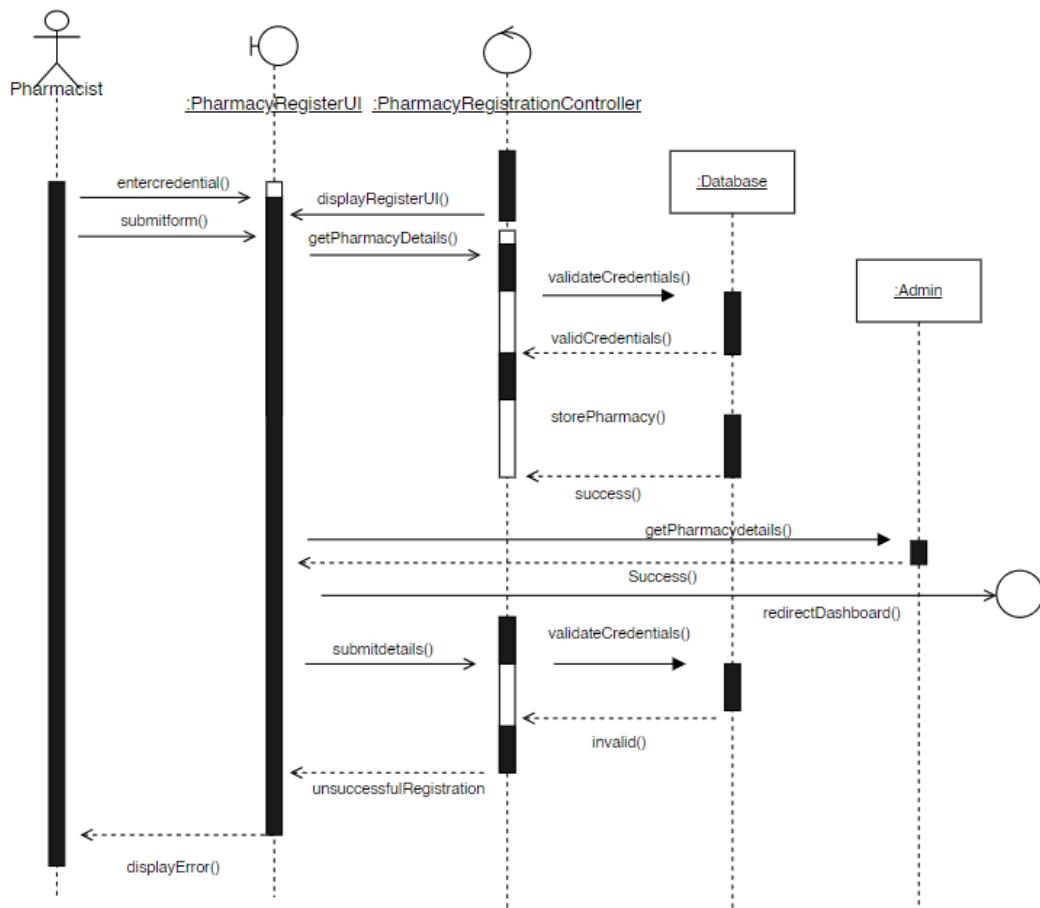


Figure 42: Sequence diagram for register pharmacy.

### 3.7.3.2.3. Register Counsellor

#### 3.7.3.2.3.1. High Level Use Case Diagram

Table 18: High level Use Case Diagram for counsellor registration.

<b>Use case Name</b>	Register Counsellor
<b>Actor(s)</b>	Counsellor
<b>Description</b>	Here, a counsellor can register themselves as a verified counsellor. The registered counsellor must be legit and legally registered as a counsellor according to the government of Nepal.

### 3.7.3.2.3.2. Use case Diagram

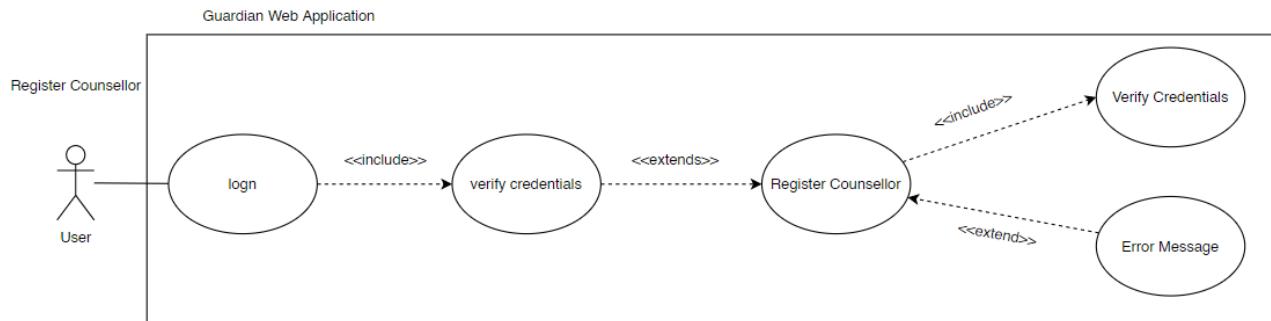


Figure 43: Use case diagram for counsellor registration

### 3.7.3.2.3.3. Collaboration Diagram

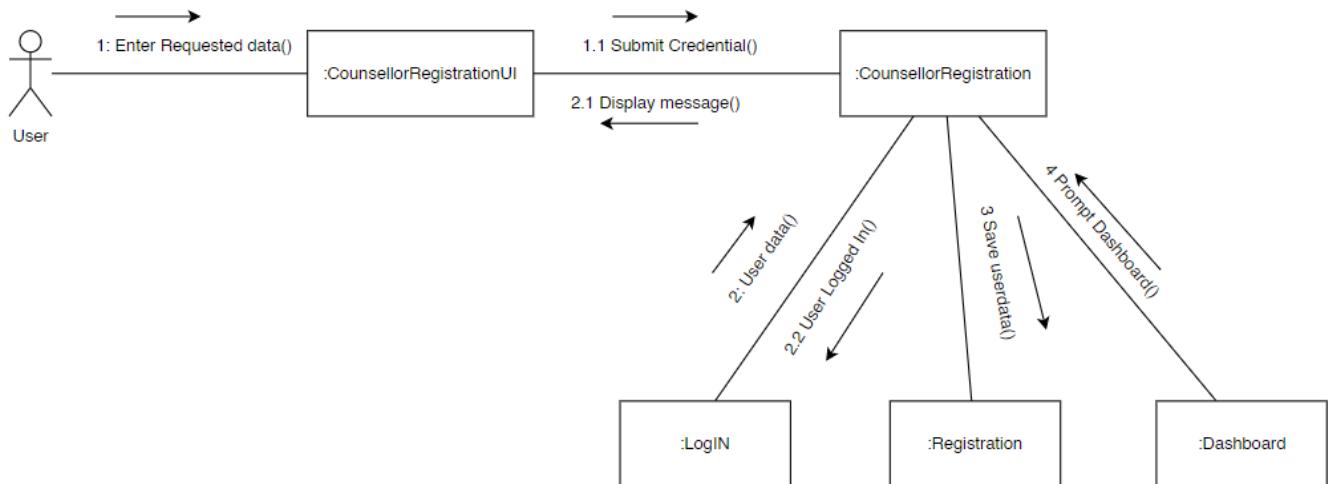


Figure 44: Collaboration diagram for counsellor registration.

### 3.7.3.2.3.4. Sequence Diagram

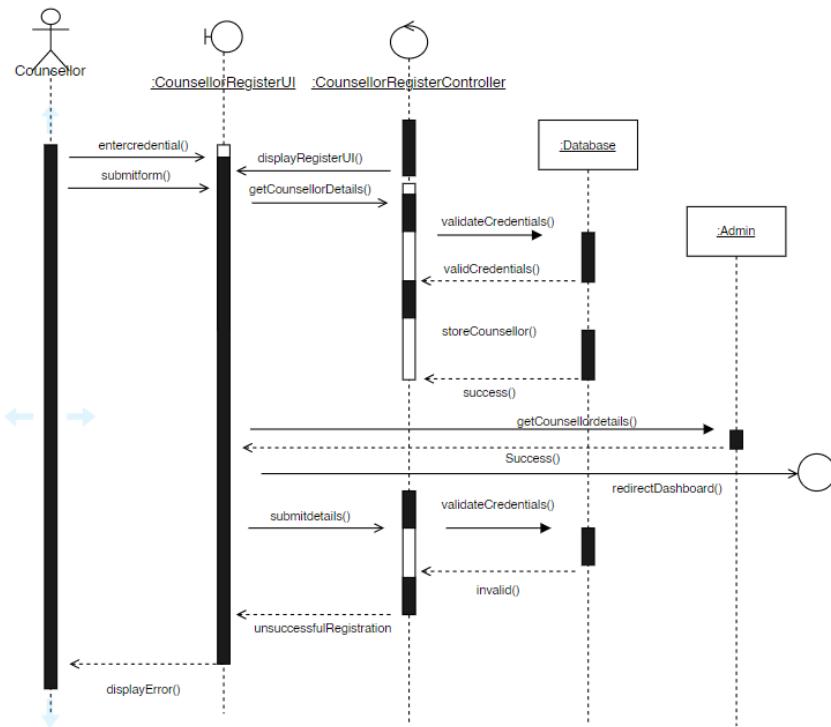


Figure 45: Sequence diagram for register counsellor.

### 3.7.3.2.4. Log In as User

#### 3.7.3.2.4.1. High Level Use Case Diagram

Table 19: High level Use Case Diagram for user login.

<b>Use case Name</b>	Login as User
<b>Actor(s)</b>	User
<b>Description</b>	The login procedure is simple in which the user needs to enter their email and password. If the entered credentials are correct, the user is redirected to the user dashboard page else an error message is displayed.

### 3.7.3.2.4.2. Use case Diagram

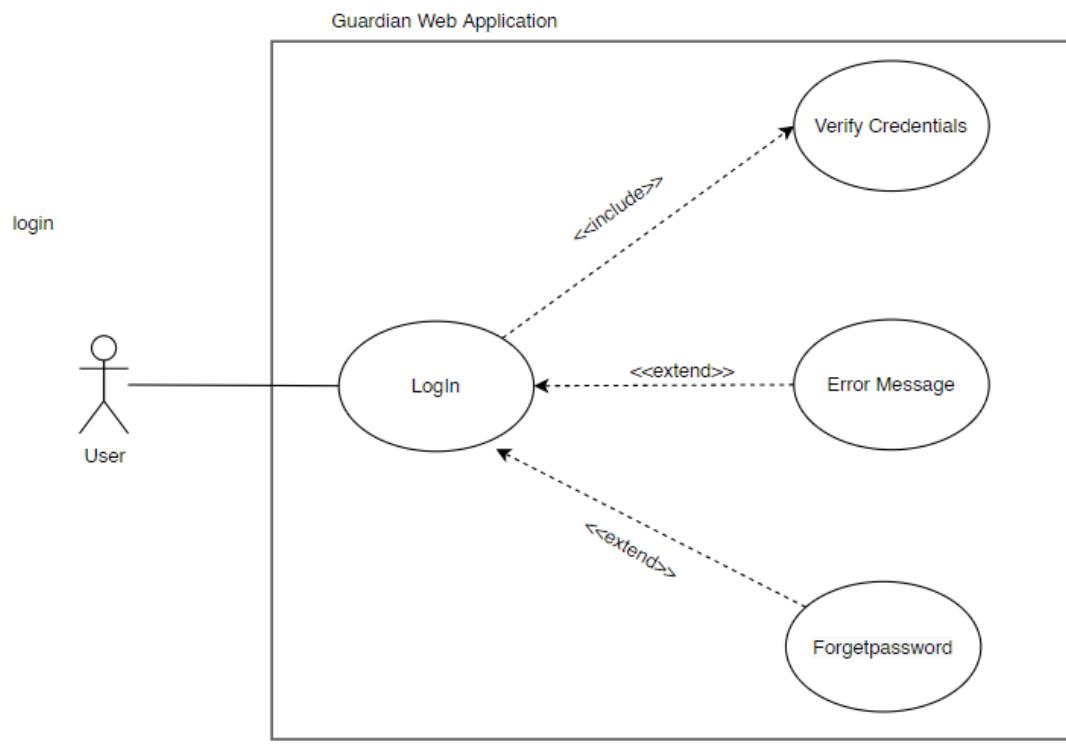


Figure 46: Use case diagram for user login.

### 3.7.3.2.4.3. Collaboration Diagram

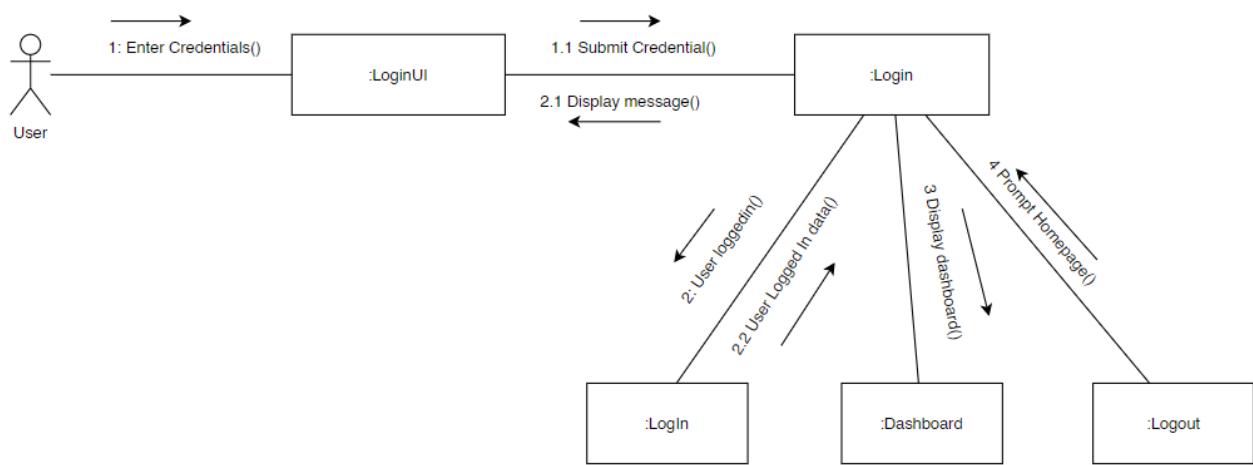


Figure 47: Collaboration diagram for user login.

### 3.7.3.2.4.4. Sequence Diagram

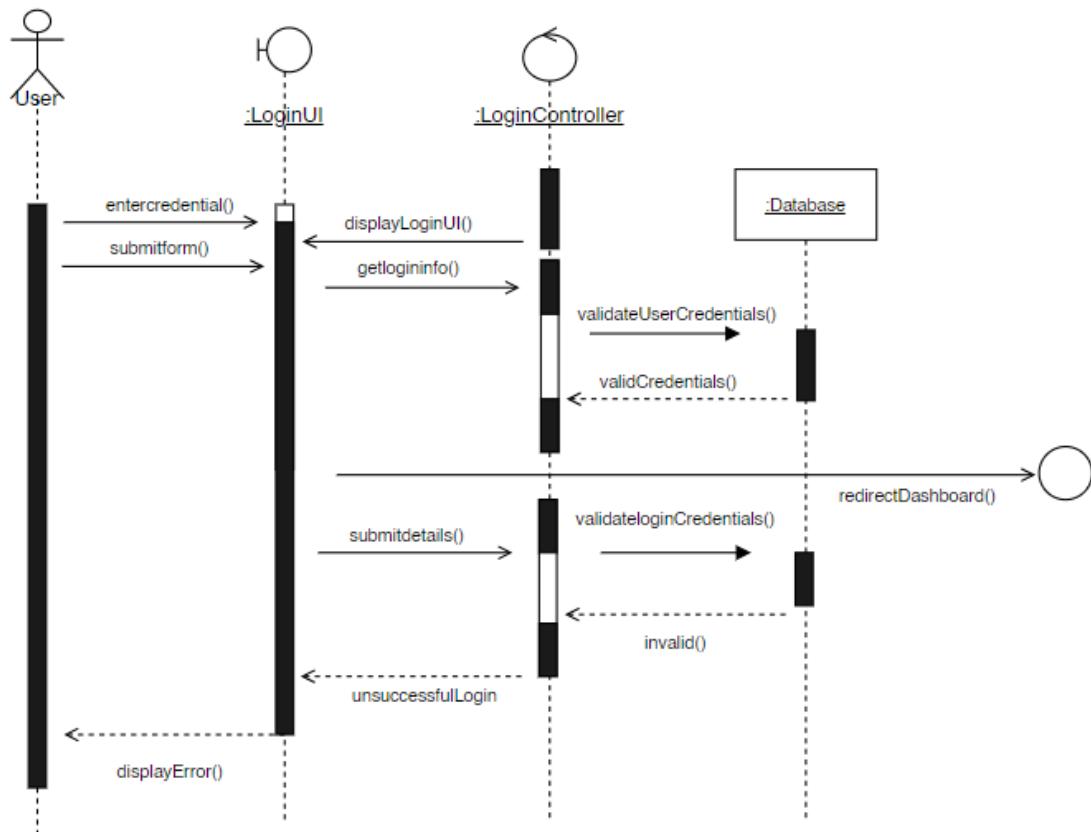


Figure 48: Sequence diagram for user login.

### 3.7.3.2.5. Log In as Admin

#### 3.7.3.2.5.1. High Level Use Case Diagram

Table 20: High level Use Case Diagram for admin login.

<b>Use case Name</b>	Login as Admin
<b>Actor(s)</b>	Admin
<b>Description</b>	The admin needs to enter their email and password. If the entered credentials are correct, the admin is redirected to the admin dashboard page else an error message is displayed.

### 3.7.3.2.5.2. Use case Diagram

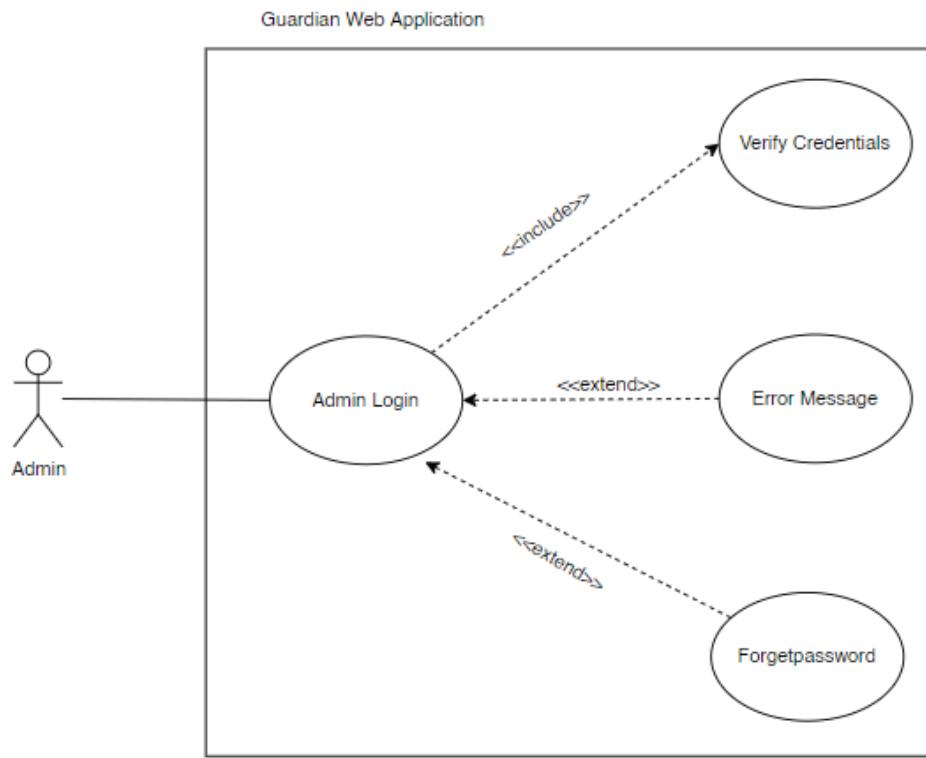


Figure 49: Use case diagram for admin login.

### 3.7.3.2.5.3. Sequence Diagram

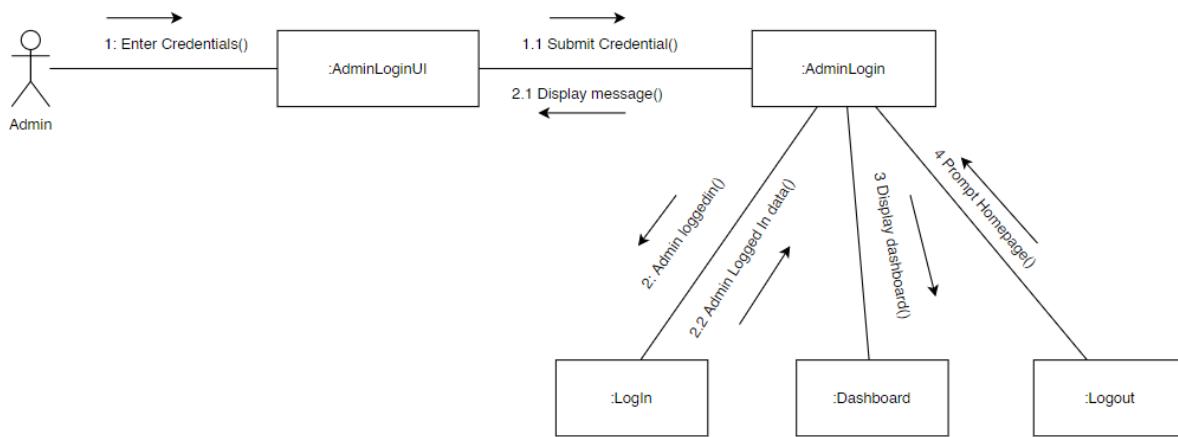


Figure 50: Collaboration diagram for admin login.

### 3.7.3.2.5.4. Sequence Diagram

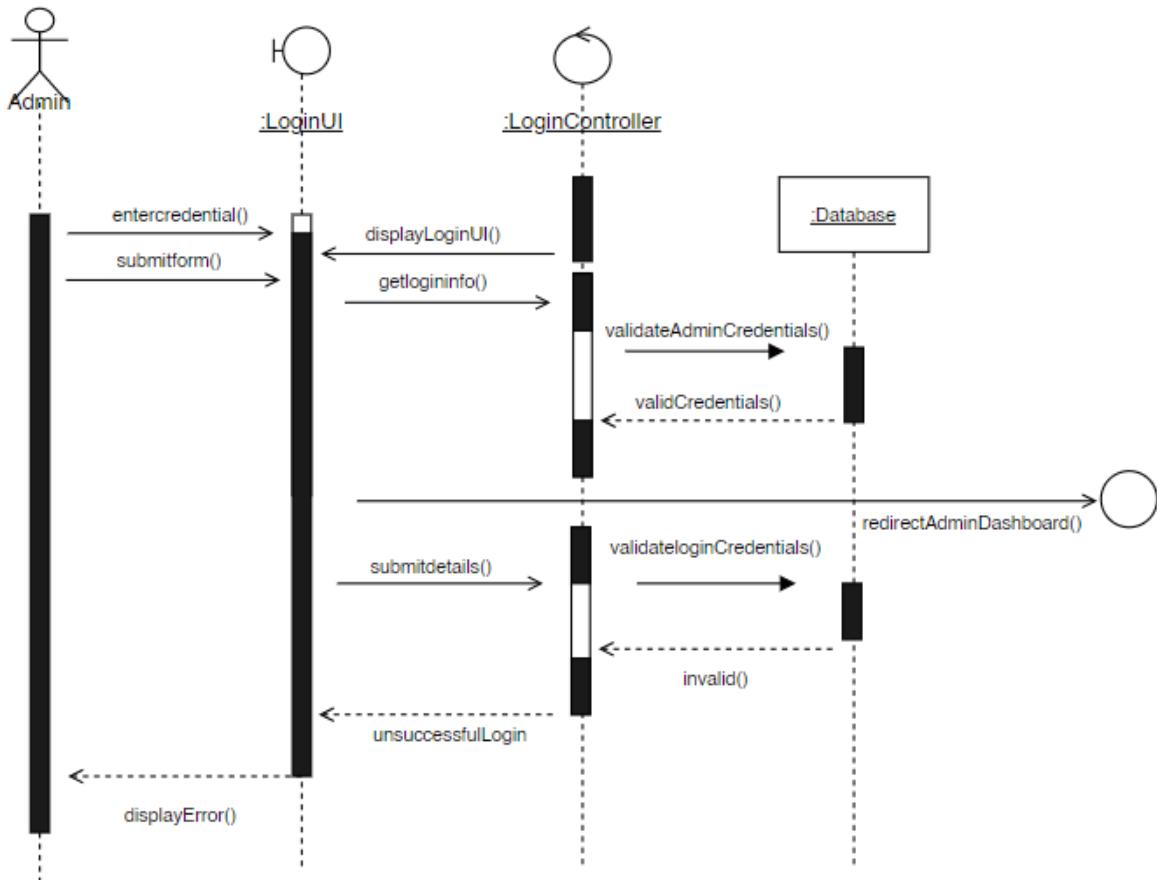


Figure 51: Sequence diagram for admin login.

### 3.7.3.2.6. Add Medicine

#### 3.7.3.2.6.1. High Level Use Case Diagram

Table 21: High level Use Case Diagram for add medicine.

<b>Use case Name</b>	Add Medicine
<b>Actor(s)</b>	Pharmacist
<b>Description</b>	The pharmacist can create and add new Medicines. The pharmacist enters all the details of the product and users can see it in respective pharmacy.

### 3.7.3.2.6.2. Use case Diagram

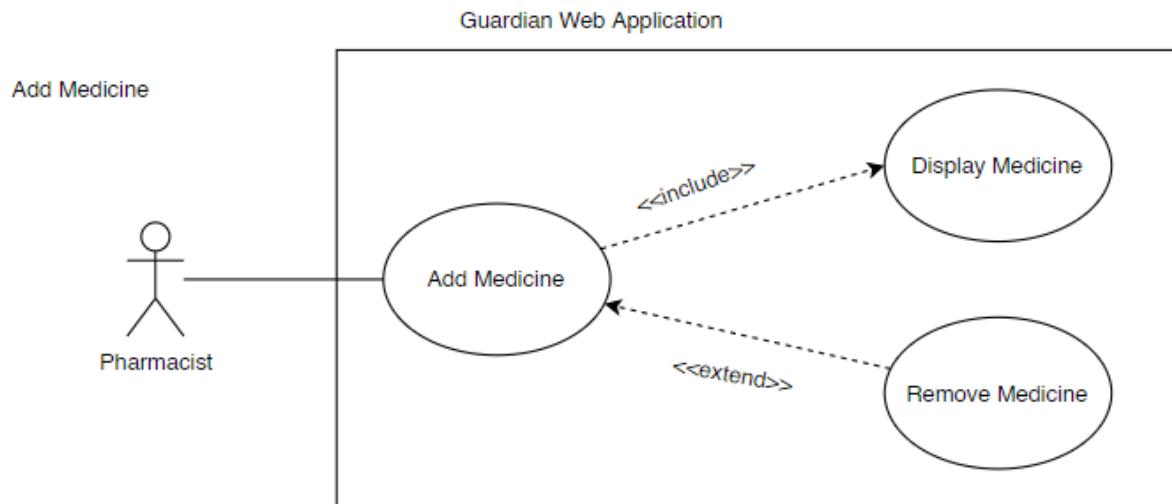


Figure 52: Use case diagram for add medicine.

### 3.7.3.2.6.3. Collaboration Diagram

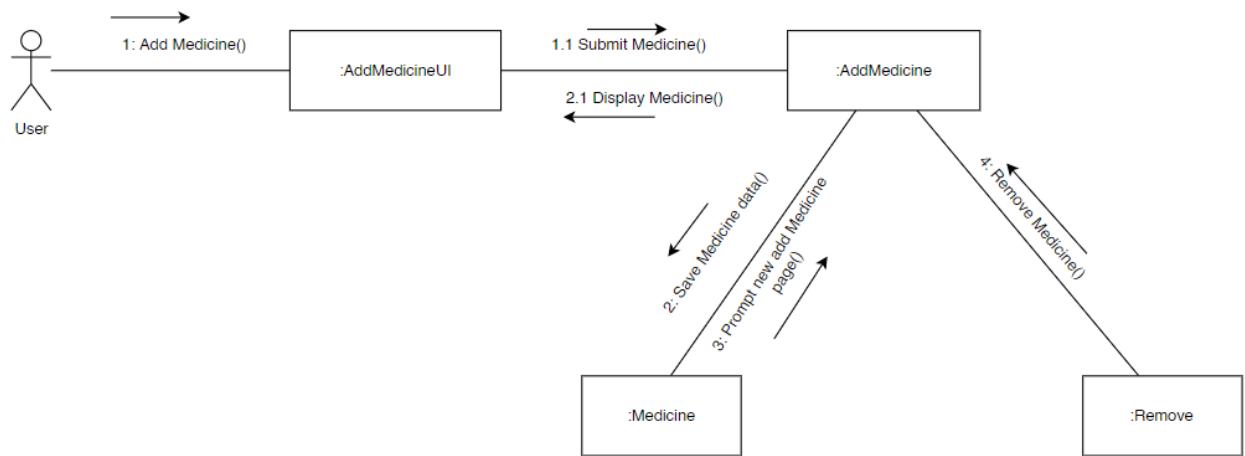


Figure 53: Collaboration diagram for add medicine.

### 3.7.3.2.6.4. Sequence Diagram

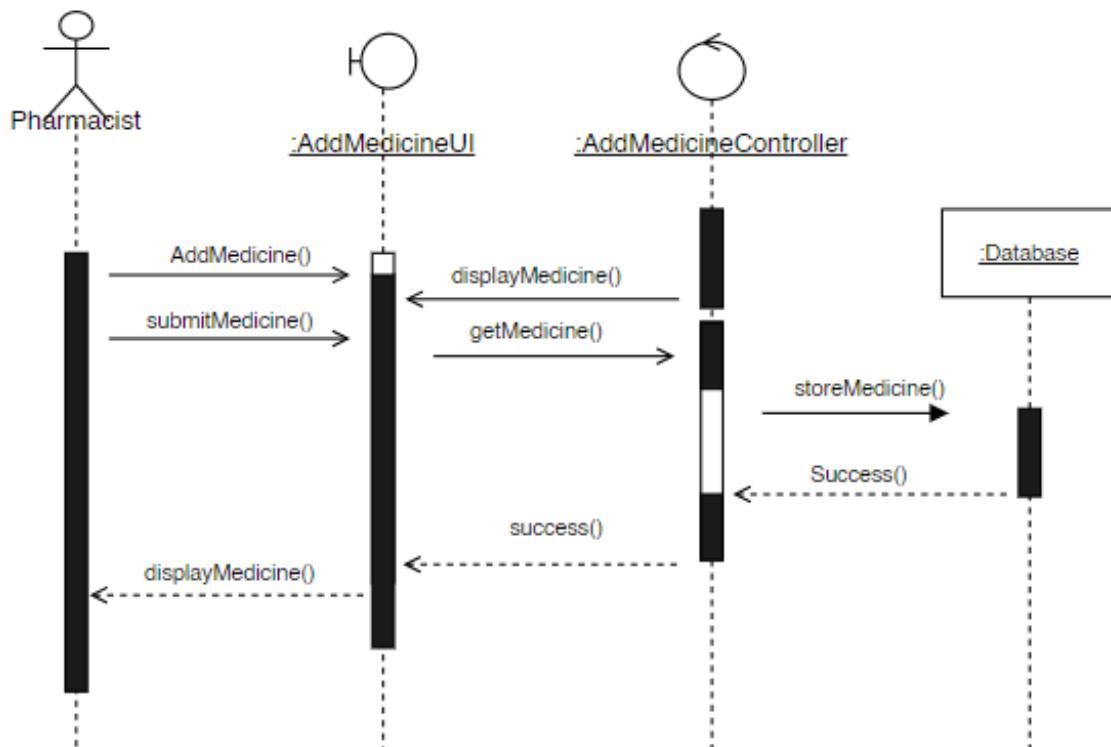


Figure 54: Sequence diagram for add medicine.

### 3.7.3.2.7. Add Blog

#### 3.7.3.2.7.1. High Level Use Case Diagram

Table 22: High level Use Case Diagram for add Blog.

<b>Use case Name</b>	Add Blog
<b>Actor(s)</b>	Counsellor
<b>Description</b>	The counsellor can create and add new Blog. The counsellor enters all the contents for the blog and users can read.

### 3.7.3.2.7.2. Use case Diagram

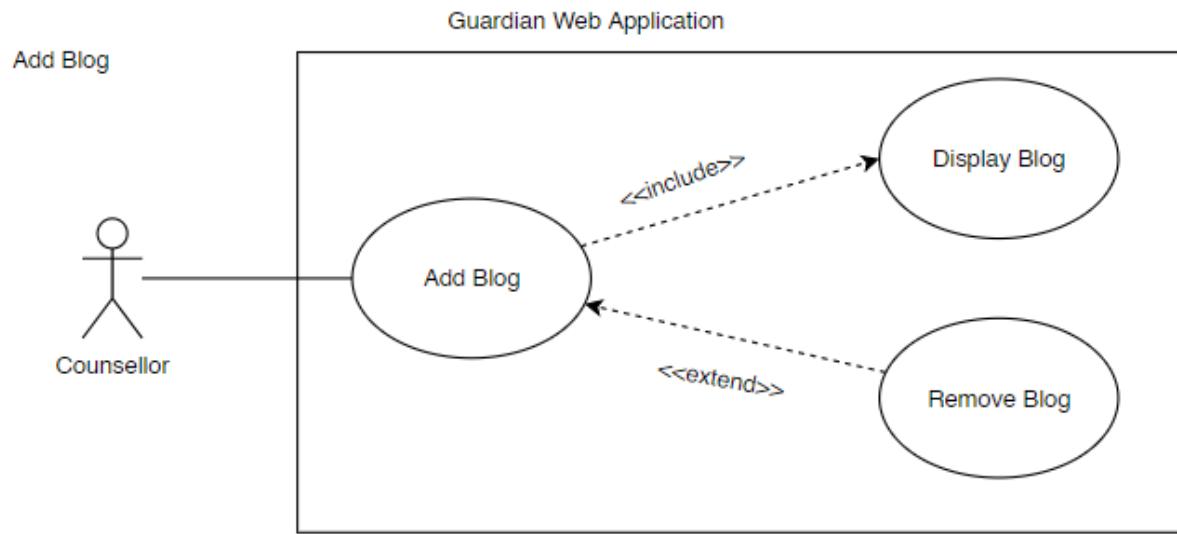


Figure 55: Use case diagram for add blog.

### 3.7.3.2.7.3. Collaboration Diagram

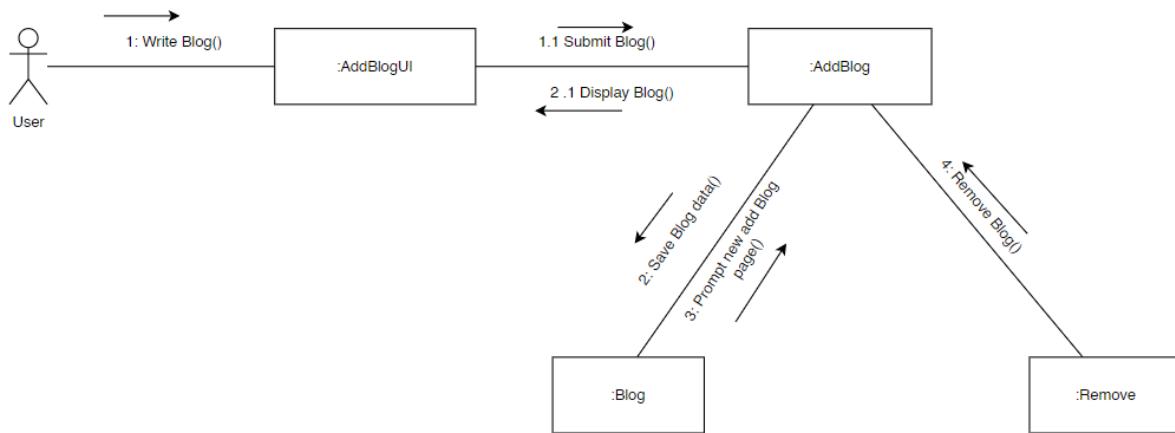


Figure 56: Collaboration diagram for add blog.

### 3.7.3.2.7.4. Sequence Diagram

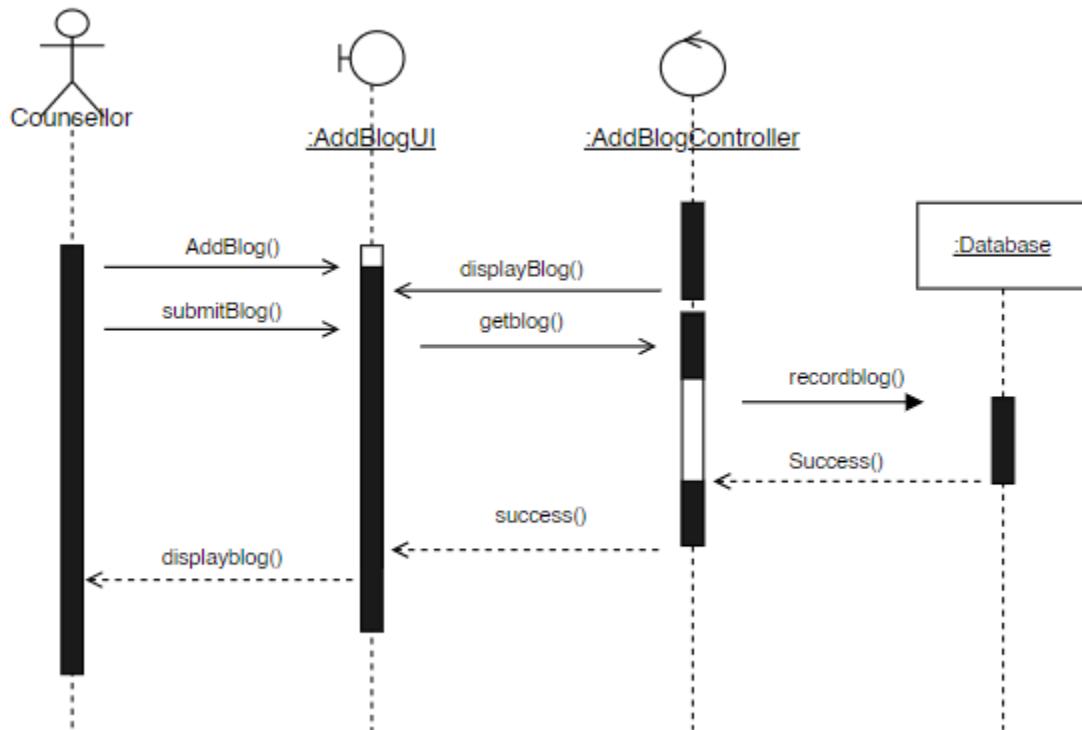


Figure 57: Sequence diagram for add blog.

### 3.7.3.2.8. Add to Cart

#### 3.7.3.2.8.1. High Level Use Case Diagram

Table 23: High level Use Case Diagram for add to cart.

<b>Use case Name</b>	Add to Cart
<b>Actor(s)</b>	User
<b>Description</b>	A user can add their desired medicines to a cart and purchase them. They can also view the medicine description along with its reviews. In order to buy a medicine a user must be registered else; they are redirected to the registration page

### 3.7.3.2.8.2. Use case Diagram

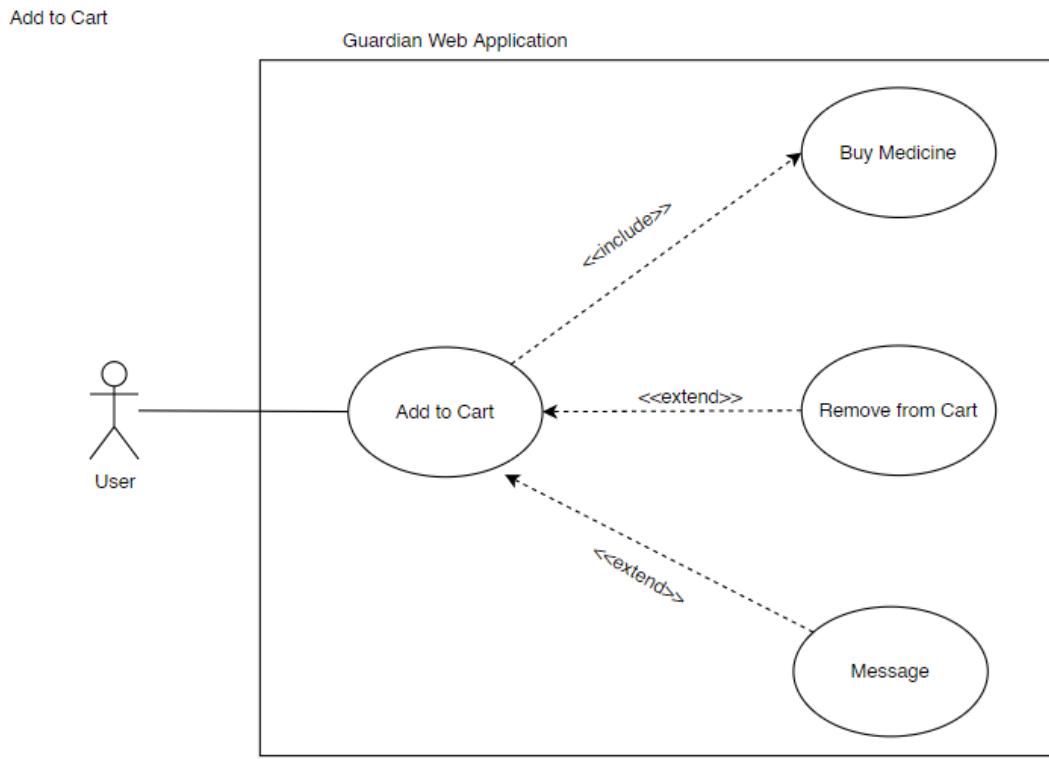


Figure 58: Use case diagram for add to cart.

### 3.7.3.2.8.3. Collaboration Diagram

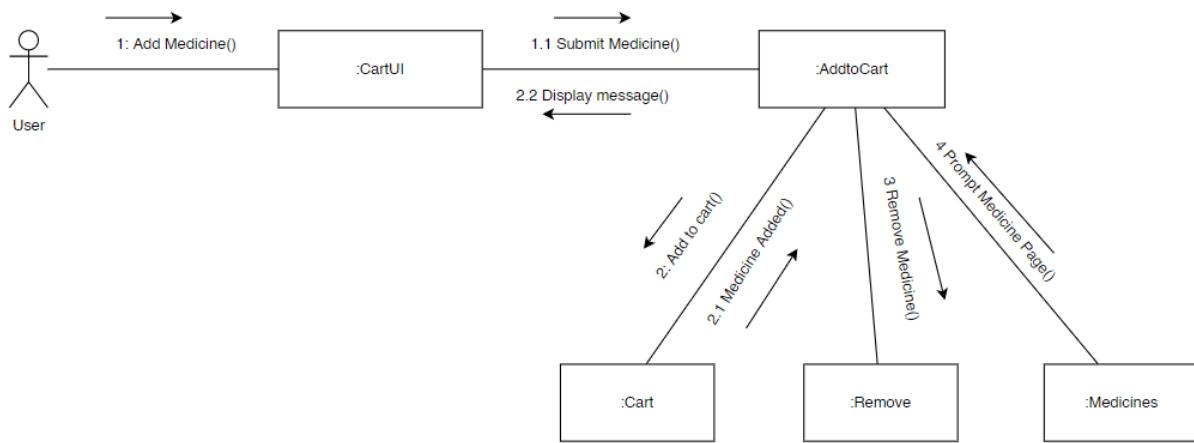


Figure 59: Collaboration diagram for add to cart.

### 3.7.3.2.8.4. Sequence Diagram

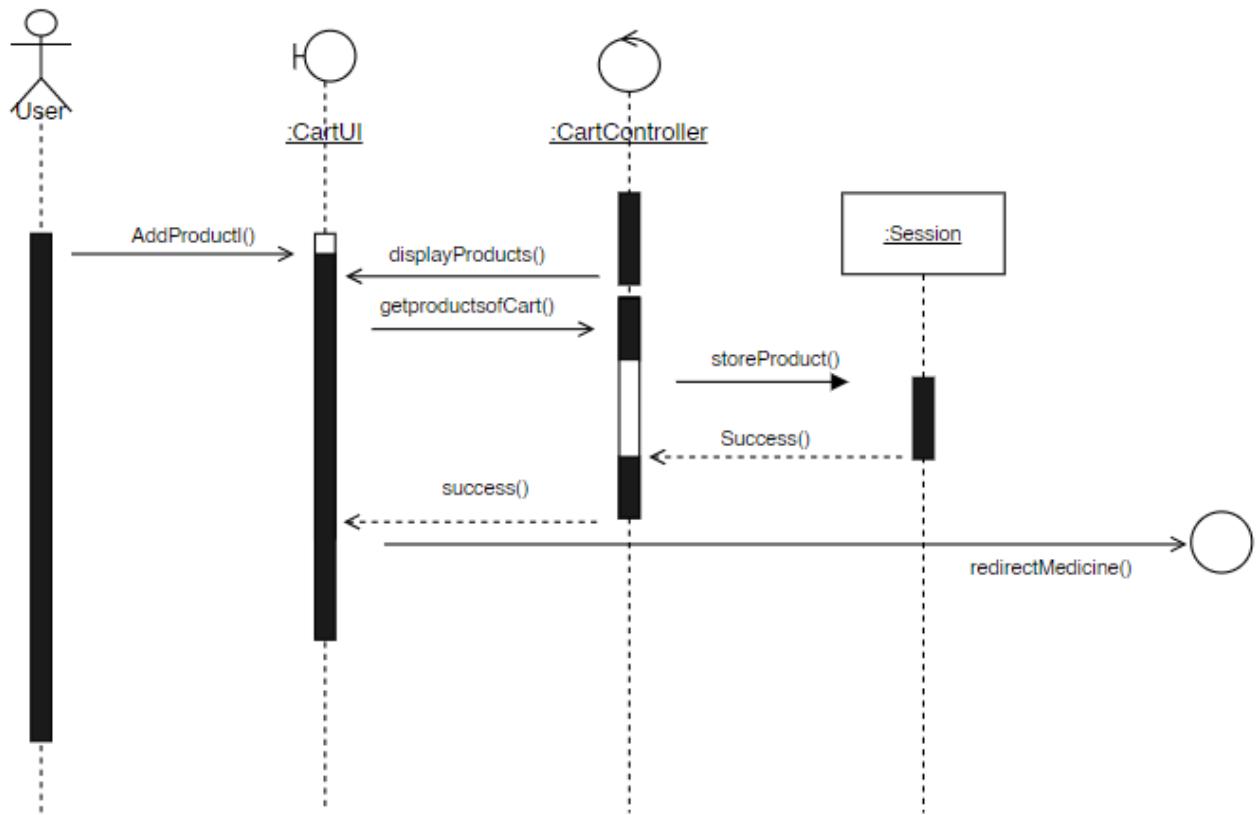


Figure 60: Sequence diagram for add to cart.

### 3.7.3.2.9. Make Payment

#### 3.7.3.2.9.1. High Level Use Case Diagram

Table 24: High level Use Case Diagram for make payment.

<b>Use case Name</b>	Make Payment
<b>Actor(s)</b>	User
<b>Description</b>	A user needs to be registered to make a payment. After selecting their desired medicines, they can either pay online or can pay after they receive their order.

### 3.7.3.2.9.2. Use case Diagram

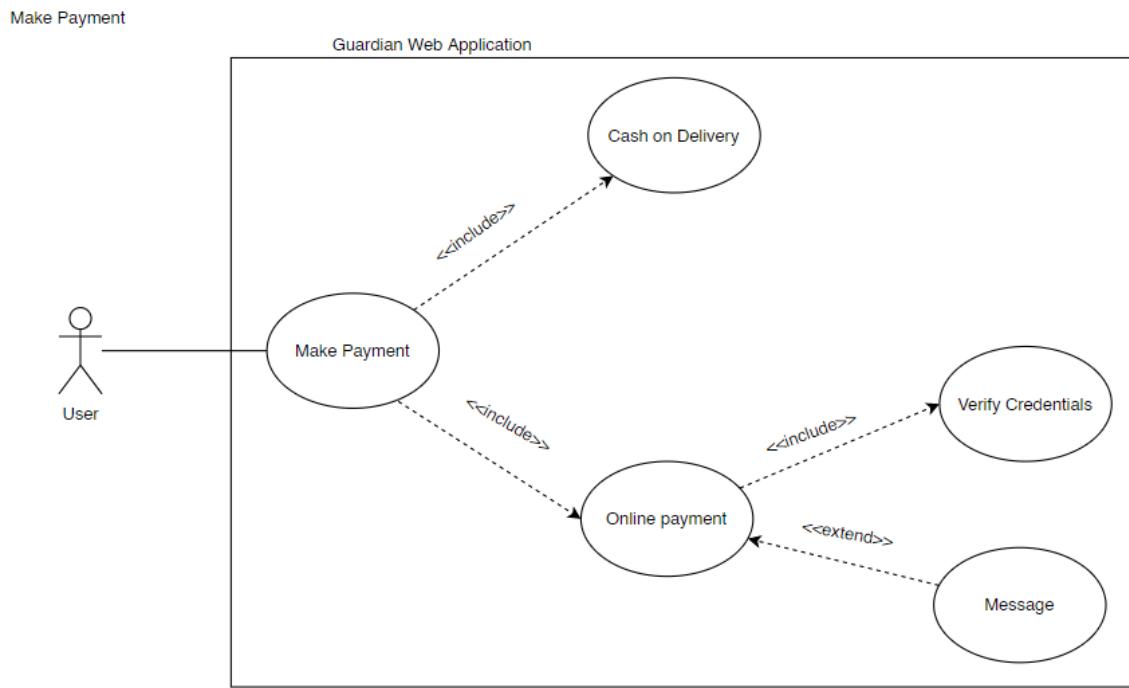


Figure 61: Use case diagram for make payment.

### 3.7.3.2.9.3. Collaboration Diagram

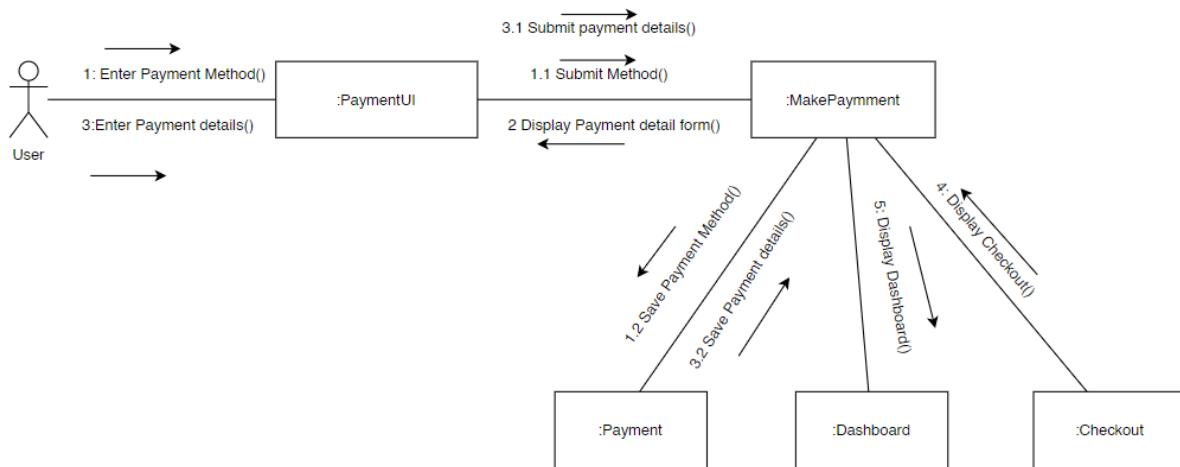


Figure 62: Collaboration diagram for making payment.

### 3.7.3.2.9.4. Sequence Diagram

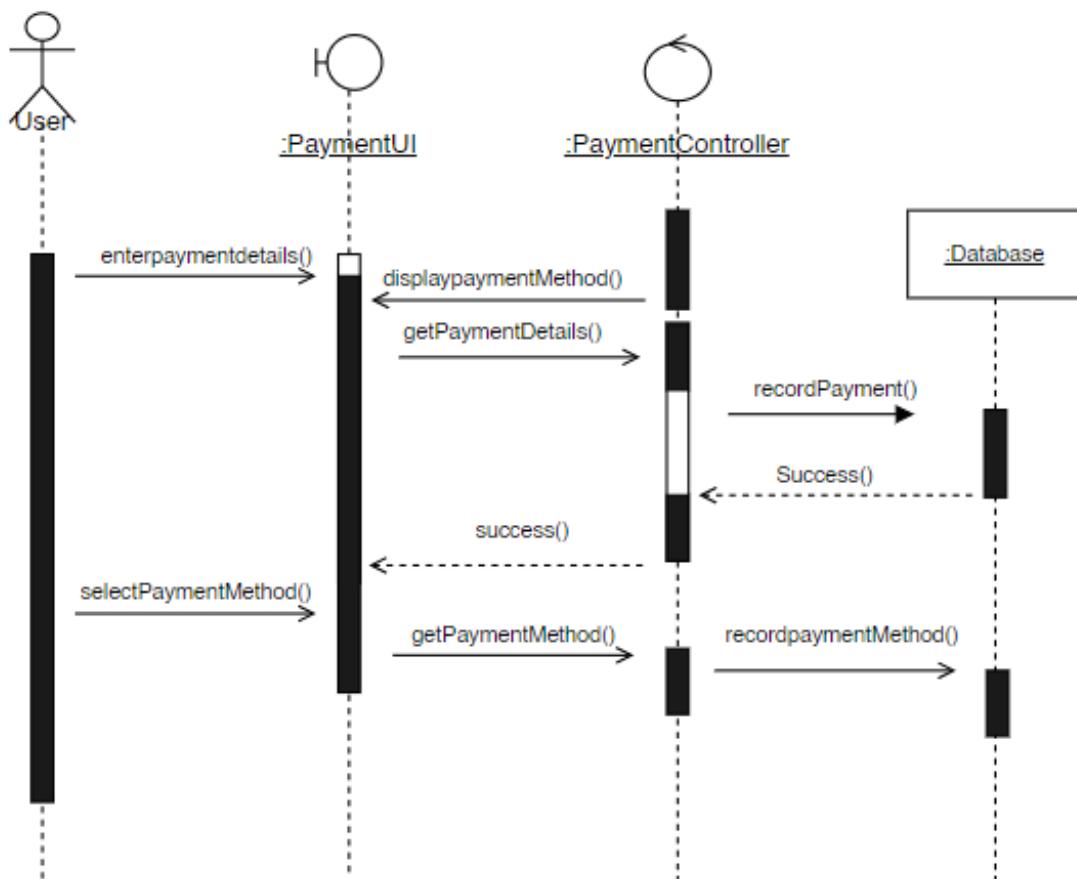


Figure 63: Sequence diagram for make payment.

### 3.7.3.2.10. Add Review

#### 3.7.3.2.10.1. High Level Use Case Diagram

Table 25: High level Use Case Diagram for add review.

<b>Use case Name</b>	Add Review
<b>Actor(s)</b>	user
<b>Description</b>	A user can give a review about the medicine they have bought and meeting session they have attend. Other users can view these reviews.

### 3.7.3.2.10.2. Use case Diagram

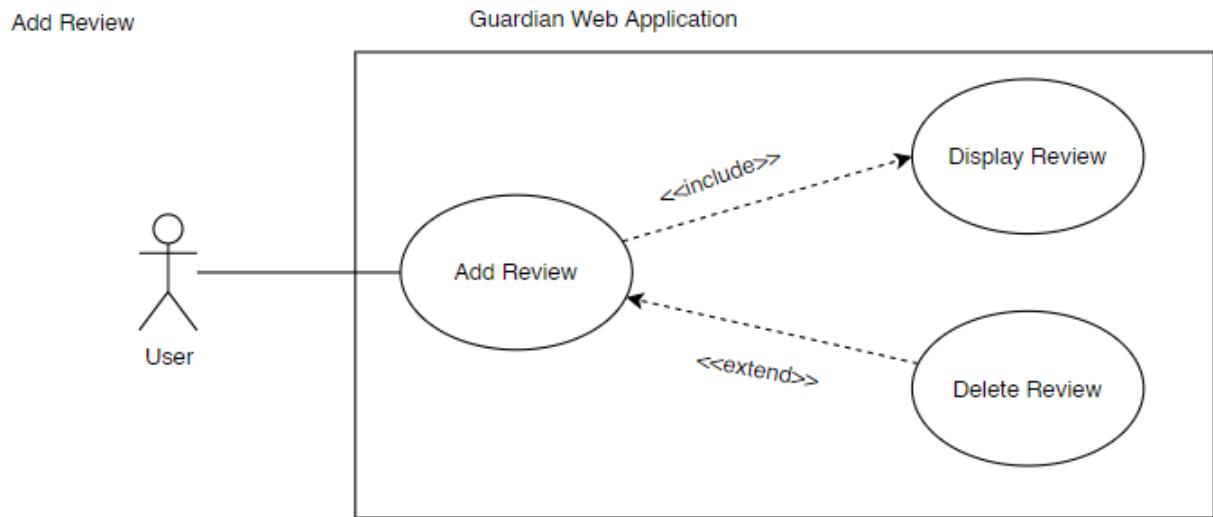


Figure 64: Use case diagram for add review.

### 3.7.3.2.10.3. Collaboration Diagram

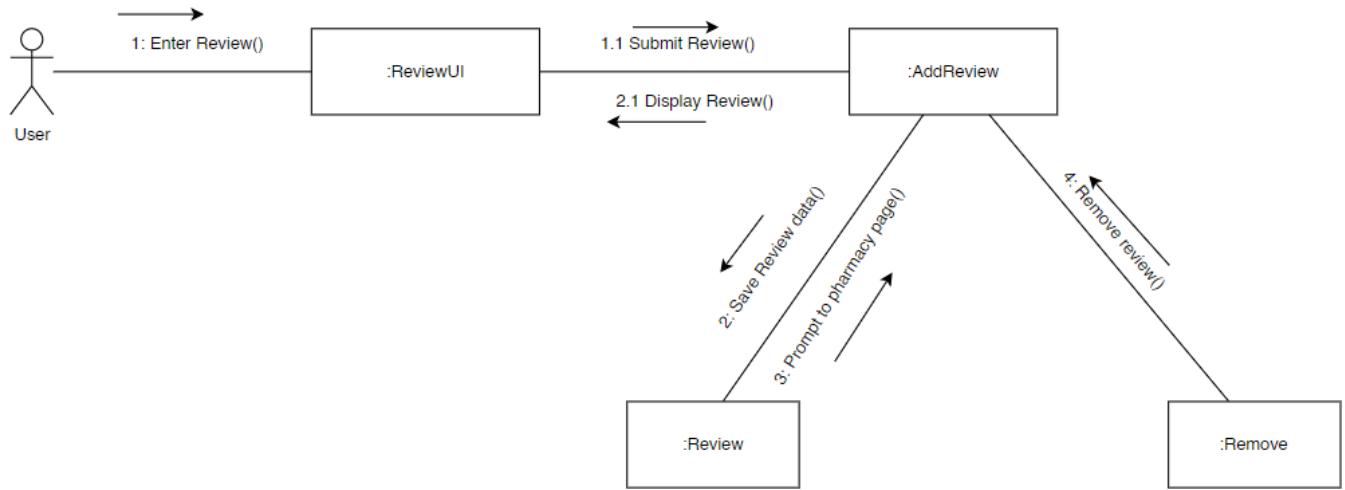


Figure 65: Collaboration diagram for add review.

### 3.7.3.2.10.4. Sequence Diagram

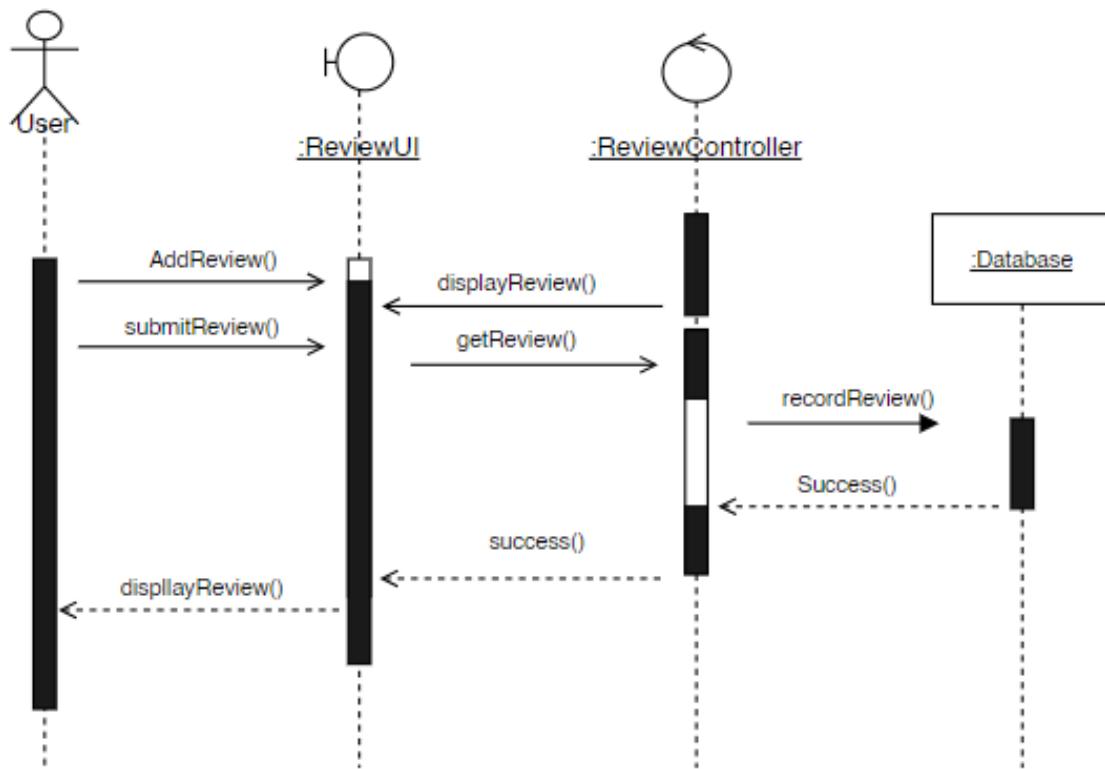


Figure 66: Sequence diagram for add review.

### 3.7.3.2.11. Book Session/Meeting

#### 3.7.3.2.11.1. High Level Use Case Diagram

Table 26: High level Use Case Diagram for book session.

<b>Use case Name</b>	Book Meeting
<b>Actor(s)</b>	User, Counsellor, Admin
<b>Description</b>	A user can book an hour of meeting session with counsellor. The user, counsellor and admin get calendar notification after the booking is success.

### 3.7.3.2.11.2. Use case Diagram

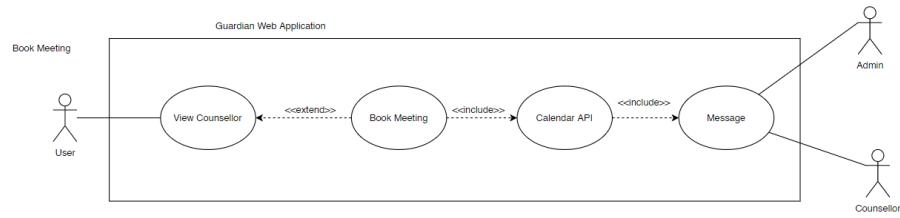


Figure 67: Use case diagram for book session/meeting.

### 3.7.3.2.11.3. Collaboration Diagram

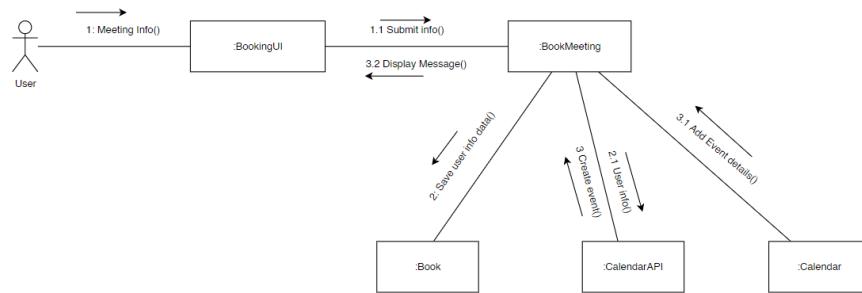


Figure 68: Collaboration diagram for booking meeting/session.

### 3.7.3.2.11.4. Sequence Diagram

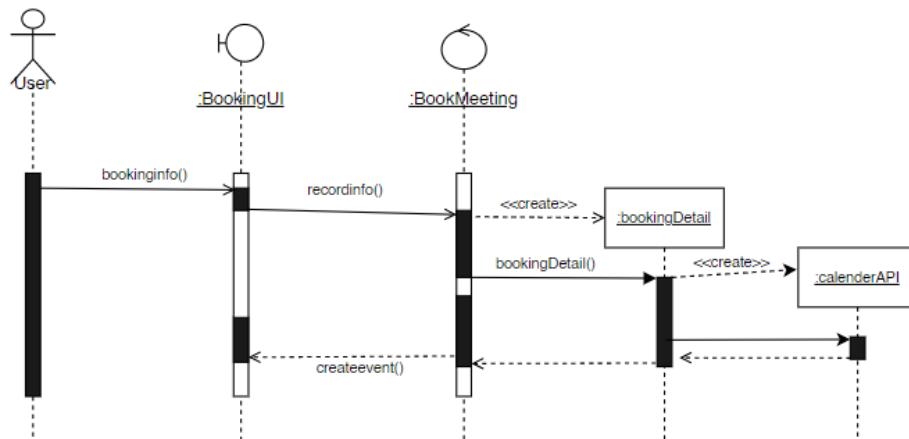


Figure 69: Sequence diagram for booking meeting/session.

### 3.7.3.2.12. Upload Prescription

#### 3.7.3.2.12.1. High Level Use Case Diagram

Table 27: High level Use Case Diagram for upload prescription.

<b>Use case Name</b>	Upload Prescription
<b>Actor(s)</b>	User, Pharmacist, Admin
<b>Description</b>	The users can upload their medicine prescription to the pharmacy or to the admin. The admin uses API to extract text whereas the pharmacist read the prescription and give them the medicines prescribed.

#### 3.7.3.2.12.2. Use case Diagram

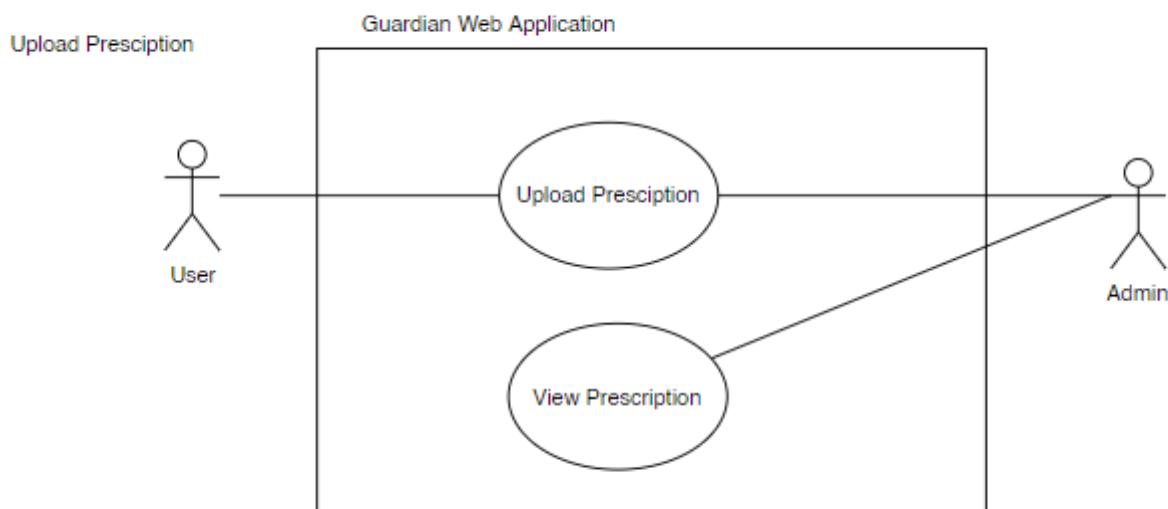


Figure 70: Use case diagram for upload prescription.

### 3.7.3.2.12.3. Collaboration Diagram

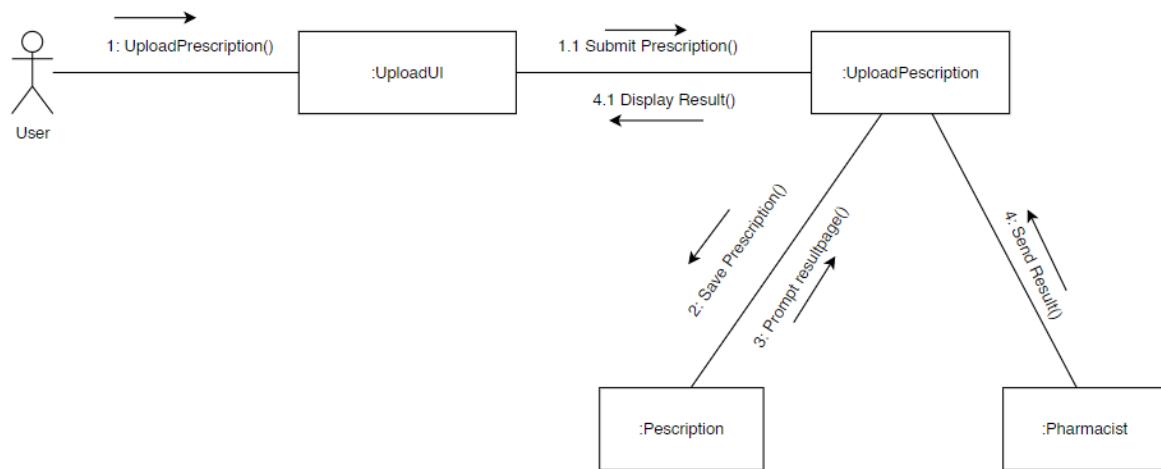


Figure 71: Collaboration diagram for upload prescription.

### 3.7.3.2.12.4. Sequence Diagram

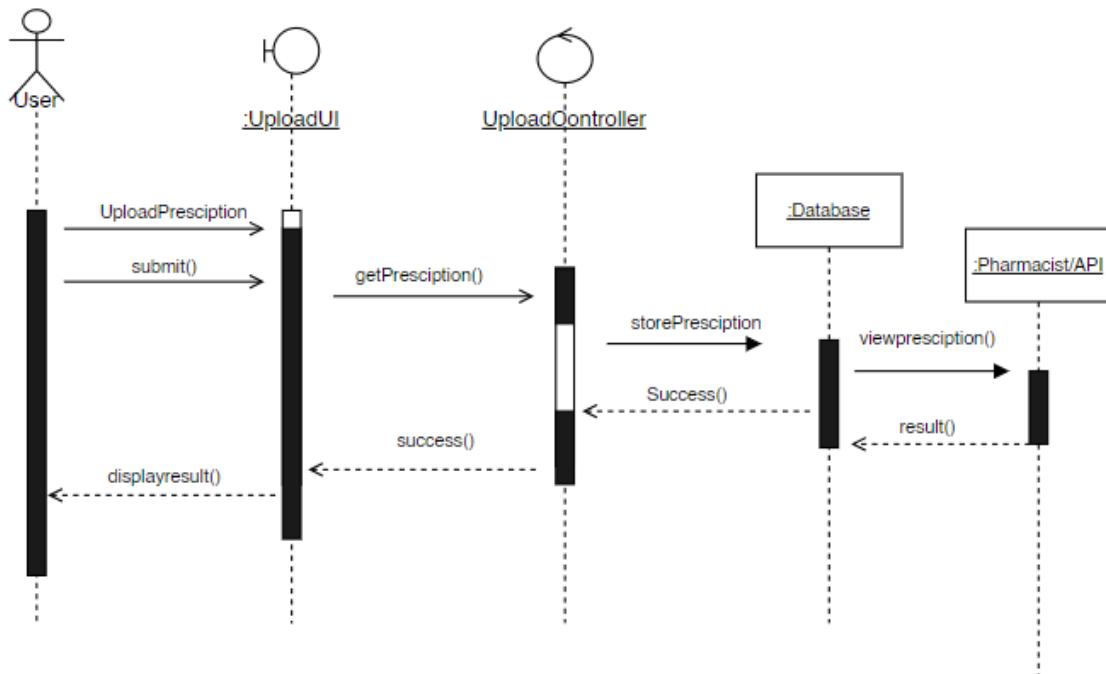


Figure 72: Sequence diagram for upload prescription.

### 3.7.3.2.13. Add Cities

#### 3.7.3.2.13.1. High Level Use Case Diagram

Table 28: High level Use Case Diagram for Add Cities.

<b>Use case Name</b>	Add Cities
<b>Actor(s)</b>	Admin
<b>Description</b>	The admin can create and add new Cities. The admin enters all the details of the city.

#### 3.7.3.2.13.2. Use Case Diagram

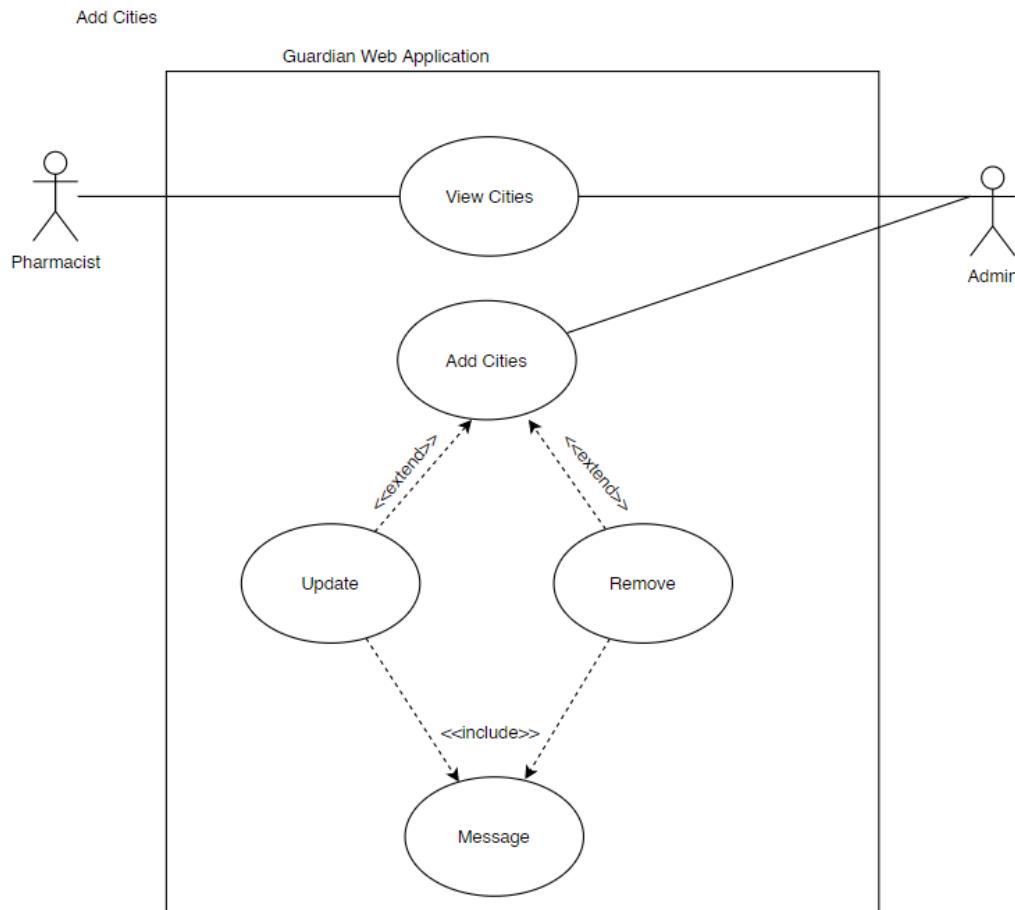


Figure 73: Use case diagram for add cities.

### 3.7.3.2.13.3. Collaboration Diagram

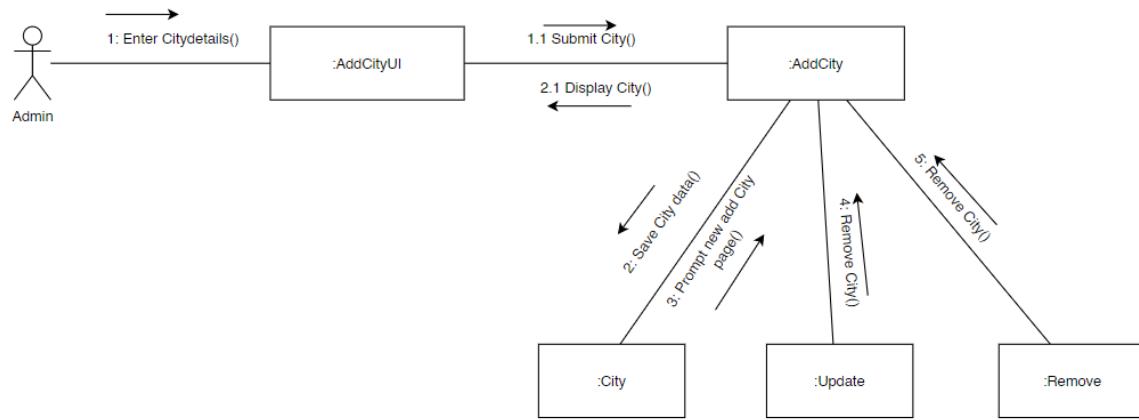


Figure 74: Collaboration diagram for add cities.

### 3.7.3.2.13.4. Sequence Diagram

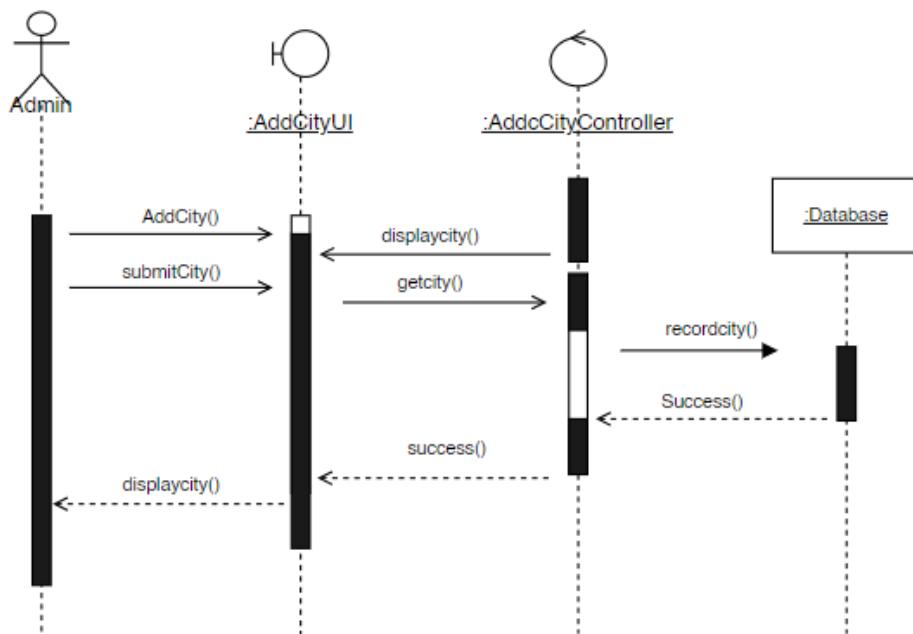


Figure 75: Sequence diagram for add cities.

### 3.7.4. Sprint 3

#### 3.7.4.1. User Interface

User interface (UI) design is the process designers use to build interfaces in software or computerized devices, focusing on looks or style. The User Interface of the project are attached below in an image format: -

##### 3.7.4.1.1. User Interface for Homepage

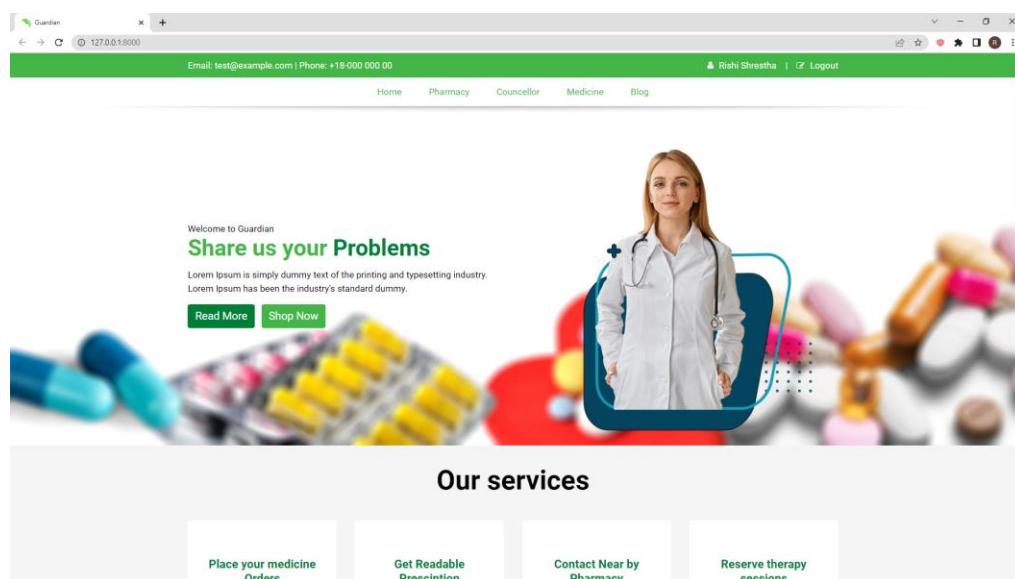


Figure 76: Screenshot of homepage.

##### 3.7.4.1.2. User Interface for Log in page

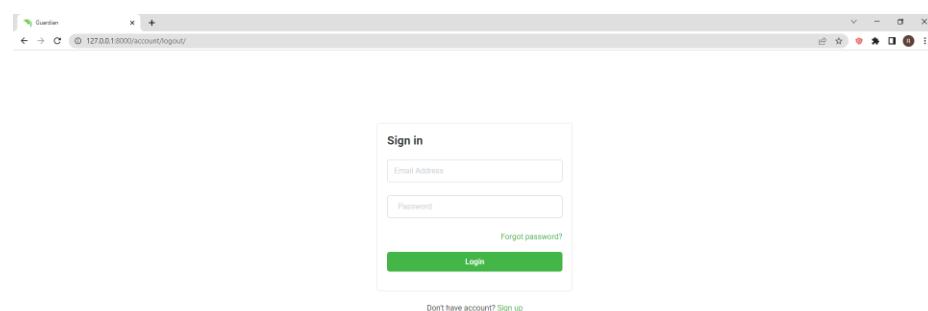
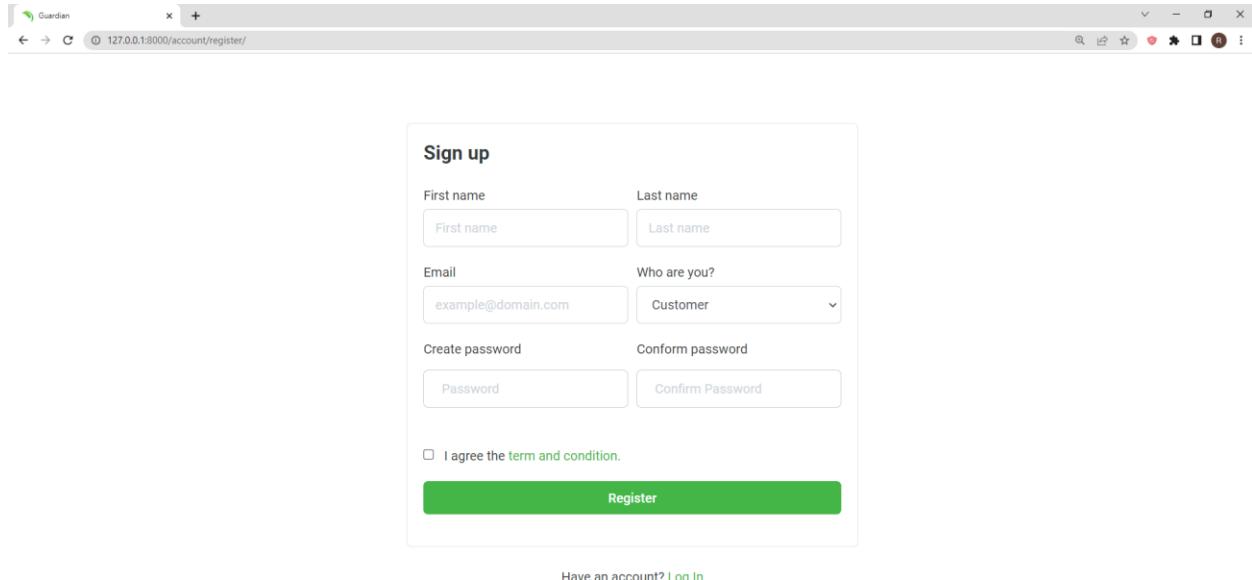


Figure 77: Screenshot of login page.

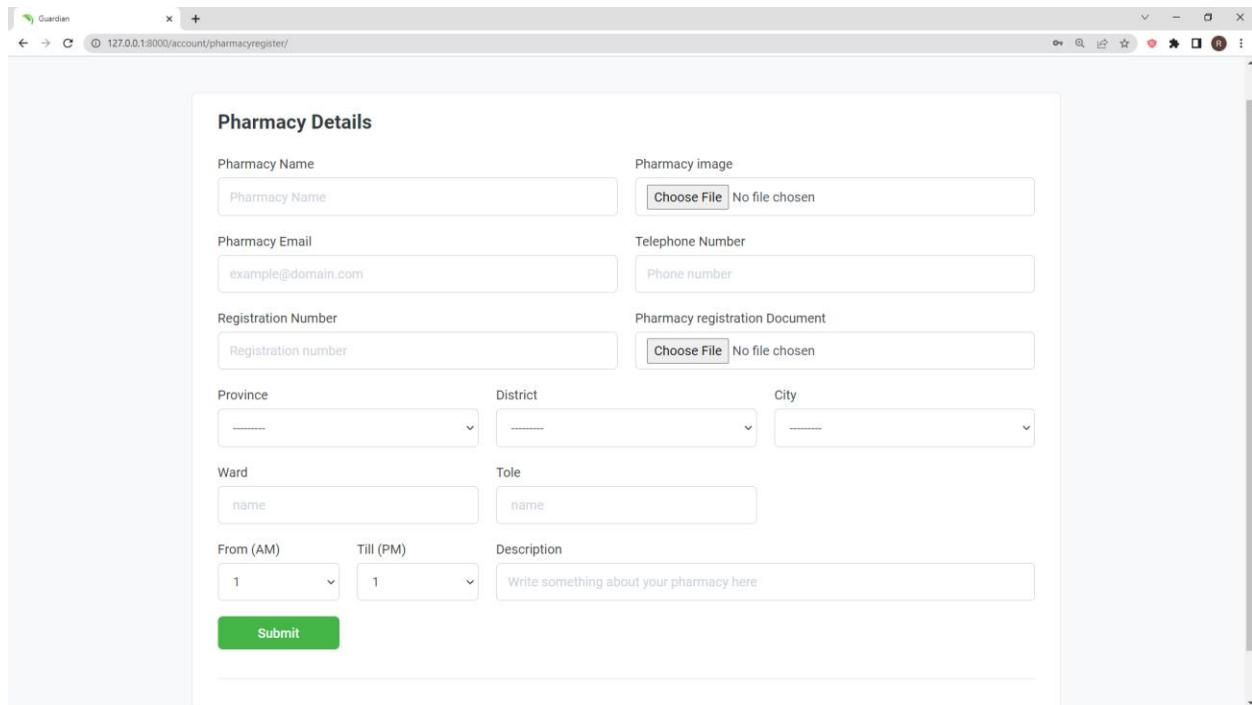
### 3.7.4.1.3. User Interface for Register page



The screenshot shows a web browser window with the title bar 'Guardian'. The address bar displays the URL '127.0.0.1:8000/account/register/'. The main content area is a form titled 'Sign up'. It contains fields for First name and Last name, both with placeholder text 'First name' and 'Last name'. There are also fields for Email (placeholder 'example@domain.com') and Who are you? (dropdown menu showing 'Customer'). Below these are fields for Create password and Conform password, both with placeholder text 'Password'. A checkbox labeled 'I agree the term and condition.' is present, followed by a large green 'Register' button. At the bottom of the form, there is a link 'Have an account? Log In'.

Figure 78: Screenshot of user registration page.

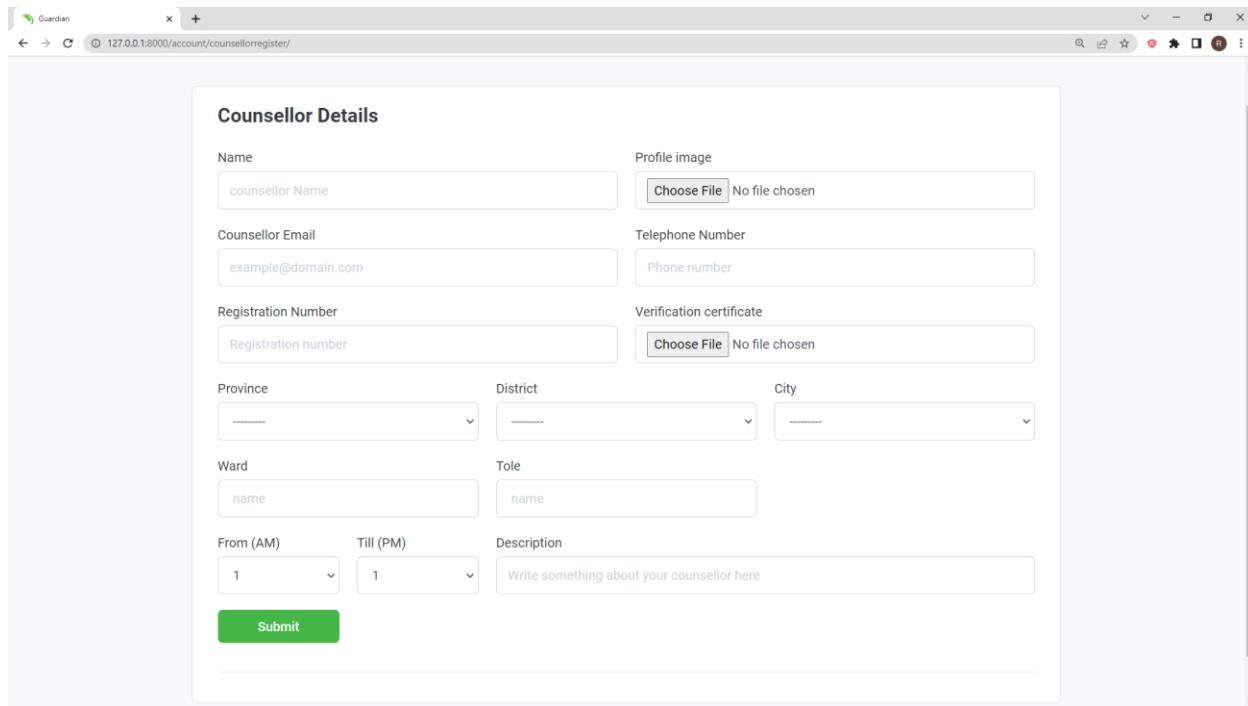
### 3.7.4.1.4. User Interface for Pharmacy Register page



The screenshot shows a web browser window with the title bar 'Guardian'. The address bar displays the URL '127.0.0.1:8000/account/pharmacyregister/'. The main content area is a form titled 'Pharmacy Details'. It includes fields for Pharmacy Name (placeholder 'Pharmacy Name'), Pharmacy image (button 'Choose File' with message 'No file chosen'), Pharmacy Email (placeholder 'example@domain.com'), Telephone Number (placeholder 'Phone number'), Registration Number (placeholder 'Registration number'), Pharmacy registration Document (button 'Choose File' with message 'No file chosen'), Province (dropdown menu), District (dropdown menu), City (dropdown menu), Ward (placeholder 'name'), Tole (placeholder 'name'), From (AM) (dropdown menu with value '1'), Till (PM) (dropdown menu with value '1'), and Description (text area with placeholder 'Write something about your pharmacy here'). A green 'Submit' button is located at the bottom left of the form.

Figure 79: Screenshot of pharmacy registration page.

### 3.7.4.1.5. User Interface for Counsellor Register page



The screenshot shows a web-based form titled "Counsellor Details". The form includes fields for Name (with placeholder "counsellor Name"), Profile image (with a "Choose File" button and message "No file chosen"), Counsellor Email (placeholder "example@domain.com"), Telephone Number (placeholder "Phone number"), Registration Number (placeholder "Registration number"), Verification certificate (with a "Choose File" button and message "No file chosen"), Province (dropdown menu), District (dropdown menu), City (dropdown menu), Ward (text input "name"), Tole (text input "name"), From (AM) and Till (PM) dropdown menus both set to "1", a Description text area ("Write something about your counsellor here"), and a green "Submit" button.

Figure 80: Screenshot of counsellor registration page.

### 3.7.4.1.6. User Interface for City page

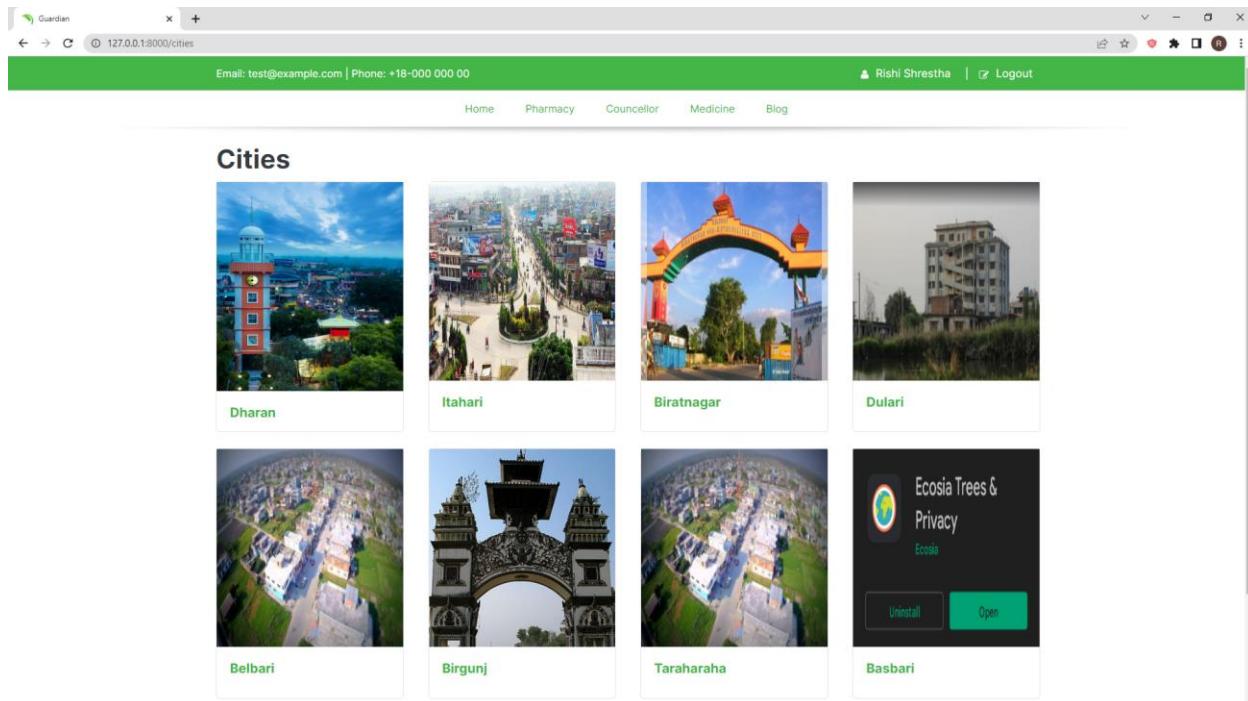


Figure 81: Screenshot of city page.

### 3.7.4.1.7. User Interface for Shop page

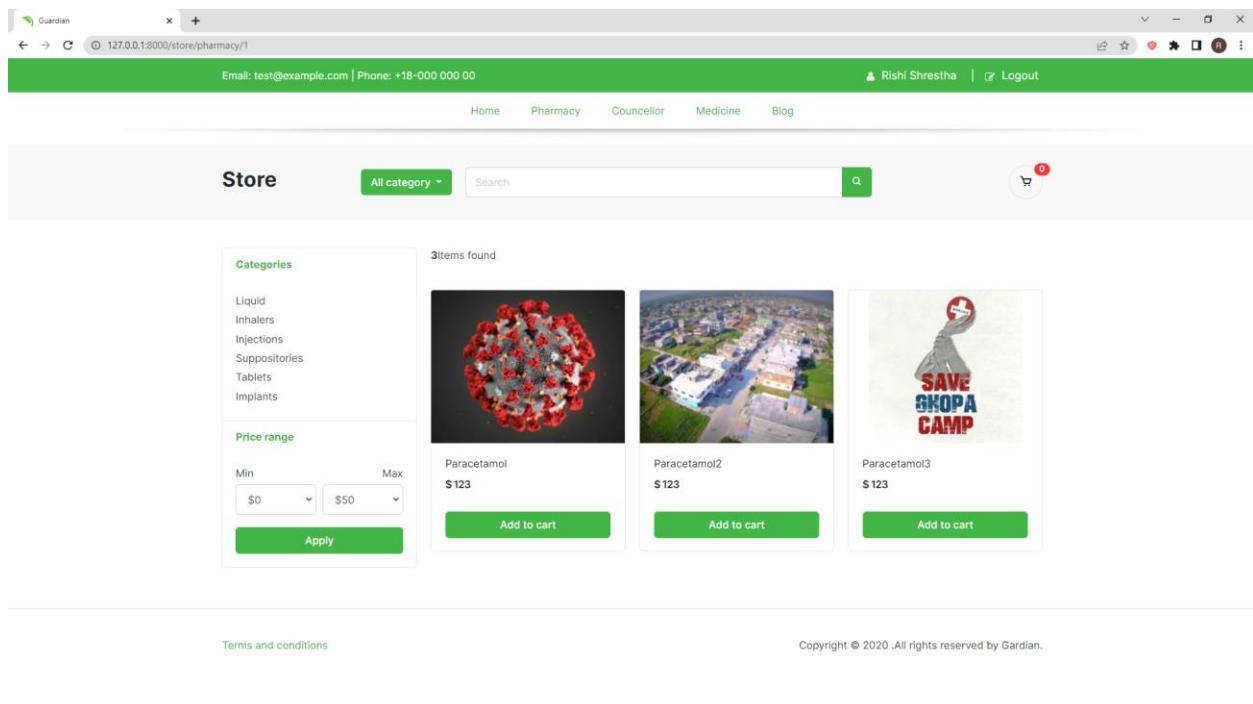


Figure 82: Screenshot of shop page.

### 3.7.4.1.8. User Interface for Booking page

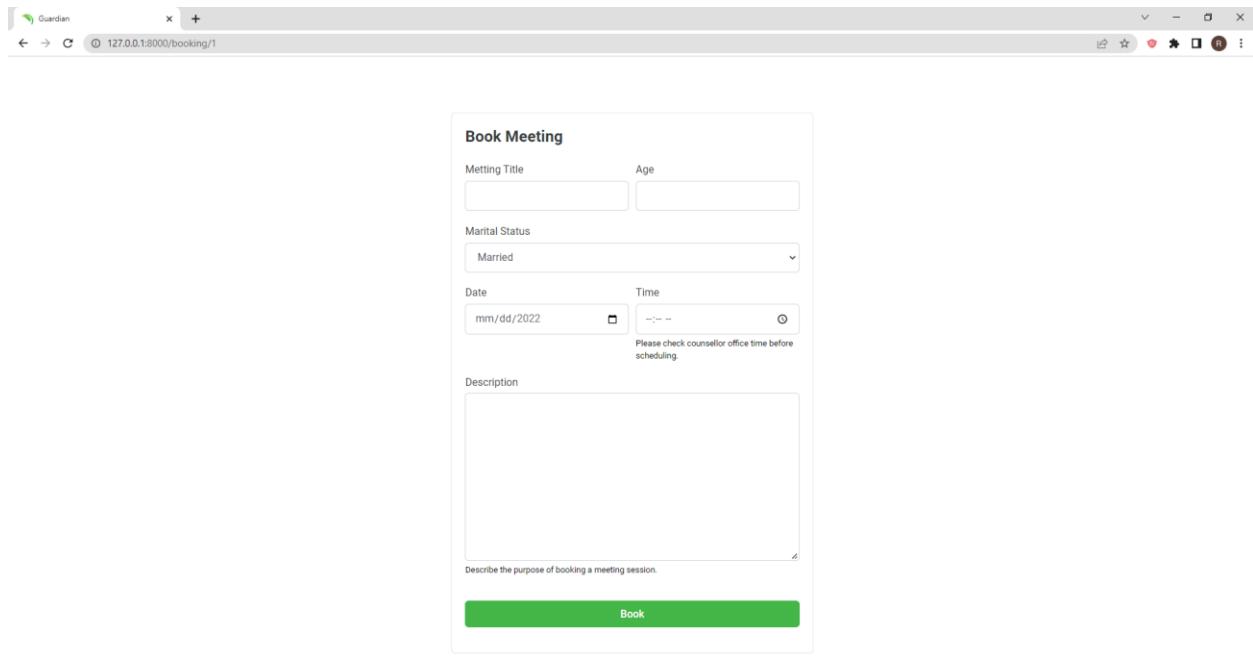


Figure 83: Screenshot of booking page.

### 3.8. Implementation

#### 3.8.1. Database design

Database design is a set of techniques that guide in the creation, development, implementation, and upkeep of enterprise data management systems. A well-designed database is simple to manage, increases data consistency, and saves money on disk storage space. The primary goals of database design in DBMS are to generate logical and physical models of the proposed database system.

An entity relationship diagram (ERD), also known as an entity relationship model, is a graphical representation that depicts relationships among people, objects, places, concepts or events within an information technology (IT) system. (Biscobing, 2021) The ER diagram shown below represents the data model of the Guardian web application. It represents the relationship between tables interlinked and interconnected with each other helping to define and show concepts about entity relationship.

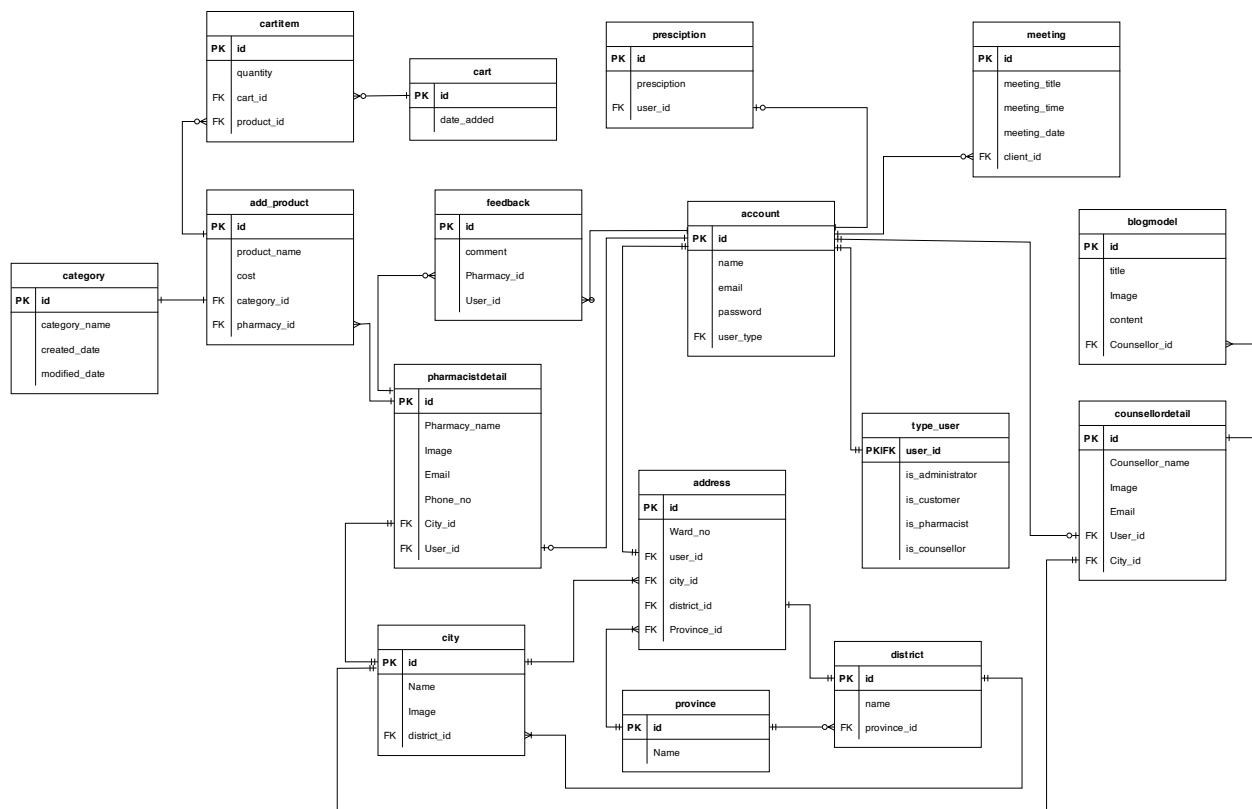


Figure 84: Entity Relationship Diagram (Drawn with help of draw.io)

The above figure is an Entity relationship diagram created before the database construction.

The Entity relationship diagram below was generated with the help of Db Visualizer after the database construction. For the database SQLite was used during development whereas later after the completion of project development PostgreSQL was used.

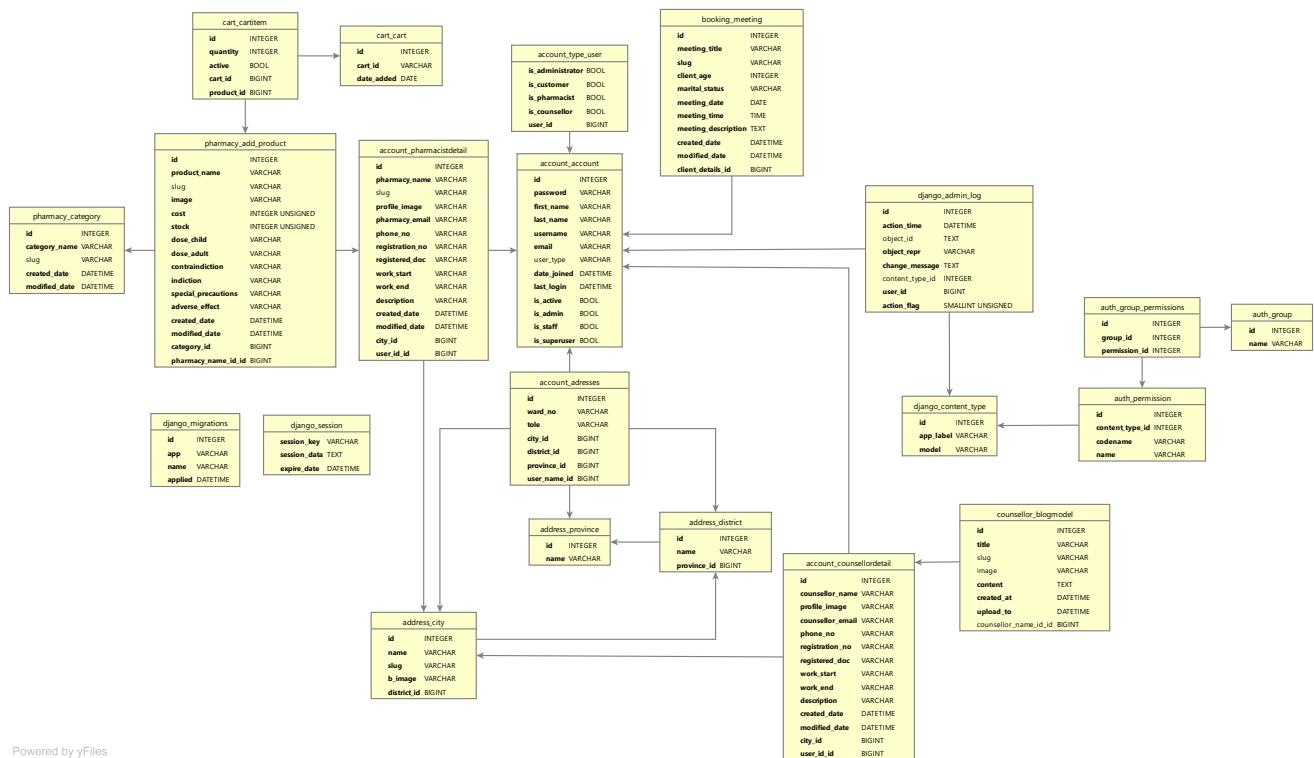


Figure 85: Entity Relationship Diagram (generated by Db visualizer)

Powered by yFiles

The Business rules and relationship of all the entities are:

- The user can have only one user type.
- The customer can view cities, blogs and counsellors.
- Pharmacies are stored according to their cities.
- One user can only register one pharmacy
- A pharmacist can add multiple medicines.
- A counsellor can add multiple blogs.
- A user can read multiple blogs.
- A user can make payment according to their own choices.
- A user can book counsellor.
- A user can add review multiple times.
- A product can have multiple items.
- A user can buy from multiple stores.
- Admin can add cities.
- Medicines can be viewed according to their categories.
- A admin can update users roles and permissions.

### 3.8.2. System Architecture

A system architecture is a conceptual model that defines a system's structure and behaviour. Django follows a Model-View-Controller (MVC) architecture, which is split up into three different parts:

- The Model is the logical data structure behind the entire application and is represented by a database (generally relational databases such as SQLite, Postgres).
- The View is the user interface — what you see in your browser when you visit a website. These are represented by HTML/CSS/JavaScript files.
- The Controller is the middleman that connects the view and model together, meaning that it is the one passing data from the model to the view.

With MVC, your application will revolve around the model—either displaying it or manipulating it.

So, say a user will enter a URL in their browser, that request will go through the internet protocols, to your server, which will call Django. Django will then process the given URL path, and if it matches an URL path you have explicitly stated, it will call the Controller, which will then perform a certain action, such as get an entry from your Model(database) and then render a View (i.e.: JSON text, HTML/CSS/JavaScript Web page).

The following diagram represents the architectural design of my application. Here, when a user submits a request with the application, a route associated with that request maps it to the appropriate controller. Then that controller interacts with the necessary model(s) to retrieve information from the database. Once the controller gets that data passed back the view grabs the data from the controller and renders the view to the user according to the templates. (Geeks for Geeks, 2021)

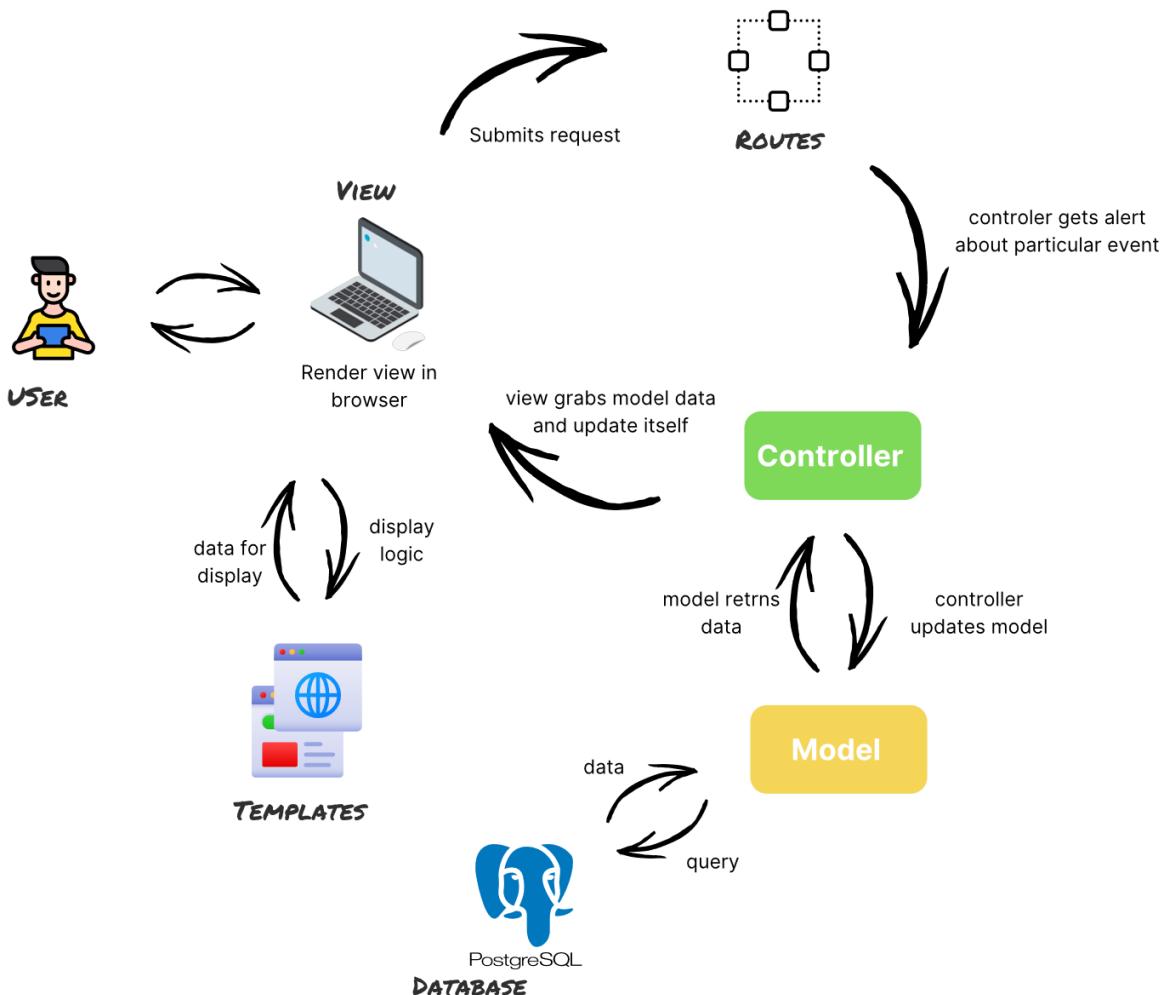


Figure 86: System Architecture of the project.

### 3.8.3. Important Screenshots

```

from django.db import models
from django.contrib.auth.models import AbstractBaseUser, BaseUserManager
from address.models import Province, District, City
from .gen_slug import *
from django.urls import reverse

class MyAccountManager(BaseUserManager):
    def create_user(self, first_name, last_name, username, email, password, user_type):
        if not email:
            raise ValueError('Users must have an email address')
        if not username:
            raise ValueError('Users must have a username')

        user = self.model(
            email=self.normalize_email(email),
            username=username,
            first_name=first_name,
            last_name=last_name,
            user_type=user_type,
        )

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

    def create_superuser(self, first_name, last_name, email, username, password, user_type):
        user = self.create_user(
            email=self.normalize_email(email),
            username=username,
            password=password,
            first_name=first_name,
            last_name=last_name,
            user_type=user_type,
        )
        user.is_superuser = True
        user.is_staff = True
        user.is_admin = True
        user.save(using=self._db)
        return user

class Account(AbstractBaseUser):
    first_name = models.CharField(max_length=30)
    last_name = models.CharField(max_length=30)
    username = models.CharField(max_length=30, unique=True)
    email = models.EmailField(max_length=100, unique=True)
    user_type = models.CharField(max_length=30, null=True, blank=True)

    # required
    date_joined = models.DateTimeField(auto_now_add=True)
    last_login = models.DateTimeField(auto_now=True)
    is_active = models.BooleanField(default=False)
    is_admin = models.BooleanField(default=False)
    is_staff = models.BooleanField(default=False)
    is_superuser = models.BooleanField(default=False)

    USERNAME_FIELD = 'email'
    REQUIRED_FIELDS = ['username', 'first_name', 'last_name', 'user_type']

    objects = MyAccountManager()

    def __str__(self):
        return self.email

    def has_perm(self, perm, obj=None):
        return self.is_admin

    def has_module_perms(self, app_label):
        return True

class Type_user(models.Model):
    is_administrator = models.BooleanField(default=False)
    is_customer = models.BooleanField(default=False)
    is_pharmacist = models.BooleanField(default=False)
    is_counselor = models.BooleanField(default=False)
    user = models.OneToOneField(
        Account, on_delete=models.CASCADE, primary_key=True)

    def __str__(self):
        return self.user.username

```

Figure 87: Custom user Model

```

from urllib import response
from django.test import TestCase
from selenium import webdriver
from .forms import RegistrationForm
from .models import Account, Type_user

class UnitTestCase(TestCase):

    def test_home_homepage_template(self):
        response = self.client.get('/')
        self.assertTemplateUsed(response, 'home.html')

    def test_home_status_code(self):
        response = self.client.get('/')
        self.assertEqual(response.status_code, 200)

    def test_register_form(self):
        form = RegistrationForm(data={'first_name': 'test', 'last_name': 'test', 'email': 'test@gmail.com',
                                      'password': 'testtest', 'confirm_password': 'testtest', 'term': 'True', 'user_type': 'pharmacist'})
        self.assertTrue(form.is_valid())

    def test_account_object(self):
        user = Account.objects.create_user(
            first_name='test',
            last_name='test',
            username='test',
            email='test@gmail.com',
            password='testtest',
            user_type='pharmacist'
        )
        user.save()
        # pulled_user = Account.objects.get(id=1)
        pulled_user = Account.objects.get(email='test@gmail.com')
        self.assertEqual(user.first_name, pulled_user.first_name)
        self.assertEqual(user.last_name, pulled_user.last_name)
        self.assertEqual(user.username, pulled_user.username)
        self.assertEqual(user.email, pulled_user.email)
        self.assertEqual(user.password, pulled_user.password)
        self.assertEqual(user.user_type, pulled_user.user_type)

```

Figure 88: Testing account object.

```

from .models import Cart, CartItem
from .views import _cart_id

def counter(request):
    cart_count = 0
    if 'admin' in request.path:
        return {}
    else:
        try:
            cart = Cart.objects.filter(cart_id=_cart_id(request))
            cart_items = CartItem.objects.all().filter(cart=cart[:1])
            for cart_item in cart_items:
                cart_count += cart_item.quantity
        except Cart.DoesNotExist:
            cart_count = 0
    return dict(cart_count=cart_count)

```

Figure 89: Context Processor for Cart item

```

from django.shortcuts import render, redirect, get_object_or_404
from .models import *
from pharmacy.models import Add_product
from django.contrib.auth.decorators import login_required

# Create your views here.

def _cart_id(request):
    cart = request.session.session_key
    if not cart:
        cart = request.session.create()
    return cart

@login_required(login_url='login')
def add_cart(request, product_id):
    product = Add_product.objects.get(id=product_id)
    try:
        cart = Cart.objects.get(cart_id=_cart_id(request))
    except Cart.DoesNotExist:
        cart = Cart.objects.create(
            cart_id = _cart_id(request)
        )
        cart.save()
    try:
        cart_item = CartItem.objects.get(product=product, cart=cart)
        cart_item.quantity += 1
        cart_item.save()
    except CartItem.DoesNotExist:
        cart_item = CartItem.objects.create(
            product = product,
            quantity = 1,
            cart = cart,
        )
        cart_item.save()
    return redirect('cart')

def remove_cart(request, product_id):
    cart = Cart.objects.get(cart_id=_cart_id(request))
    product = get_object_or_404(Add_product, id=product_id)
    cart_item = CartItem.objects.get(product=product, cart=cart)
    if cart_item.quantity > 1:
        cart_item.quantity -= 1
        cart_item.save()
    else:
        cart_item.delete()
    return redirect('cart')

def remove_cart_item(request, product_id):
    cart = Cart.objects.get(cart_id=_cart_id(request))
    product = get_object_or_404(Add_product, id=product_id)
    cart_item = CartItem.objects.get(product=product, cart=cart)
    cart_item.delete()
    return redirect('cart')

def cart(request, total=0, quantity=0, cart_items=None):
    try:
        cart = Cart.objects.get(cart_id=_cart_id(request))
        cart_items = CartItem.objects.filter(cart=cart, active=True)
        for cart_item in cart_items:
            total += (cart_item.product.cost * cart_item.quantity)
            quantity += cart_item.quantity
        shipping_cost = (2 * total)/100
        grand_total = total + shipping_cost
    except ObjectDoesNotExist:
        pass

    context = {
        'total' : total,
        'quantity' : quantity,
        'cart_items' : cart_items,
        'shipping_cost' : shipping_cost,
        'grand_total' : grand_total,
    }
    return render(request, 'cart.html', context)

```

Figure 90: Cart Views.

*Figure 91: Booking View with Calendar API being Implemented.*

#### 4. Testing and Analysis

Testing is the process of assessing and confirming that a software product or application performs as expected. The advantages of testing include the prevention of bugs, the reduction of development expenses, and the improvement of performance. For the betterment of software testing is very important as it can identify any bugs or errors in a software. Properly tested software ensures reliability, security and high performance which further results in time saving, cost effectiveness and customer satisfaction. (Hamilton, 2022)

Typically, there are three types of testing: -

- Functional Testing
- Non-Functional Testing and
- Maintenance

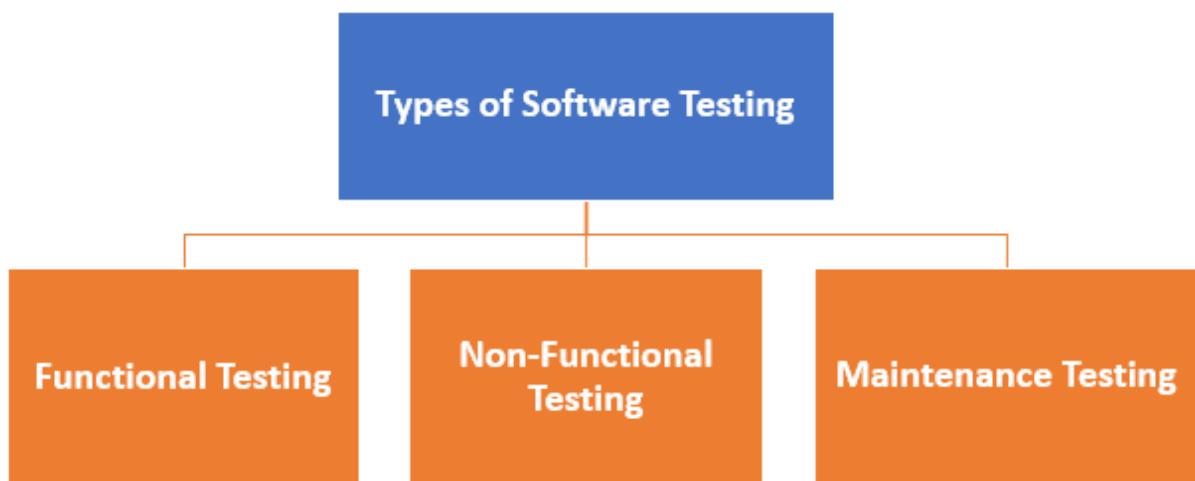


Figure 92: Types of Software testing.

#### Benefits of Software testing

The benefits of using software testing are explained below: -

- Cost-Effective: It is one of the important advantages of software testing. Testing any IT project on time helps you to save your money for the long term. In case if the bugs caught in the earlier stage of software testing, it costs less to fix.

- Security: It is the most vulnerable and sensitive benefit of software testing. People are looking for trusted products. It helps in removing risks and problems earlier.
- Product quality: It is an essential requirement of any software product. Testing ensures a quality product is delivered to customers.
- Customer Satisfaction: The main aim of any product is to give satisfaction to their customers. UI/UX Testing ensures the best user experience.

## Testing Strategies

Some of the important strategies of Software testing are:

a. Unit Testing

This software testing basic approach is followed by the programmer to test the unit of the program. It helps developers to know whether the individual unit of the code is working properly or not.

b. Integration testing

It focuses on the construction and design of the software. You need to see that the integrated units are working without errors or not.

c. System testing

In this method, your software is compiled as a whole and then tested as a whole. This testing strategy checks the functionality, security, portability, amongst others.

d. Acceptance testing

In this method of software testing, a system is tested for acceptability. The aim of this test is to evaluate the compliance of the system with the business requirements and assess whether it is acceptable for delivery or not.

(Hamilton, 2022)

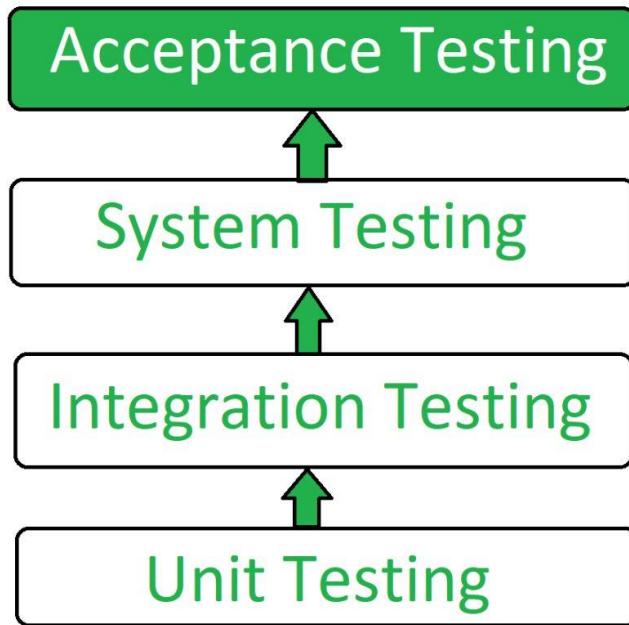


Figure 93: Flow of testing strategies.

## 4.1. Test Plan

### 4.1.1. Testing Plan

The plans for unit testing are listed below: -

- To check if the login button appears after user login.
- To logout from the web application
- To check if the medicine is added to card and count is displayed or not.
- To check if the medicine is added to card and count is displayed or not.
- To check if the cart page shows message or not.
- To add and subtract individual item from the cart page.
- To check if the booking button is clickable or not. If clickable it renders booking form or not.
- To check if the individual pharmacy shows its details or not.
- To check if the individual counsellor shows its details or not.
- To check if the medicine details are displayed or not.
- To check if blog content is displayed on the individual blog page

- To check if the PayPal sign in form will pop or not.
- To test if a new user is registered successfully.
- To test if a new user with a role pharmacist can register their pharmacy.
- To test if a new user with a role counsellor can register themselves as counsellor.
- To test if a user can login successfully.
- To test if forget password link is working.
- To test if reset password button is working.
- To test if an admin can login successfully
- To check if cities are displayed on the city page.
- To check if pharmacy is displayed on the city page.
- To check if admin can view user details.
- To check if admin can view document of users.
- To check if the admin can approve user.
- To check if the added medicine is store in database
- To check if the added blog is store in database
- To check if the PayPal payment is successful.
- To test if an error message is displayed when user tries to login unverified or login with bad credential.
- To test if the error message is displayed when a user tries to log in with an already registered email.
- To test if pharmacist can add a new medicine.
- To test if counsellor can add a new blog.
- To test if admin and user can view all products.
- To test if admin and user can view all blogs.
- To test if pharmacist can delete a medicine
- To test if counsellor can delete a blog.
- To run the application in different browsers.
- To run the application in online terminal for deployment.

## 4.2. Black Box Testing

### 4.2.1. Testing Login button

Table 29: Testing on login button 1.

<b>Objective</b>	To check if the login button appears after user login or not.
<b>Action</b>	Login with a valid email and password and check for login button.
<b>Expected Result</b>	There must be logout button.
<b>Actual Result</b>	The login button was found even after the user login.
<b>Conclusion</b>	Test Unsuccessful.

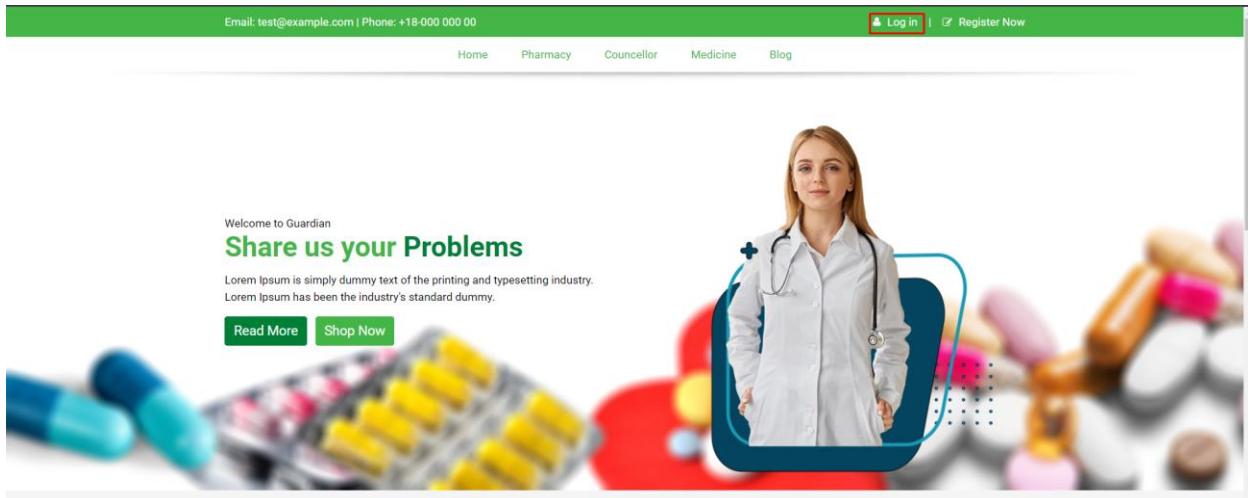


Figure 94: Screenshot of Homepage before user login.

The image above shows the user is not logged in into the web application.

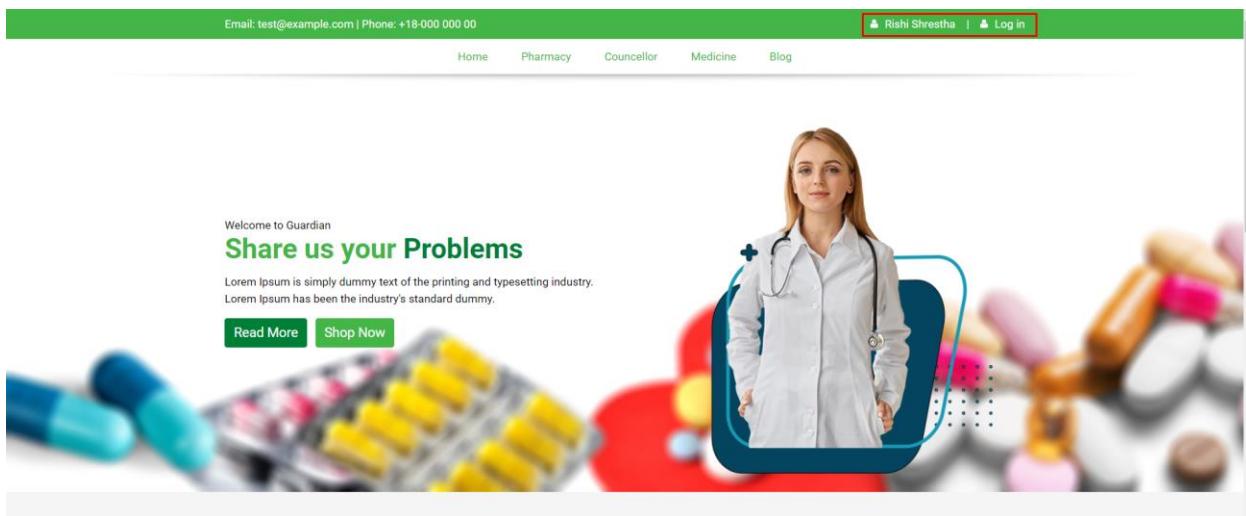


Figure 95: Screenshot of homepage after user login.

The image above shows the user named Rishi Shrestha is logged in into the application but the button beside the username is still logged in.

```

<div class="site-content">
  <header class="topbar">
    <div class="container flex justify-between items-center">
      <div class="icons">
        <p>Email: test@example.com | Phone: +18-000 000 00</p>
      </div>
      {% if user.id is not None %}
        <div class="auth flex items-center">
          <div>
            
            <a style="text-decoration:none !important; color:white" href="#">{{user.first\_name}} {{user.last\_name}}
          </div>
          <span class="divider">|</span>
          <div>
            
            <a style="text-decoration:none !important; color:white" href="#">Log in>
          </div>
        </div>
      {% else %}
        <div class="auth flex items-center">
          <div>
            
            <a style="text-decoration:none !important; color:white" href="#">Log in>
          </div>
          <span class="divider">|</span>
          <div>
            
            <a style="text-decoration:none !important; color:white" href="#">Register Now>
          </div>
        </div>
      {% endif %}
    </div>
  </header>

```

Figure 96: Screenshot of initial code.

As the error of login button appeared after the user is logged in minor change was made on the code. The above image is an initial code, and the image below is the code with solved error.

```

<div class="site-content">
    <header class="topbar">
        <div class="container flex justify-between items-center">
            <div class="icons">
                <p>Email: test@example.com | Phone: +18-000 000 00</p>
            </div>
            {% if user.id is not None %}
            <div class="auth flex items-center">
                <div>
                    
                    <a style="text-decoration:none !important; color:white" href="#">{{user.first_name}} {{user.last_name}}</a>
                </div>
                <span class="divider">|</span>
                <div>
                    
                    <a style="text-decoration:none !important; color:white" href="{% url 'logout' %}">Logout</a>
                </div>
            
```

</div>

```

            {% else %}
            <div class="auth flex items-center">
                <div>
                    
                    <a style="text-decoration:none !important; color:white" href="{% url 'login' %}">Log in</a>
                </div>
                <span class="divider">|</span>
                <div>
                    
                    <a style="text-decoration:none !important; color:white" href="{% url 'register' %}">Register Now</a>
                </div>
            
```

</div>

```

            {% endif %}
        </div>
    </header>

```

Figure 97: Screenshot of after editing the code.

Table 30: Testing login button 2.

<b>Objective</b>	To check if the login button appears after user login or not.
<b>Action</b>	Login with a valid email and password and check for login button.
<b>Expected Result</b>	There must be logout button.
<b>Actual Result</b>	The logout button was found.
<b>Conclusion</b>	Test Successful.

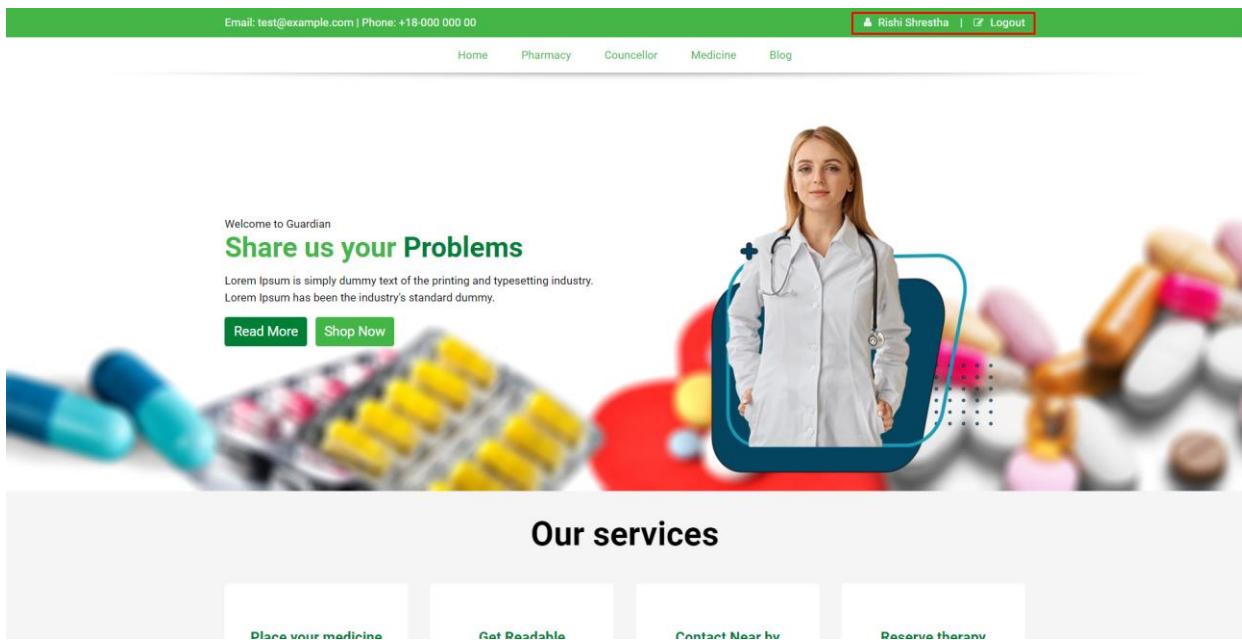


Figure 98: Screenshot of homepage after updating the code.

After updating the code, again the user named Rishi Shrestha logged in into the application, the error on initial test was successfully solved.

#### 4.2.2. Testing logout button

Table 31: Testing logout button.

<b>Objective</b>	To logout from the web application
<b>Action</b>	Click on “Log out” button from the header navigation.
<b>Expected Result</b>	The user should be logged out and must be redirected to login page.
<b>Actual Result</b>	The user was redirected to login page after the action performed.
<b>Conclusion</b>	Test Successful.

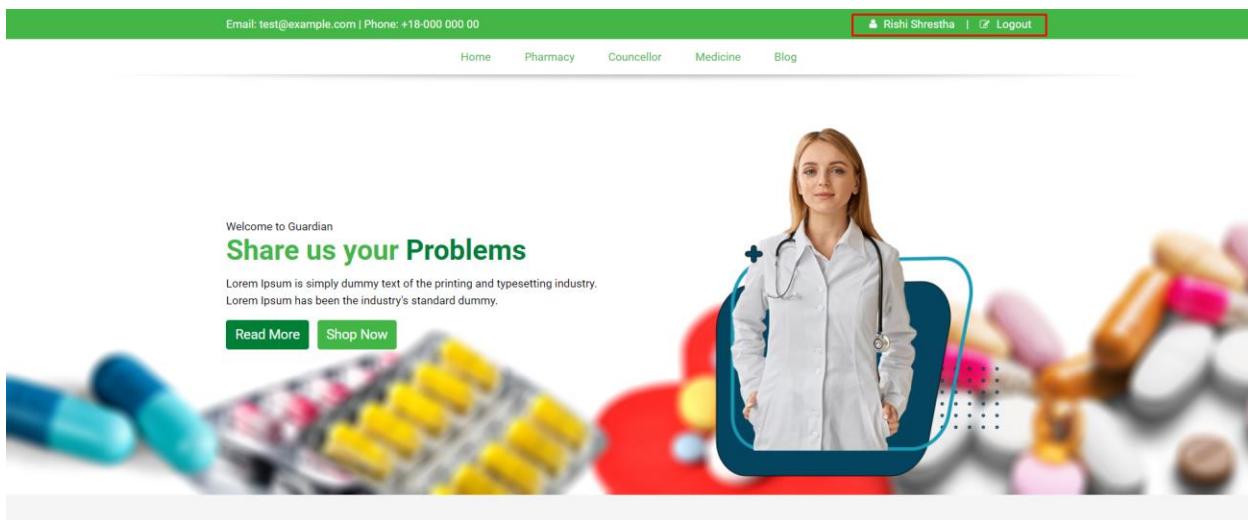


Figure 99: Screenshot of home page.

The image above shows the user is name Rishi Shrestha is logged in into the application.

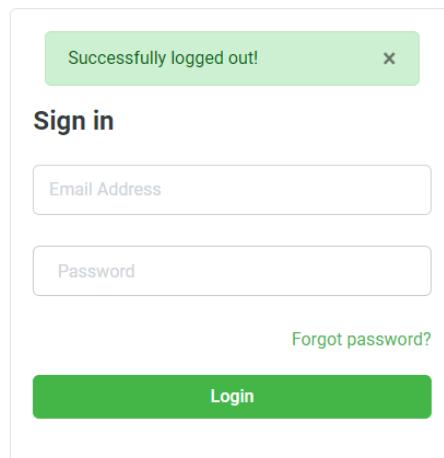


Figure 100: Screenshot of login page.

As soon as the logout button was clicked the sign in page rendered as shown in the above image.

#### 4.2.3. Testing Cart

Table 32: Testing cart 1.

<b>Objective</b>	To check if the medicine is added to card and count is displayed or not.
<b>Action</b>	Add Medicine to cart.
<b>Expected Result</b>	Medicine which was added must be listed on cart page and with each addition the cart count must be increased and vice versa.
<b>Actual Result</b>	After adding medicine onto the cart, the medicines were displayed on the cart page, but the cart count was not increased.
<b>Conclusion</b>	Test Unsuccessful.

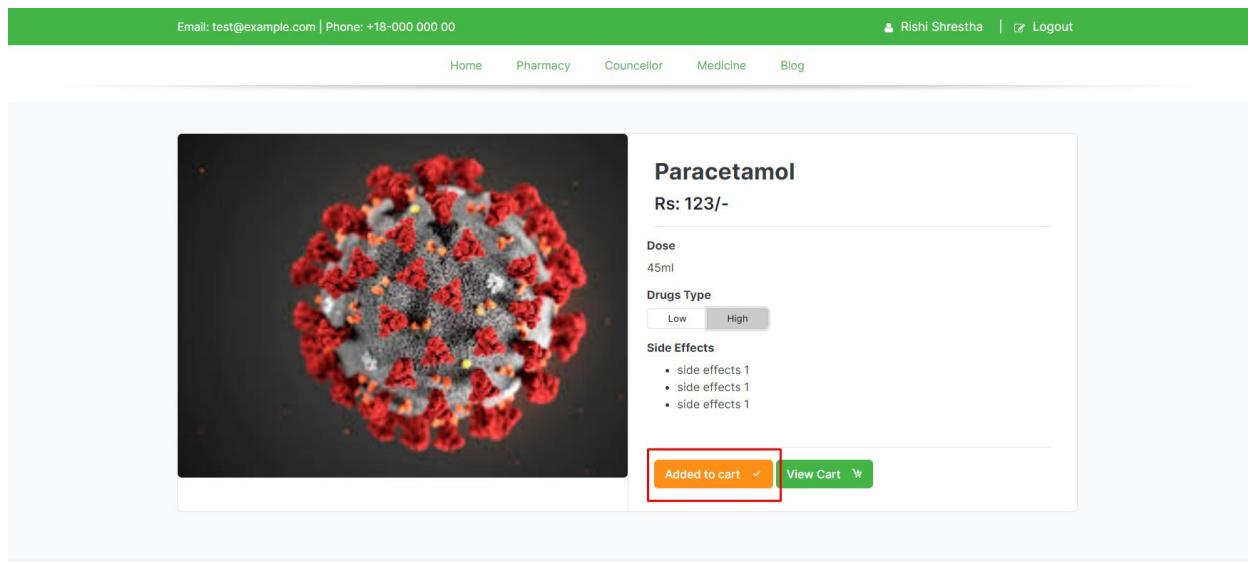


Figure 101: Screenshot of product detail page.

The above image shows the detail of a medicine named Paracetamol and the red rectangle box says that the medicine is already added into the cart.

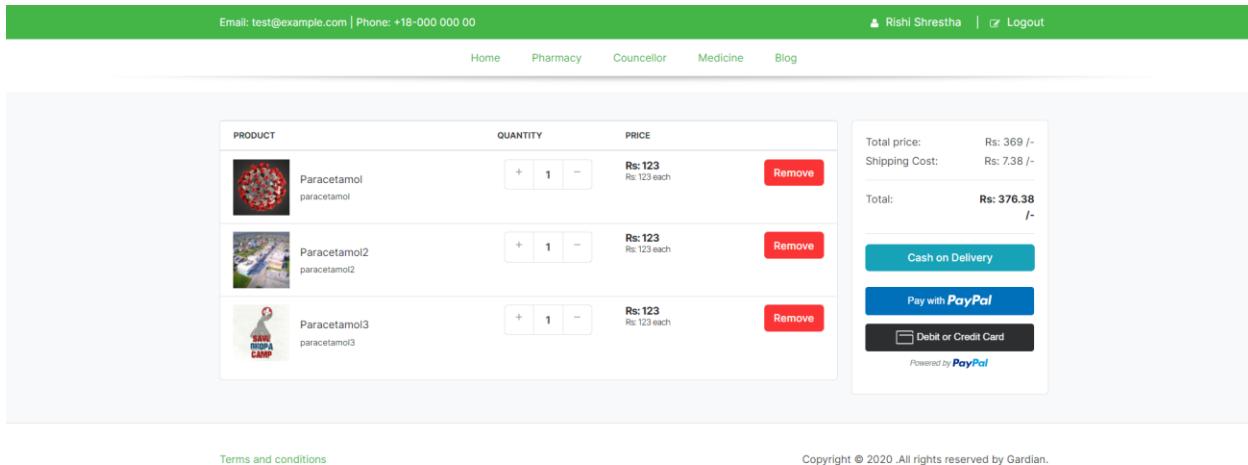


Figure 102: Screenshot of cart page.

The above image shows the cart items. There are three different medicines on the cart.

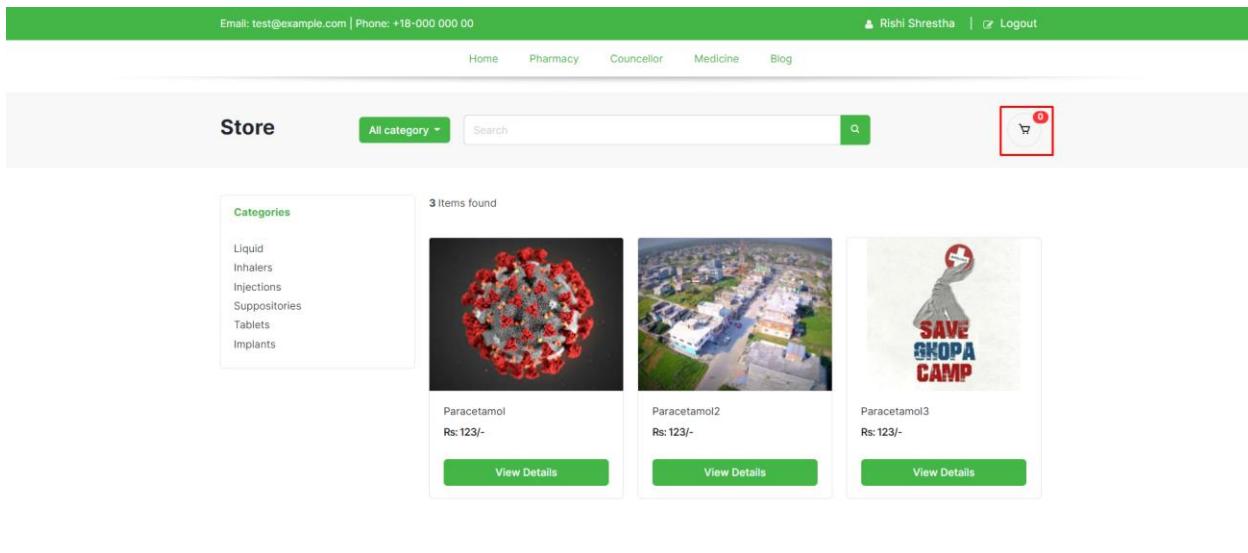


Figure 103: Screenshot of store page.

The image is taken from the store page. The rectangle box represents the cart items are not counted.

```
from django.shortcuts import render, redirect, get_object_or_404
from .models import *
from pharmacy.models import Add_product
from django.contrib.auth.decorators import login_required

# Create your views here.

def _cart_id(request):
    cart = request.session.session_key
    if not cart:
        cart = request.session.create()
    return cart

@login_required(login_url='login')
def add_cart(request, product_id):
    product = Add_product.objects.get(id=product_id)
    try:
        cart = Cart.objects.get(cart_id=_cart_id(request))
    except Cart.DoesNotExist:
        cart = Cart.objects.create(
            cart_id = _cart_id(request)
        )
        cart.save()
    try:
        cart_item = CartItem.objects.get(product=product, cart=cart)
        cart_item.quantity += 1
        cart_item.save()
    except CartItem.DoesNotExist:
        cart_item = CartItem.objects.create(
            product = product,
            quantity = 1,
            cart = cart,
        )
        cart_item.save()
    return redirect('cart')
```

Figure 104: Screenshot of code for cart session.

As into previous image the cart item was not count so session was used as shown in the above image. And again, testing was performed.

Table 33: Testing cart 2

<b>Objective</b>	To check if the medicine is added to card and count is displayed or not.
<b>Action</b>	Add Medicine to cart.
<b>Expected Result</b>	Medicine which was added must be listed on cart page and with each addition the cart count must be increased and vice versa.
<b>Actual Result</b>	After adding medicine onto the cart, the medicines were displayed on the cart page similarly the cart count was also increased. With each addition and was decreased with each removal of items.
<b>Conclusion</b>	Test Successful.

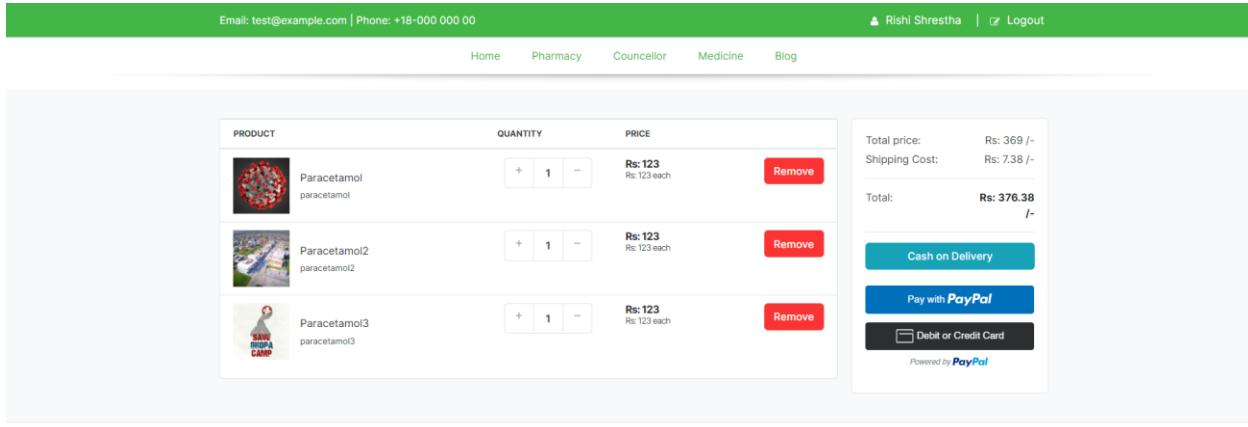


Figure 105: Screenshot of cart page after updating the code.

The above image shows three items on the cart.

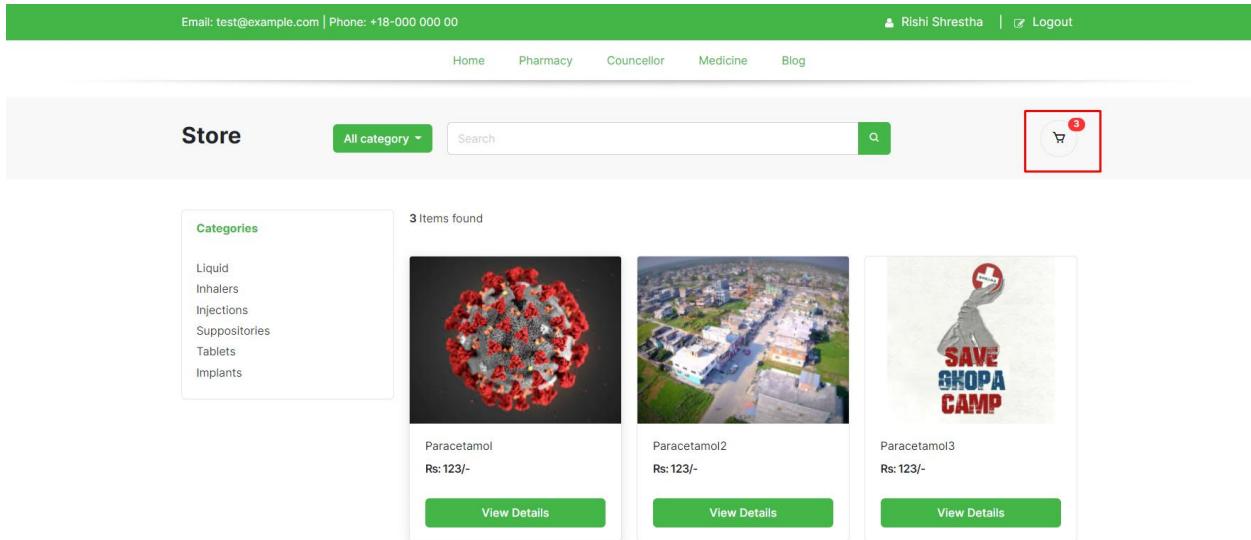


Figure 106: Screenshot of store page after updating the code.

The on the previous image there was three items on the cart so here in this image the red rectangle box shows the cart item count which is three.

#### 4.2.4. Testing Message in Cart page

Table 34: Testing cart message 1.

<b>Objective</b>	To check if the cart page shows message or not.
<b>Action</b>	To view cart page with zero item on the cart.
<b>Expected Result</b>	A message must be shown to the user regarding the empty of cart.
<b>Actual Result</b>	A blank page was shown.
<b>Conclusion</b>	Test unsuccessful.

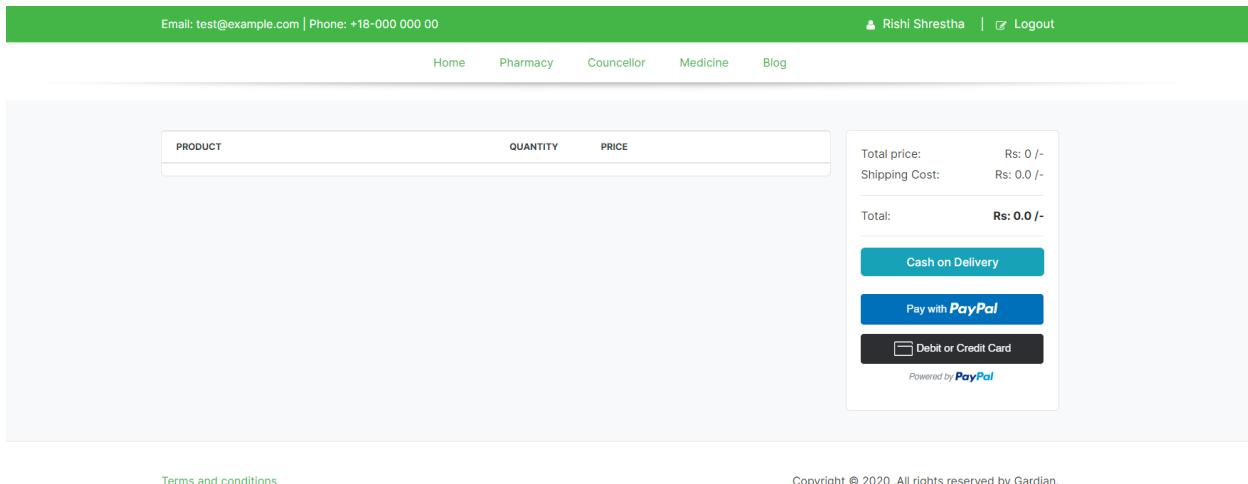


Figure 107: Screenshot of empty cart page.

The above image shows the empty table on cart page.

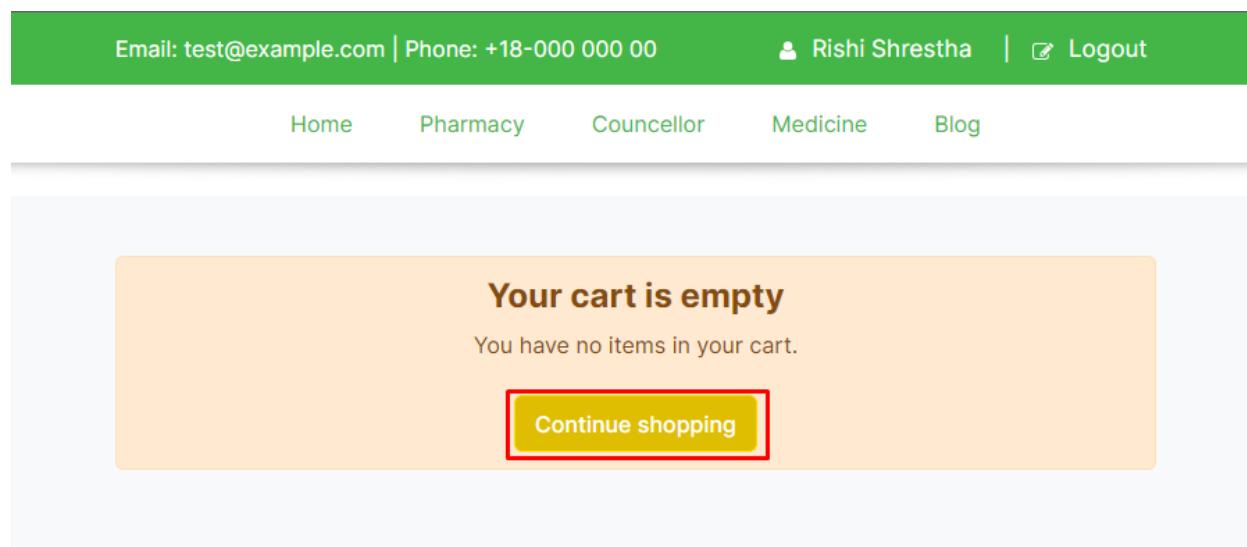
```
{% if not cart_items %}
  <div class="row">
    <div class="col-md-12">
      <div class="alert alert-warning">
        <div class="text-center">
          <h4>Your cart is empty</h4>
          <p>You have no items in your cart. <br><br><a href="{% url 'cities' %}" class="btn btn-success">Continue shopping</a></p>
        </div>
      </div>
    </div>
  </div>
{% else %}
```

Figure 108: Screenshot of code added on cart page.

According to the testing objective the empty table should not be displayed rather some message must be shown so minor changes was done on code as shown in above image and again testing was performed.

Table 35: Testing cart message 2.

<b>Objective</b>	To check if the cart page shows message or not.
<b>Action</b>	To view cart page with zero item on the cart.
<b>Expected Result</b>	A message must be shown to the user regarding the empty of cart.
<b>Actual Result</b>	A valid message was shown with a button that redirects to the cities page to continue their shopping.
<b>Conclusion</b>	Test Successful.



[Terms and conditions](#)

Copyright © 2020 .All rights reserved by Gardian.

*Figure 109: Screenshot of cart page with zero item on cart.*

As changes were made on code. The empty cart shows the message as shown by the above image.

#### 4.2.5. Testing increment and decrement of Cart item

*Table 36: Testing cart item increment and decrement.*

<b>Objective</b>	To add and subtract individual item from the cart page.
<b>Action</b>	Click on (+) button to add item and (-) to reduce its count.
<b>Expected Result</b>	The medicine count must be increase if (+) button is clicked, and (-) button should reduce its count with each click. If the number is equals to 0 then the item must be removed from the cart.
<b>Actual Result</b>	The medicine item count was altered by the (+) and (-) button. Similarly, after the item is less than 1 the item was removed.
<b>Conclusion</b>	Test Successful.

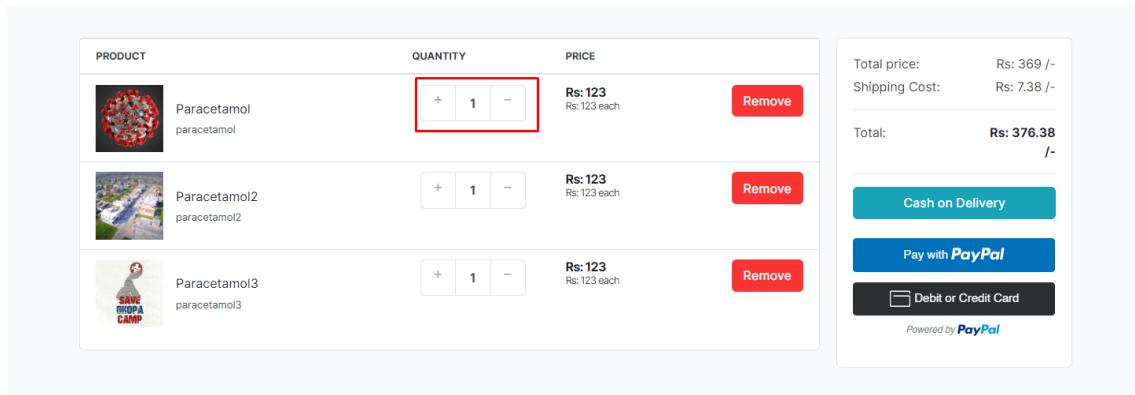


Figure 110: Screenshot of cart page. (1)

The above image shows the items on cart and the rectangle box represent there is one item of particular medicine.

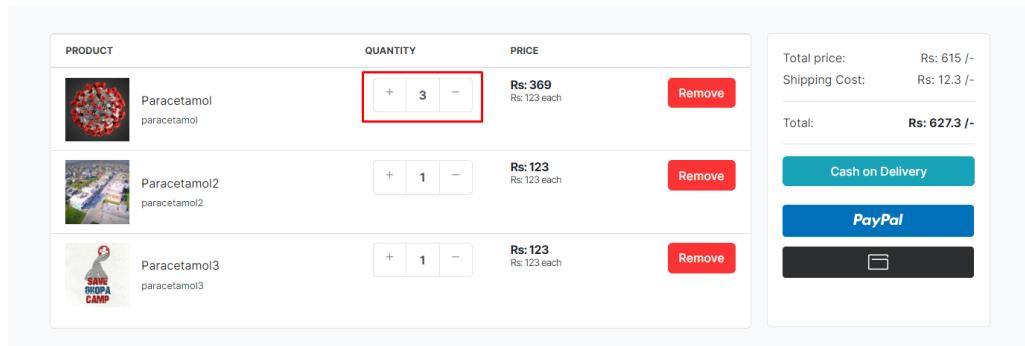


Figure 111: Screenshot of cart page. (2)

After the + button was clicked twice the item count was increased by two and its cost was multiplied by two as well as shown in the above figure.

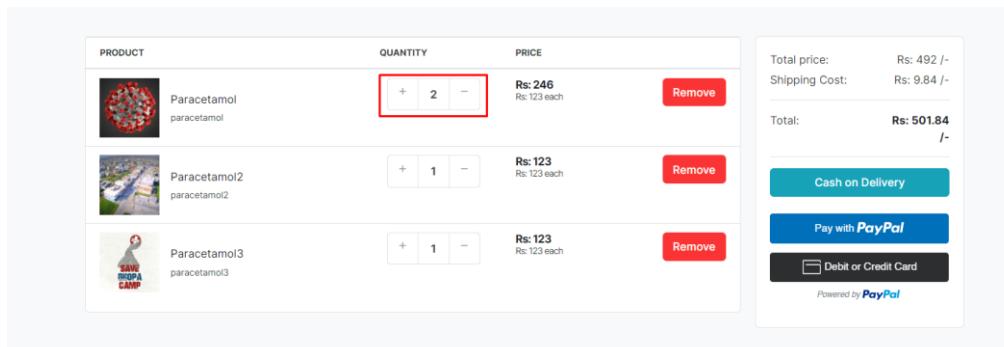


Figure 112: Screenshot of cart page. (3)

Similarly, after the – button was clicked the item count was subtracted by one and its cost was also subtracted as shown in the above figure.

#### 4.2.6. Testing Booking button

Table 37: Testing booking button.

<b>Objective</b>	To check if the booking button is clickable or not. If clickable it renders booking form or not.
<b>Action</b>	To view a counsellor and click on booking button.
<b>Expected Result</b>	A booking form page should render after clicking the booking button.
<b>Actual Result</b>	A new page was rendered. The render page was found to be a booking page.
<b>Conclusion</b>	Test Successful.

Email: test@example.com | Phone: +18-000 000 00

Rishi Shrestha |  Logout

Home Pharmacy Counsellor Medicine Blog

**Rishi Raj Shrestha**

License Number: 123/122/2342345  
 Working Hour : 8:00AM to 10:00PM  
 Working Days : Sun, Mon, Tue, Wen, Thu, Fri  
 Cell Phone number : +977-9810503530  
 Email : example@gmail.com

**Description**  
 this is a description

**Book an Appointment**

Figure 113: Screenshot of counsellor detail page.

The above image shows the details of a counsellor named Rishi Raj Shrestha and the red rectangle box represents the button Book an Appointment.

### Book Meeting

Meeting Title      Age

Marital Status

Married

Date      Time

mm/dd/2022

- : -

Please check counsellor office time before scheduling.

Description

Describe the purpose of booking a meeting session.

Book

Figure 114: Screenshot of booking form page.

The above image shows the booking form page that is rendered after the Book Appointment button is pressed from the previous image.

#### 4.2.7. Testing Pharmacy detail

Table 38: Testing pharmacy detail.

<b>Objective</b>	To check if the individual pharmacy shows its details or not.
<b>Action</b>	To click on a city and select a pharmacy of that city.
<b>Expected Result</b>	Details of pharmacy with a button that redirects to their store must be displayed.
<b>Actual Result</b>	Details of pharmacy with a button that redirects to their store was displayed.
<b>Conclusion</b>	Test Successful.

The screenshot shows a user interface for a city database. At the top, there is a green header bar with the text 'Email: test@example.com | Phone: +18-000 000 00' and a user profile icon. Below the header, there is a navigation menu with links: Home, Pharmacy, Councillor, Medicine, and Blog. The main content area is titled 'Cities' and displays eight city thumbnails arranged in two rows of four. Each thumbnail includes a small image and the city name below it. A red rectangle highlights the thumbnail for 'Dulari', which shows a building by a body of water. The other cities shown are Dharan, Itahari, Biratnagar, Belbari, Birgunj, Taraharaha, and Budi Khola.

Figure 115: Screenshot of city page.

The above image shows the different cities which are added on the database. The red rectangle shows the user is going through that city.

Email: test@example.com | Phone: +18-000 000 00       Rishi Shrestha |  Logout

[Home](#) [Pharmacy](#) [Councillor](#) [Medicine](#) [Blog](#)

## Pharmacies



**Test Pharmacy nepal**  
this is a description of test pharmacy nepal  
[Visit](#)

[Terms and conditions](#)      Copyright © 2020 .All rights reserved by Gardian.

Figure 116: Screenshot of pharmacy page.

The above image shows the pharmacy registered on that city.

Email: test@example.com | Phone: +18-000 000 00       Rishi Shrestha |  Logout

[Home](#) [Pharmacy](#) [Councillor](#) [Medicine](#) [Blog](#)

## Test Pharmacy nepal



**Test Pharmacy nepal**  
OGRM: 099312312  
INN : 099312312  
Working Hour : 9 to 6  
Working Days : Sun, Mon, Tue, Wen, Thu, Fri  
Cell Phone number : 9810503530  
Email : Submittyc@gmail.com

**Description**  
this is a description of test pharmacy nepal  
[Store](#)

Figure 117: Screenshot of pharmacy detail page.

The above image shows the detail of Test Pharmacy Nepal.

#### 4.2.8. Testing Counsellor detail

Table 39: Counsellor detail

<b>Objective</b>	To check if the individual counsellor shows its details or not.
<b>Action</b>	To select a counsellor to view their detail page.
<b>Expected Result</b>	Details of counsellor with a button that redirects to their booking form must be displayed.
<b>Actual Result</b>	Details of counsellor with a button that redirects to their booking form was displayed.
<b>Conclusion</b>	Test Successful.

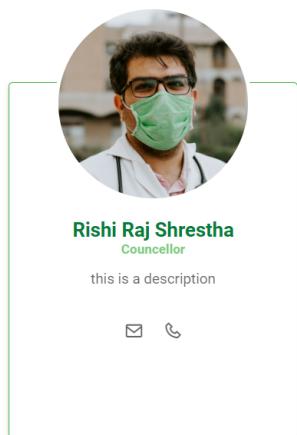


Figure 118: Screenshot of counsellor page.

The above images show all the registered counsellor.

*Figure 119: Screenshot of counsellor detail page.*

The above image shows the detail of a counsellor.

#### 4.2.9. Testing Medicine detail

*Table 40: Testing medicine detail*

<b>Objective</b>	To check if the medicine details are displayed or not.
<b>Action</b>	Click on store button of a pharmacy and click on any medicine.
<b>Expected Result</b>	The medicine detail should be displayed.
<b>Actual Result</b>	The medicine detail was displayed.
<b>Conclusion</b>	Test Successful.

Email: test@example.com | Phone: +18-000 000 00      Rishi Shrestha | Logout

Home Pharmacy Councillor Medicine Blog

**Store**

All category ▾ Search  Cart 1

Categories

- Liquid
- Inhalers
- Injections
- Suppositories
- Tablets
- Implants

3 Items found

	Paracetamol Rs: 123/-		Paracetamol2 Rs: 123/-		Paracetamol3 Rs: 123/-
<input type="button" value="View Details"/>					

[Terms and conditions](#)

Copyright © 2020 .All rights reserved by Gardian.

*Figure 120: Screenshot of store page.*

The above image shows the store with three different medicines. The red rectangle indicates the user is going to select that medicine.

Email: test@example.com | Phone: +18-000 000 00      Rishi Shrestha | Logout

Home Pharmacy Councillor Medicine Blog

**Paracetamol**

Rs: 123/-

Dose  
45ml

Drugs Type  
 Low  High

Side Effects

- side effects 1
- side effects 1
- side effects 1

*Figure 121: Screenshot of medicine detail page.*

The above image shows the detail of medicine named Paracetamol.

#### 4.2.10. Testing Blog Page

Table 41: Testing blog page.

<b>Objective</b>	To check if blog content is displayed on the individual blog page
<b>Action</b>	The Blog page link is clicked from the navigation bar and a blog was clicked.
<b>Expected Result</b>	The blog page should be displayed and detail content of the blog with their writer must be displayed.
<b>Actual Result</b>	The blog page was displayed. Similarly, after clicking an individual blog the detail content of the blog was displayed.
<b>Conclusion</b>	Test Successful.

Figure 122: Screenshot of blog page.

The above image shows all the blogs present in the database.

## Ban of online Medicine

Author: Manish Shrestha

Added on: April 25, 2022, 5:04 a.m.

The **Department of Drug Administration** has put an immediate ban on the sale of drugs online. The department has banned pharmacies selling drugs online through various apps and websites.

Santosh KC, information office of the department, said that the sale of medicines online has been stopped for the time being as there are no legal directives to sell medicines online even though there is a law to sell other items online.

He said that the sale and distribution of medicines online has been banned as businesses have been found to be promoting and advertising medicines through social media and offering various discount facilities to lure customers which is illegal.

He also said that there is no practice of registering any pharmacy in the Department of Drug Management to provide online pharmacy service.

He said that the consumers concerned should be responsible for the safety, quality and effectiveness of such medicines.

The department has also requested the general public and patients not to buy and consume medicines from online pharmacies. He informed that action would be taken as per the rules if found otherwise.

Stating that many online drug stores in India have been shutdown some time ago, he said, "The new technology being adopted all over the world is to be practiced in Nepal as well, but a clear law has to be identified for this matter."

Currently, there are over a dozen of online pharmacy operating in Nepali market. Some of the notable ones include Jeevee, Annapurna Pharmacy, Nepills, ATM Pharmacy and daba124.

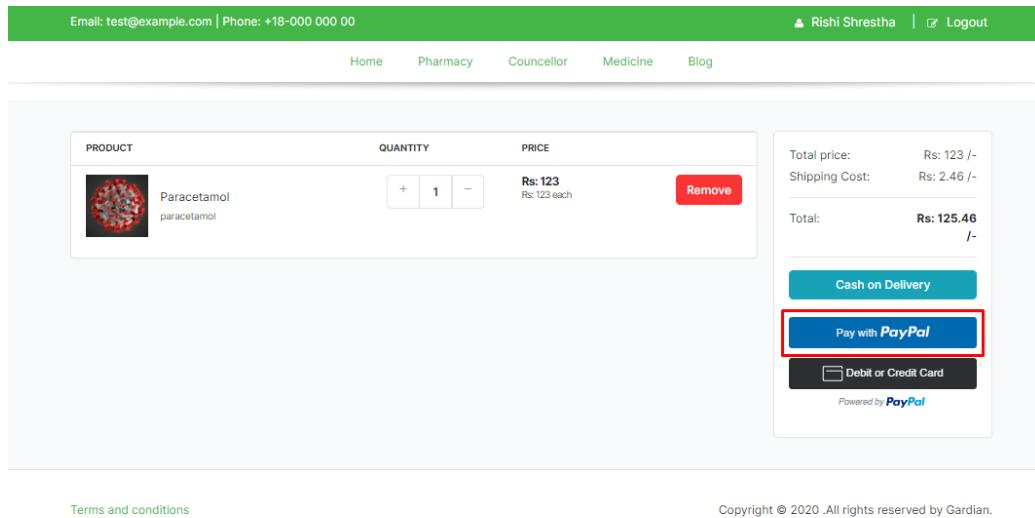
*Figure 123: Screenshot of read blog page.*

The above image is showing the individual blog page.

#### **4.2.11. Testing Payment Button**

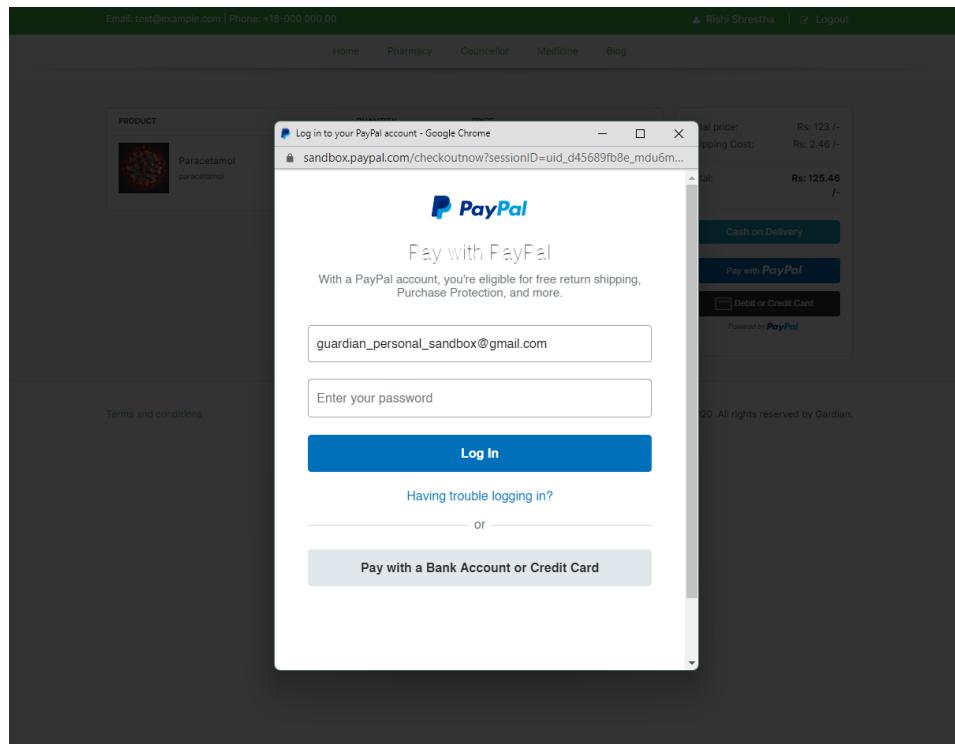
Table 42: Testing payment button.

<b>Objective</b>	To check if the PayPal sign in form will pop or not.
<b>Action</b>	Click on payment with PayPal.
<b>Expected Result</b>	PayPal payment login will pop up after performing the action.
<b>Actual Result</b>	As expected, the PayPal loin form was popped up.
<b>Conclusion</b>	Test Successful.



*Figure 124: Screenshot of cart page.*

The above image is taken from cart page and the rectangle shows the button pay with PayPal and indicates the user is selecting that button.



*Figure 125: Screenshot of PayPal login.*

After the user clicks the pay with PayPal button the sign in of PayPal is popped out.

### 4.3. White Box testing

#### 4.3.1. Testing User registration

Table 43: Whitebox testing for user registration.

<b>Objective</b>	To test if a new user is registered successfully.
<b>Action</b>	Fill out all the details required for the registration form and click register button
<b>Expected Result</b>	<p>A verification email should be sent to the user.</p> <p>After the link send via email is clicked then the user is verified and the user is redirected to login page.</p> <p>The data of new user should be recorded as a normal user database.</p>
<b>Actual Result</b>	<p>A verification mail is sent to the user.</p> <p>After the email is verified the user id redirected to login page.</p> <p>The data of the new user is recorded as a normal user in the database.</p>
<b>Conclusion</b>	Test Successful.

The screenshot shows a user registration form titled "Sign up". The form includes fields for First name and Last name (each in its own input box), Email (input box) and Who are you? (dropdown menu showing "Customer"), Create password and Conform password (each in its own input box), and a checkbox labeled "I agree the term and condition." Below the form is a large green "Register" button. At the bottom of the page, there is a link "Have an account? Log In".

Figure 126: Screenshot of user registration page.

The image shows the registration page with empty text fields.

## Sign up

First name      Last name

Mandip      Gurung

Email      Who are you?

gurungmandip111@gmail.com      Customer

Create password      Conform password

.....      .....

I agree the [term and condition](#).

**Register**

Have an account? [Log In](#)

Figure 127: Screenshot of user registration page with filled form.

The above image shows the registration page with filled data and the rectangle box is indicating the user is going to click Register button.

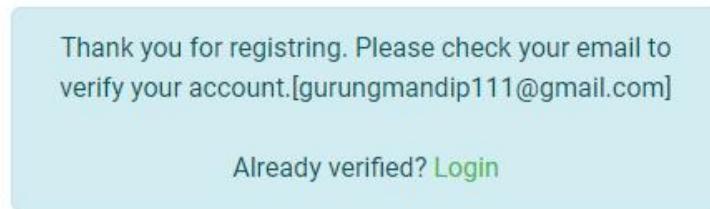


Figure 128: Screenshot of alert message.

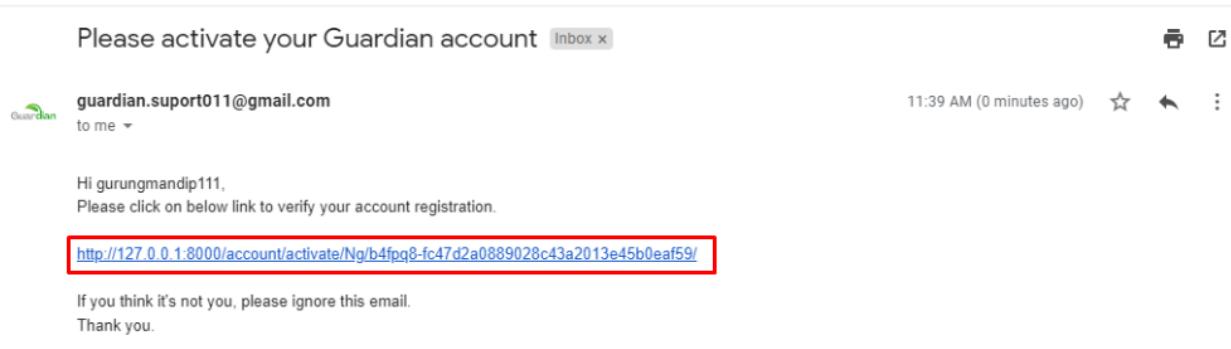
The above image is popped up after the user click Register button. It says the user to verify their email address.

Select account to change								ADD ACCOUNT +
<input type="checkbox"/> EMAIL		FIRST NAME	LAST NAME	USERNAME	DATE JOINED	LAST LOGIN	USER TYPE	IS ACTIVE
<input type="checkbox"/>	test@guardian.com	Guardian_f	I_G	test_guardian	April 14, 2022, 6:21 a.m.	April 14, 2022, 6:21 a.m.	admin	✓
<input type="checkbox"/>	guardian@gaurdian.com	guar	dian	guardian	April 16, 2022, 3:13 p.m.	April 25, 2022, 5:55 a.m.	admin	✓
<input type="checkbox"/>	rrs0053@my.londonmet.ac.uk	Rishi	Shrestha	rrs0053	April 17, 2022, 12:28 a.m.	April 25, 2022, 4:37 a.m.	pharmacist	✓
<input type="checkbox"/>	manishtha338@gmail.com	Manish	Shrestha	manishtha338	April 22, 2022, 7:36 a.m.	April 25, 2022, 5:09 a.m.	counsellor	✓
<input type="checkbox"/>	gurun.123@gma.com	Rishi	Shrestha	gurun.123	April 24, 2022, 4:15 p.m.	April 24, 2022, 4:16 p.m.	pharmacist	✓
<input type="checkbox"/>	gurungmandip111@gmail.com	Mandip	Gurung	gurungmandip111	April 25, 2022, 5:54 a.m.	April 25, 2022, 5:54 a.m.	customer	✗

6 accounts

Figure 129: Screenshot of all the user's account.

The above image is taken by admin. Here the recently registered user is highlighted the red cross on right shows the email is not verified yet.



With Regards,  
Guardian

This email and any files transmitted with it are confidential and intended solely for the use of the individual or entity to whom they are addressed. If you have received this email in error please notify the system manager. This message contains confidential information and is intended only for the individual named. If you are not the named addressee you should not disseminate, distribute or copy this e-mail. Please notify the sender immediately by e-mail if you have received this e-mail by mistake and delete this e-mail from your system. If you are not the intended recipient you are notified that disclosing, copying, distributing or taking any action in reliance on the contents of this information is strictly prohibited.

Figure 130: Screenshot of email verification send to user.

The above image is taken by the user. It's the verification email send by the admin after the user submitted their registered form. The highlighted link is generated by Django for each registration the link is unique and protected.

Select account to change								
<input type="text"/> Search								
Action:	EMAIL	FIRST NAME	LAST NAME	USERNAME	DATE JOINED	LAST LOGIN	USER TYPE	IS ACTIVE
<input type="checkbox"/>	test@guardian.com	Guardian_f	L_G	test_guardian	April 14, 2022, 6:21 a.m.	April 14, 2022, 6:21 a.m.	admin	<span style="color: green;">✓</span>
<input type="checkbox"/>	guardian@guardian.com	guar	dian	guardian	April 16, 2022, 3:13 p.m.	April 25, 2022, 5:55 a.m.	admin	<span style="color: green;">✓</span>
<input type="checkbox"/>	rrs0053@my.londonmet.ac.uk	Rishi	Shrestha	rrs0053	April 17, 2022, 12:28 a.m.	April 25, 2022, 4:37 a.m.	pharmacist	<span style="color: green;">✓</span>
<input type="checkbox"/>	manishstha338@gmail.com	Manish	Shrestha	manishstha338	April 22, 2022, 7:36 a.m.	April 25, 2022, 5:09 a.m.	counsellor	<span style="color: green;">✓</span>
<input type="checkbox"/>	gurun.123@gma.com	Rishi	Shrestha	gurun.123	April 24, 2022, 4:15 p.m.	April 24, 2022, 4:16 p.m.	pharmacist	<span style="color: green;">✓</span>
<input type="checkbox"/>	gurungmandip111@gmail.com	Mandip	Gurung	gurungmandip111	April 25, 2022, 5:54 a.m.	April 25, 2022, 5:57 a.m.	customer	<span style="color: green;">✓</span>

6 accounts

Figure 131: Screenshot of user activation.

The above image is taken by admin. Here the recently registered user is highlighted the green tick on right shows the email is verified and user can login successfully.

The form is a 'Sign in' page with a light gray background. At the top center is the title 'Sign in' in bold black font. Below it are two input fields: the first for 'Email Address' and the second for 'Password', both with placeholder text. To the right of the password field is a blue 'Forgot password?' link. At the bottom center is a large blue 'Login' button. Below the button is a link for users who don't have an account, reading 'Don't have account? [Sign up](#)'.

*Figure 132: Screenshot of login page.*

The above image shows the login page that is rendered as soon as the verification link is clicked.

#### 4.3.2. Testing Pharmacy Registration

*Table 44: Whitebox testing for pharmacy registration.*

<b>Objective</b>	To test if a new user with a role pharmacist can register their pharmacy.
<b>Action</b>	Fill out all the details required for the registration form and click register button
<b>Expected Result</b>	The admin views the user documents. The pharmacy is recorded into the data base and is added to their respective city.
<b>Actual Result</b>	The pharmacy data was recorded on the database as a pharmacist. The pharmacy was added into the view.
<b>Conclusion</b>	Test Successful.

## Sign up

First name      Last name

Paras      Rai

Email      Who are you?

raiparasmani101@gmail.com      Pharmacist

Create password      Conform password

.....      .....

I agree the [term and condition](#).

[Register](#)

Have an account? [Log In](#)

Figure 133: Screenshot of user registration page.

The above image shows the user registration page. The highlighted box shows the user is registration as a pharmacist.

Action:	EMAIL	FIRST NAME	LAST NAME	USERNAME	DATE JOINED	LAST LOGIN	USER TYPE	IS ACTIVE
<input type="checkbox"/>	test@guardian.com	Guardian_f	L_G	test_guardian	April 14, 2022, 6:21 a.m.	April 14, 2022, 6:21 a.m.	admin	✓
<input type="checkbox"/>	guardian@guardian.com	guar	dian	guardian	April 16, 2022, 3:13 p.m.	April 25, 2022, 5:55 a.m.	admin	✓
<input type="checkbox"/>	rrs0053@my.londonmet.ac.uk	Rishi	Shrestha	rrs0053	April 17, 2022, 12:28 a.m.	April 25, 2022, 4:37 a.m.	pharmacist	✓
<input type="checkbox"/>	manishtha338@gmail.com	Manish	Shrestha	manishtha338	April 22, 2022, 7:36 a.m.	April 25, 2022, 5:09 a.m.	counsellor	✓
<input type="checkbox"/>	gurun.123@gma.com	Rishi	Shrestha	gurun.123	April 24, 2022, 4:15 p.m.	April 24, 2022, 4:16 p.m.	pharmacist	✓
<input type="checkbox"/>	gurungmandip111@gmail.com	Mandip	Gurung	gurungmandip111	April 25, 2022, 5:54 a.m.	April 25, 2022, 5:57 a.m.	customer	✓
<input type="checkbox"/>	raiparasmani101@gmail.com	Paras	Rai	raiparasmani101	April 25, 2022, 6:04 a.m.	April 25, 2022, 6:05 a.m.	pharmacist	✓

7 accounts

Figure 134: Screenshot of admin page viewing users' details.

The above image is taken by admin showing the user is verified.

**Pharmacy Details**

Pharmacy Name

Pharmacy image  No file chosen

Pharmacy Email

Telephone Number

Registration Number

Pharmacy registration Document  No file chosen

Province

District

City

Ward

Tole

From (AM)

Till (PM)

Description

**Submit**

Figure 135: Screenshot of pharmacy registration page.

The above image shows the pharmacy registration page. Here the pharmacist should register their pharmacy.

**Pharmacy Details**

Pharmacy Name	Pharmacy image	
Hamro Swastiya	<input type="button" value="Choose File"/> gettyimages-1170287898-612x612.jpg	
Pharmacy Email	Telephone Number	
swasityahamro@gmail.com	9810000000	
Registration Number	Pharmacy registration Document	
123/122/2342311	<input type="button" value="Choose File"/> Registration Doc Dummy.pdf	
Province	District	City
Province-1		
Ward	Tole	
3	basbari	
From (AM)	Till (PM)	Description
10	4	Hamro swastiya pharmacy is the clinical health science that links medical science with c

**Submit**

Figure 136: Screenshot of pharmacy registration page with filled data.

The above image shows the details of pharmacy being registered.

Select pharmacist detail to change		<input type="button" value="ADD PHARMACIST DETAIL +"/>
Action:	——	<input type="button" value="Go"/> 0 of 2 selected
<input type="checkbox"/>	PHARMACIST DETAIL	
<input checked="" type="checkbox"/>	Hamro Swastiya	
<input type="checkbox"/>	Test Pharmacy nepal	
2 pharmacist details		

Figure 137: Screenshot of admin page viewing pharmacy registered.

The above image is taken by the admin who is viewing the pharmacies which are registered. The highlighted box shows the recently registered pharmacy.

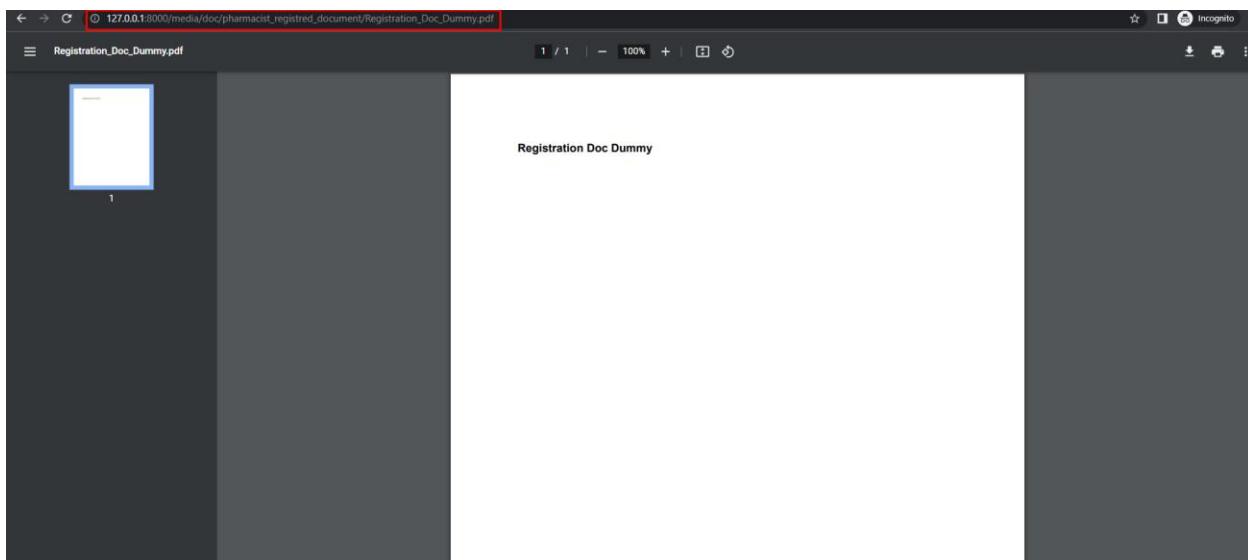


Figure 138: Screenshot of admin viewing pharmacy document.

The above image shows the admin viewing pharmacy registration document.

A screenshot of a website titled 'Pharmacies'. It lists two pharmacies: 'Test Pharmacy nepal' and 'Hamro Swastiya'. The 'Hamro Swastiya' entry is highlighted with a red box. Below the list, there are links for 'Terms and conditions' and 'Copyright © 2020 .All rights reserved by Gardian.'

**Pharmacies**

**Test Pharmacy nepal**  
this is a description of test pharmacy nepal  
[Visit](#)

**Hamro Swastiya**  
Hamro swastiya pharmacy is the clinical health science that links medical science with chemistry and it is charged with the discovery, production, disposal, safe and effective use, and control of medications and drugs.  
[Visit](#)

[Terms and conditions](#)

Copyright © 2020 .All rights reserved by Gardian.

Figure 139: Screenshot of pharmacy page.

The above image shows the recently registered pharmacy viewed by the user.

### 4.3.3. Testing Counsellor Registration

Table 45: Whitebox testing for counsellor registration.

<b>Objective</b>	To test if a new user with a role counsellor can register themselves as counsellor.
<b>Action</b>	Fill out all the details required for the registration form and click register button
<b>Expected Result</b>	The admin views the user documents. The counsellor is recorded into the data base. The counsellor after registration is redirected to its dashboard.
<b>Actual Result</b>	The counsellor data was recorded on the database. The counsellor was added into the view. The counsellor was redirected to dashboard.
<b>Conclusion</b>	Test Successful.

Sign up

First name: Anish      Last name: KC

Email: Kcanish@gmail.com      Who are you? Counsellor

Create password: .....      Conform password: .....

I agree the [term and condition.](#)

**Register**

Have an account? [Log In](#)

Figure 140: Screenshot of user registration page.

The above image shows the user registration page. The highlighted box shows the user is registration as a counsellor.

Select account to change								
<input type="button" value="Search"/> <input type="button" value="ADD ACCOUNT +"/>								
Action:	EMAIL	FIRST NAME	LAST NAME	USERNAME	DATE JOINED	LAST LOGIN	USER TYPE	IS ACTIVE
<input type="checkbox"/>	test@guardian.com	Guardian_f	L_G	test_guardian	April 14, 2022, 6:21 a.m.	April 14, 2022, 6:21 a.m.	admin	<input checked="" type="checkbox"/>
<input type="checkbox"/>	guardian@londonmet.ac.uk	guar	dian	guardian	April 16, 2022, 3:13 p.m.	April 25, 2022, 5:55 a.m.	admin	<input checked="" type="checkbox"/>
<input type="checkbox"/>	rs0053@my.londonmet.ac.uk	Rishi	Shrestha	rrs0053	April 17, 2022, 12:28 a.m.	April 25, 2022, 4:37 a.m.	pharmacist	<input checked="" type="checkbox"/>
<input type="checkbox"/>	manishtha338@gmail.com	Manish	Shrestha	manishtha338	April 22, 2022, 7:36 a.m.	April 25, 2022, 5:09 a.m.	counsellor	<input checked="" type="checkbox"/>
<input type="checkbox"/>	gurun.123@gma.com	Rishi	Shrestha	gurun.123	April 24, 2022, 4:15 p.m.	April 24, 2022, 4:16 p.m.	pharmacist	<input checked="" type="checkbox"/>
<input type="checkbox"/>	gurungmandip111@gmail.com	Mandip	Gurung	gurungmandip111	April 25, 2022, 5:54 a.m.	April 25, 2022, 5:57 a.m.	customer	<input checked="" type="checkbox"/>
<input type="checkbox"/>	raiparasmani101@gmail.com	Paras	Rai	raiparasmani101	April 25, 2022, 6:04 a.m.	April 25, 2022, 6:11 a.m.	pharmacist	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Kcanish@gmail.com	Anish	KC	Kcanish	April 25, 2022, 6:16 a.m.	April 25, 2022, 6:17 a.m.	counsellor	<input checked="" type="checkbox"/>

8 accounts

Figure 141: Screenshot of admin page viewing users' details.

The above image is taken by admin showing the user is verified.

### Counsellor Details

Name	Profile image	
<input type="text" value="counsellor Name"/>	<input type="button" value="Choose File"/> No file chosen	
Counsellor Email	Telephone Number	
<input type="text" value="example@domain.com"/>	<input type="text" value="Phone number"/>	
Registration Number	Verification certificate	
<input type="text" value="Registration number"/>	<input type="button" value="Choose File"/> No file chosen	
Province	District	City
<input type="text" value="-----"/>	<input type="text" value="-----"/>	<input type="text" value="-----"/>
Ward	Tole	
<input type="text" value="name"/>	<input type="text" value="name"/>	
From (AM)	Till (PM)	Description
<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="Write something about your counsellor here"/>
<input type="button" value="Submit"/>		

Figure 142: Screenshot of counsellor registration page.

The above image shows the counsellor registration page. Here the counsellor should register themselves as a valid counsellor.

The screenshot shows a form titled "Counsellor Details". The fields are as follows:

- Name: Teen Counselling Center
- Profile image: Choose File doc1.jpg
- Counsellor Email: teen@counselling.com
- Telephone Number: 9801010101
- Registration Number: 3224/2002/122
- Verification certificate: Choose File Registration Doc Dummy.pdf
- Province: Province-1
- District: Sunsari
- City: Dharan
- Ward: 3
- Tole: Phushe
- From (AM): 9
- Till (PM): 4
- Description: The Teen Counselling center provides services like teen counselling, group counselling, ar

**Submit**

Figure 143: Screenshot of counsellor registration page with filled data.

The above image shows the details of counsellor being registered.

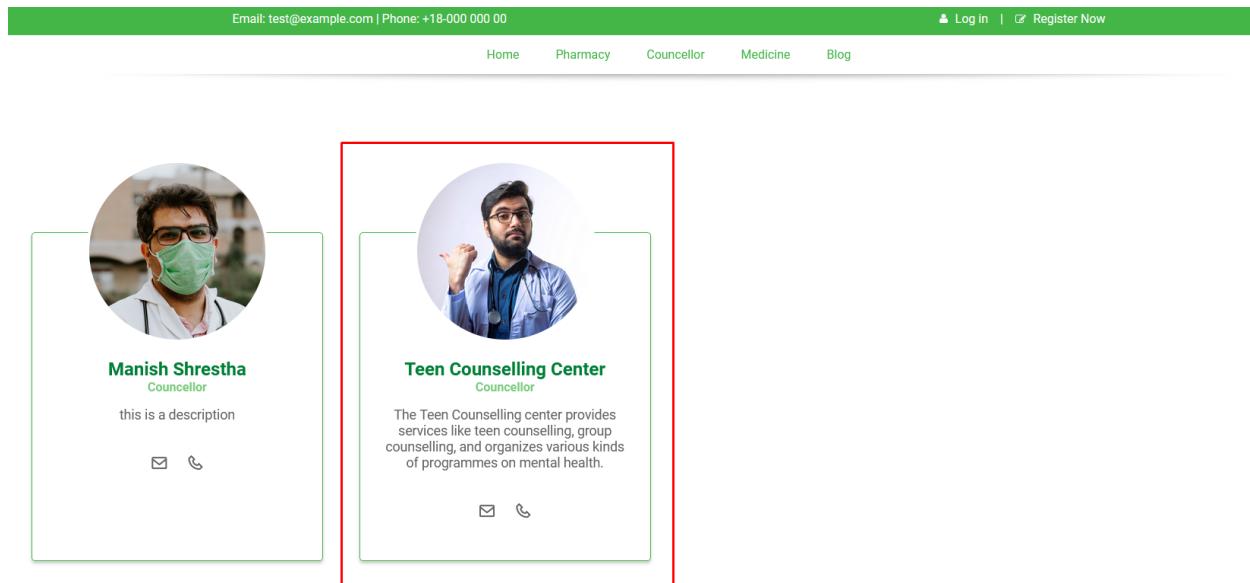


Figure 144: Screenshot of counsellor page.

The above image shows the recently registered counsellor viewed by the user.

#### 4.3.4. Testing User Login

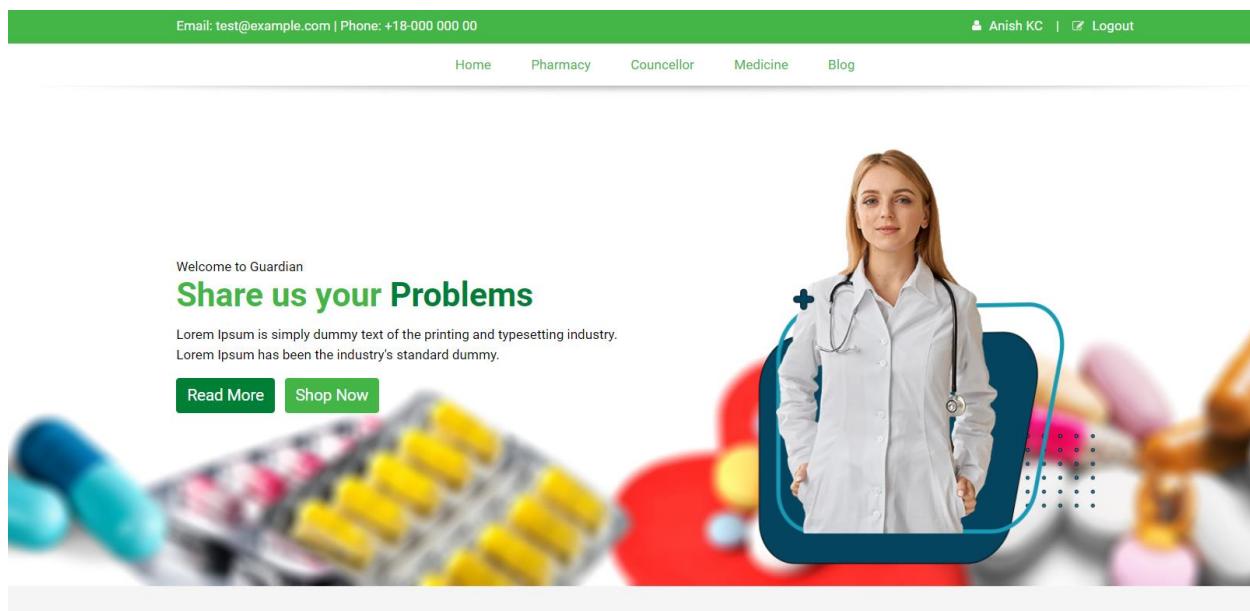
Table 46: Whitebox testing for user login.

<b>Objective</b>	To test if a user can login successfully.
<b>Action</b>	Fill out email and password in form and click the login button.
<b>Expected Result</b>	A user should be logged in and the home page should be displayed.
<b>Actual Result</b>	A user is logged in and the user dashboard is displayed.
<b>Conclusion</b>	Test Successful.

The screenshot shows a 'Sign in' form. It has two input fields: one for email with 'Kcanish@gmail.com' and one for password with '.....'. Below the fields are 'Forgot password?' and 'Login' buttons. At the bottom, it says 'Don't have account? [Sign up](#)'.

Figure 145: Screenshot of login page.

The above image shows the user with email Kcanish@gmail.com logging in into the application.



*Figure 146: Screenshot of home page.*

The above image shows the home page that is rendered after the user successfully logged in into the application.

#### 4.3.5. Testing Forget Password

*Table 47: Whitebox testing for forget password.*

<b>Objective</b>	To test if forget password link is working.
<b>Action</b>	Click on the forget password link and enter valid email address.
<b>Expected Result</b>	The forget password page should open After entering the email address the user should get an email to reset the password.
<b>Actual Result</b>	The forget password page is opened. The user got an email to reset the password.
<b>Conclusion</b>	Test Successful.

The screenshot shows a 'Sign in' form. It has two input fields: one for email ('rrs0053@my.londonmet.ac.uk') and one for password. Below the password field is a 'Forgot password?' link, which is enclosed in a red rectangular box. A large green 'Login' button is at the bottom. At the bottom of the page, there is a link 'Don't have account? [Sign up](#)'.

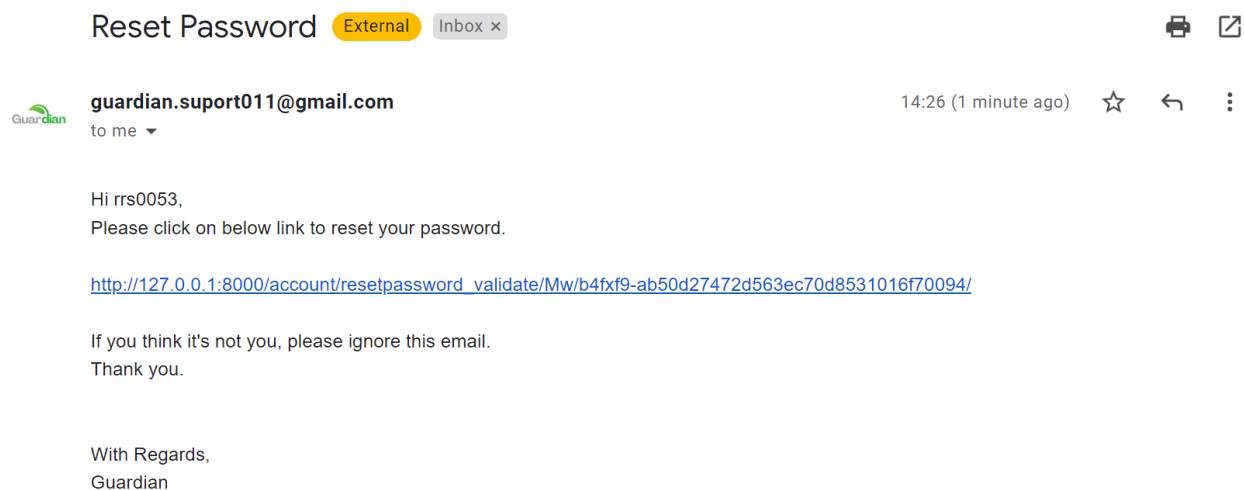
Figure 147: Screenshot of login page.

The above image shows the user with email rrs0053@my.londonmet.ac.uk logging in into the app. The highlighted box shows the user is going to click the forget password link.

The screenshot shows a 'Forgot password' form. It has a single input field for email ('rrs0053@my.londonmet.ac.uk'). To the right of the input field is a 'Log In' link. At the bottom is a large green 'Submit' button.

Figure 148: Screenshot of forget password page.

The above image shows the forget password page redirected after the forget password link is clicked.



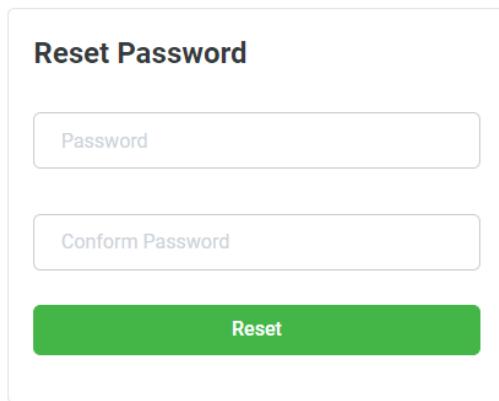
*Figure 149: Screenshot of email verification send to user.*

The above image shows the email verification send to user after submitting their email.

#### 4.3.6. Testing Reset Password

*Table 48: Whitebox testing for reset password.*

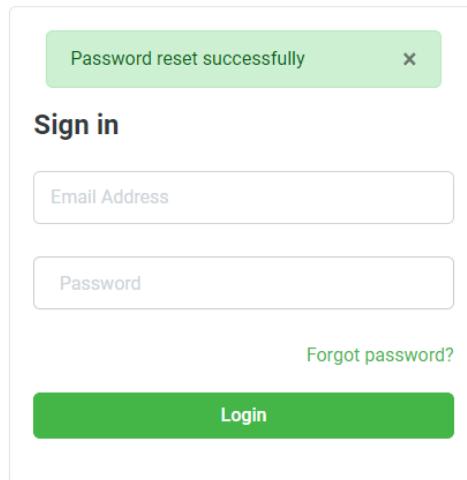
<b>Objective</b>	To test if reset password button is working.
<b>Action</b>	Click on the reset password link and enter the new password.
<b>Expected Result</b>	The user should be redirect to the reset password page. The new password should be updated. User should be able to login with the new password.
<b>Actual Result</b>	The user is redirected to the reset password page. The new password is updated. The user is able to login with the new password.
<b>Conclusion</b>	Test Successful.



The image shows a simple form titled "Reset Password". It contains two input fields: "Password" and "Conform Password", both with placeholder text. Below the fields is a green "Reset" button.

Figure 150: Screenshot of reset password page.

The above image is rendered after the user click the reset password link here the user needs to enter their new password.



The image shows a sign-in form with a green header bar containing the text "Password reset successfully" and a close button. Below the header is the title "Sign in". The form has two input fields: "Email Address" and "Password". To the right of the "Password" field is a "Forgot password?" link. At the bottom is a green "Login" button. Below the form is a link "Don't have account? [Sign up](#)".

Figure 151: Screenshot of sign page.

The above page is rendered after the user successful resets their password.

#### 4.3.7. Testing Admin Login

Table 49: Whitebox testing for admin login.

<b>Objective</b>	To test if an admin can login successfully
<b>Action</b>	Fill out email and password in form and click the login button.
<b>Expected Result</b>	Admin should be logged in and the admin dashboard should be displayed.
<b>Actual Result</b>	An admin is logged in and the admin dashboard is displayed.
<b>Conclusion</b>	Test Successful.

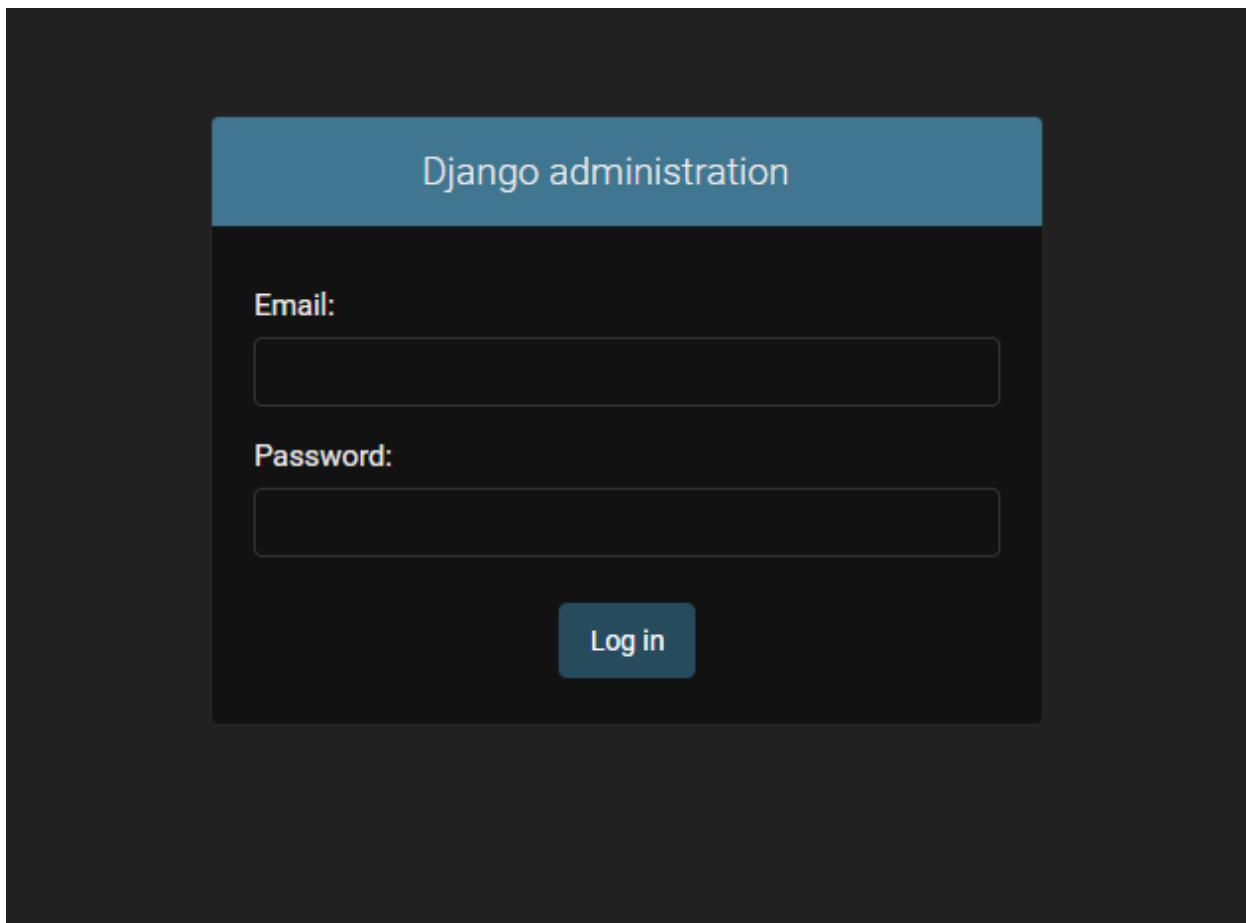


Figure 152: Screenshot of admin login page.

The above image shows the admin log in form.

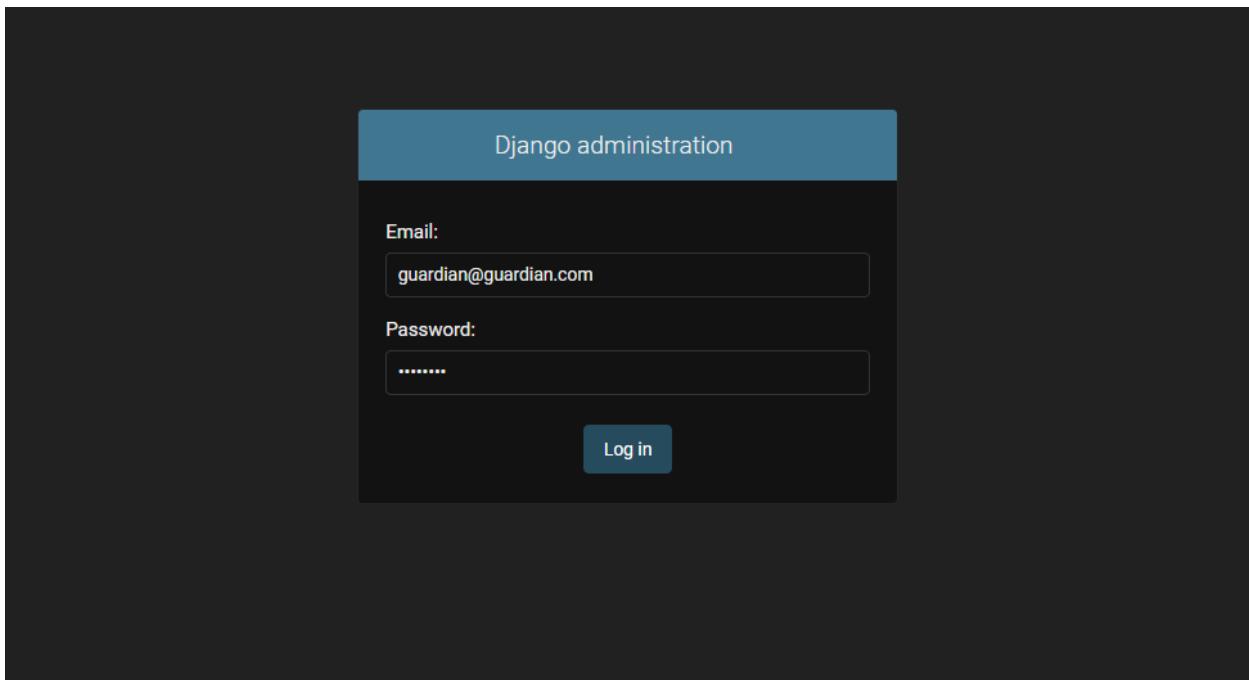


Figure 153: Screenshot of admin login page with filled data.

The above image shows the admin log in form with admin credentials.

A screenshot of the Django administration dashboard. The top navigation bar includes "Django administration", the user's welcome message "WELCOME, GUARDIAN@GUARDIAN.COM", and links for "VIEW SITE / CHANGE PASSWORD / LOG OUT". The main content area is titled "Site administration" and is divided into sections: "ACCOUNT" and "ADDRESS". The "ACCOUNT" section lists "Accounts", "Adresses", "Counsellor details", "Pharmacist details", and "Type\_users", each with "Add" and "Change" buttons. The "ADDRESS" section lists "Citys", "Districts", and "Provinces", also with "Add" and "Change" buttons. To the right, there are two sidebar boxes: "Recent actions" (empty) and "My actions" which lists recent activities such as account creations and blog model entries by users like Kcanish@gmail.com, raiparasmani101@gmail.com, gurungmandip111@gmail.com, Manish Shrestha, and manishstha338@gmail.com.

Figure 154: Screenshot of admin dashboard.

The above image shows the admin dashboard.

#### 4.3.8. Testing City Page

Table 50: Whitebox testing for city page.

<b>Objective</b>	To check if cities are displayed on the city page.
<b>Action</b>	The city page link is clicked.
<b>Expected Result</b>	The city page should be displayed with all the cities present in the database.
<b>Actual Result</b>	The city page is displayed with all the cities.
<b>Conclusion</b>	Test Successful.

The screenshot shows a dark-themed administrative interface for managing cities. At the top, there is a header with the text "Select city to change" and a button labeled "ADD CITY +". Below the header, there is a search bar with the placeholder "Action: -----" and a "Go" button, followed by the text "0 of 11 selected". A table lists 11 cities, each with a checkbox next to its name and its corresponding district listed to the right. The cities and their districts are:

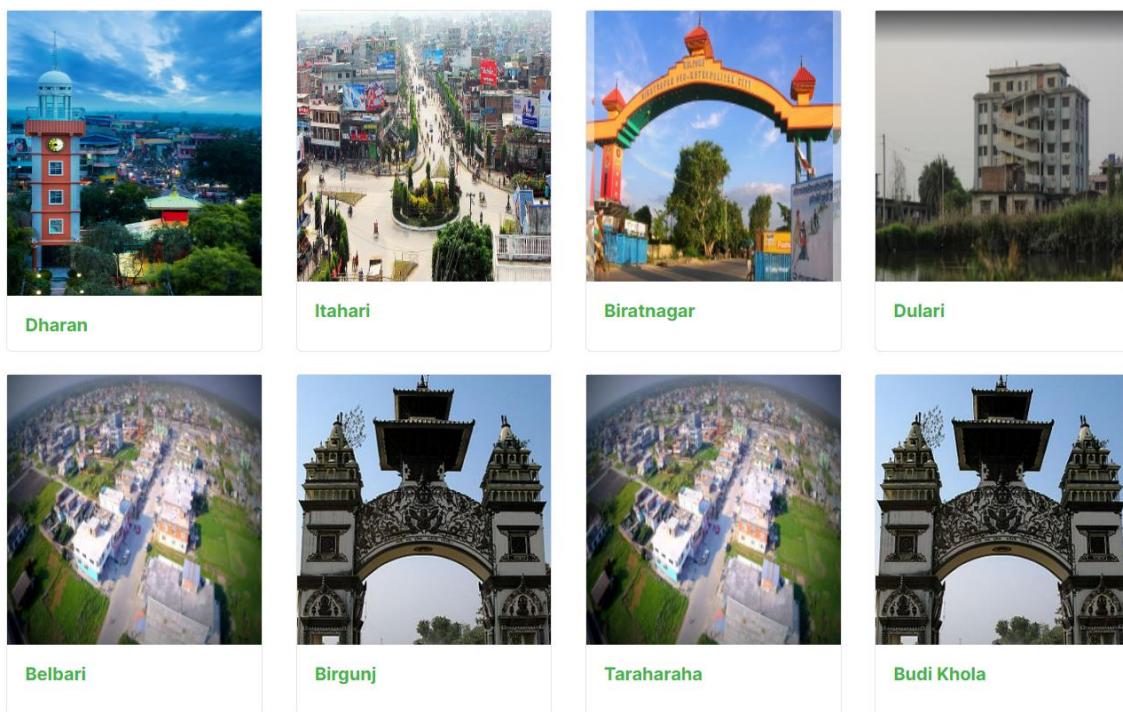
NAME	DISTRICT
Khanar	Sunsari
Duhabi	Sunsari
Railway	Sunsari
Budi Khola	Sunsari
Taraharaha	Sunsari
Birgunj	Parsa
Belbari	Jhapa
Dulari	Morang
Biratnagar	Sunsari
Itahari	Sunsari
Dharan	Sunsari

At the bottom of the list, it says "11 citys".

Figure 155: Screenshot of cities added on admin.

The above image shows all the cities from the admin panel.

## Cities



Previous 1 2 Next

*Figure 156: Screenshot of cities page.*

The above image shows all the cities viewed by the user.

### 4.3.9. Testing Pharmacy Page

*Table 51: Whitebox testing for pharmacy page.*

<b>Objective</b>	To check if pharmacy is displayed on the city page.
<b>Action</b>	The pharmacy page link is clicked.
<b>Expected Result</b>	The pharmacy page should be displayed with all the pharmacies present in the database.
<b>Actual Result</b>	The pharmacy page is displayed with all the pharmacies.
<b>Conclusion</b>	Test Successful.

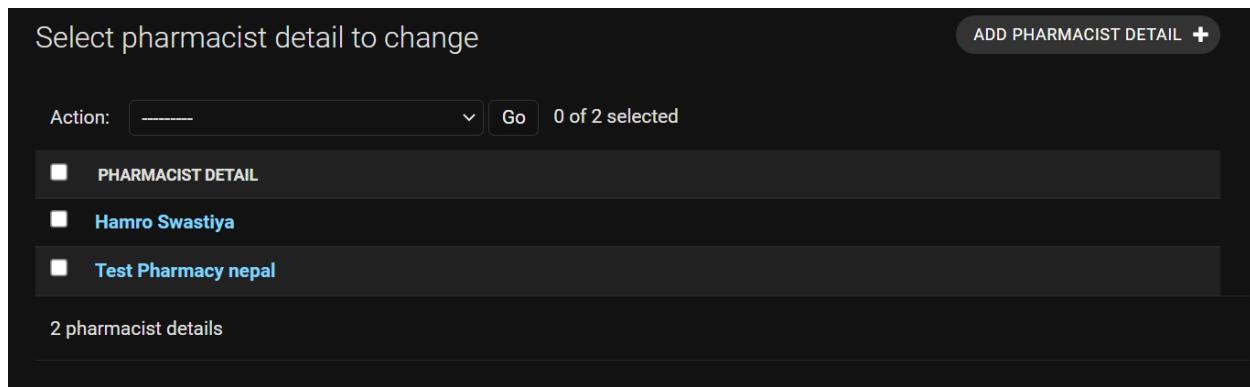


Figure 157: Screenshot of admin page showing registered pharmacy.

The above image shows all the registered pharmacy with admin view.

The screenshot shows a user-facing website with a green header bar containing the text "Email: test@example.com | Phone: +18-000 000 00" and links for "Log In" and "Register". Below the header is a navigation menu with links for "Home", "Pharmacy", "Councillor", "Medicine", and "Blog". The main content area is titled "Pharmacies" and features two cards. The first card is for "Test Pharmacy nepal", showing a thumbnail image of a pharmacy interior, the name, a description ("this is a description of test pharmacy nepal"), and a "Visit" button. The second card is for "Hamro Swastiya", showing a thumbnail image of a pharmacist, the name, a detailed description ("Hamro swastiya pharmacy is the clinical health science that links medical science with chemistry and it is charged with the discovery, production, disposal, safe and effective use, and control of medications and drugs."), and a "Visit" button.

Figure 158: Screenshot of pharmacy viewed by user.

The above image shows the pharmacies on the frontend.

#### 4.3.10. Testing View user details

Table 52: Whitebox testing for viewing user details.

<b>Objective</b>	To check if admin can view user details.
<b>Action</b>	Login to admin panel and click on account model form the side navigation.
<b>Expected Result</b>	All the users with their details should be displayed.
<b>Actual Result</b>	All the users with their details are displayed.
<b>Conclusion</b>	Test Successful.

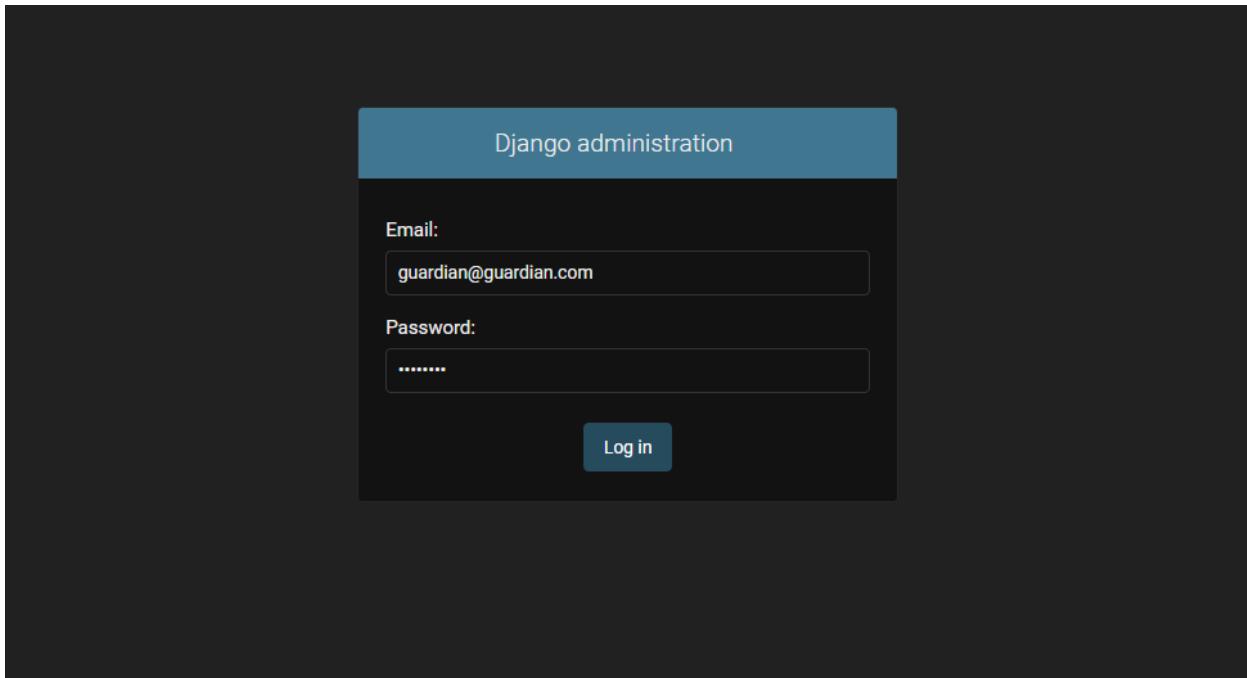


Figure 159: Screenshot of admin login page.

The above image shows the user logging as admin.

<input type="checkbox"/>	EMAIL	FIRST NAME	LAST NAME	USERNAME	DATE JOINED	LAST LOGIN	USER TYPE	IS ACTIVE
<input checked="" type="checkbox"/>	test@gaurdian.com	Guardian_f	L_G	test_guardian	April 14, 2022, 6:21 a.m.	April 14, 2022, 6:21 a.m.	admin	✓
<input checked="" type="checkbox"/>	guardian@gaurdian.com	guar	dian	guardian	April 16, 2022, 3:13 p.m.	April 25, 2022, 8:45 a.m.	admin	✓
<input checked="" type="checkbox"/>	rrs0053@my.londonmet.ac.uk	Rishi	Shrestha	rrs0053	April 17, 2022, 12:28 a.m.	April 25, 2022, 9:18 a.m.	pharmacist	✓
<input checked="" type="checkbox"/>	manishtha338@gmail.com	Manish	Shrestha	manishtha338	April 22, 2022, 7:36 a.m.	April 25, 2022, 5:09 a.m.	counsellor	✓
<input checked="" type="checkbox"/>	gurun.123@gmail.com	Rishi	Shrestha	gurun.123	April 24, 2022, 4:15 p.m.	April 24, 2022, 4:16 p.m.	pharmacist	✓
<input checked="" type="checkbox"/>	gurungmandip111@gmail.com	Mandip	Gurung	gurungmandip111	April 25, 2022, 5:54 a.m.	April 25, 2022, 5:57 a.m.	customer	✓
<input checked="" type="checkbox"/>	raiparasmani101@gmail.com	Paras	Rai	raiparasmani101	April 25, 2022, 6:04 a.m.	April 25, 2022, 6:11 a.m.	pharmacist	✓
<input checked="" type="checkbox"/>	Keanish@gmail.com	Anish	KC	Kcanish	April 25, 2022, 6:16 a.m.	April 25, 2022, 8:39 a.m.	counsellor	✓

Figure 160: Screenshot of admin viewing user details.

The above image shows all the user account credentials.

The screenshot shows a user profile editing page. At the top, it says "Change account" and displays the email "gurungmandip111@gmail.com". A "HISTORY" button is in the top right corner. Below the email, a note states: "algorithm: pbkdf2\_sha256 Iterations: 260000 salt: 2efUqJ\*\*\*\*\* hash: QW Gh8\*\*\*\*\*". It also notes: "Raw passwords are not stored, so there is no way to see this user's password, but you can change the password using this form." The user's first name is listed as "Mandip" and last name as "Gurung". The username is "gurungmandip111" and the email is "gurungmandip111@gmail.com". The user type is "customer". There are several checkboxes for roles: "Is active" (checked), "Is admin" (unchecked), "Is staff" (unchecked), and "Is superuser" (unchecked). At the bottom left is a red "Delete" button. On the right are three buttons: "Save and add another", "Save and continue editing", and a large blue "SAVE" button.

Figure 161: Screenshot of admin updating user details.

The above image shows the admin viewing the detail of a user with first name Mandip.

#### 4.3.11. Testing view user document

Table 53: Whitebox testing for viewing user document.

<b>Objective</b>	To check if admin can view document of users.
<b>Action</b>	Login to admin panel and click on counsellor detail model or pharmacist detail form the side navigation.
<b>Expected Result</b>	All the registered counsellor/pharmacist should be displayed, and the admin should be able to check their details and view their documents.
<b>Actual Result</b>	All the registered counsellor/pharmacist is displayed, and the admin is able to check their details and view their documents.
<b>Conclusion</b>	Test Successful.

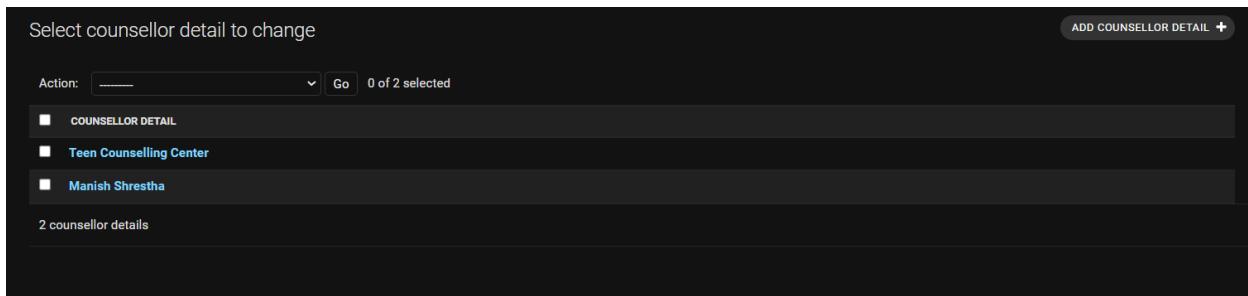


Figure 162: Screenshot of registered counsellor.

The above image shows the registered counsellors.

Change counsellor detail

Teen Counselling Center

Counsellor name: Teen Counselling Center

Profile image: Currently: profile/counsellor\_profile\_image/doc1.jpg  
Change: Choose File No file chosen

Counsellor email: teen@counselling.com

Phone no: 9801010101

Registration no: 3224/2002/122

Registered doc: Currently: doc/counsellor\_registered\_document/Registration\_Doc\_Dummy.pdf  
Change: Choose File No file chosen

Work start: 9

Work end: 4

Description: The Teen Counselling center provides service

City: Dharan

User id: Kcanish@gmail.com

Delete Save and add another Save and continue editing SAVE

Figure 163: Screenshot of admin checking counsellor detail.

The above image shows the details of counselling centre named Teen Counselling Centre. The highlighted text shows the admin is going to view the file.

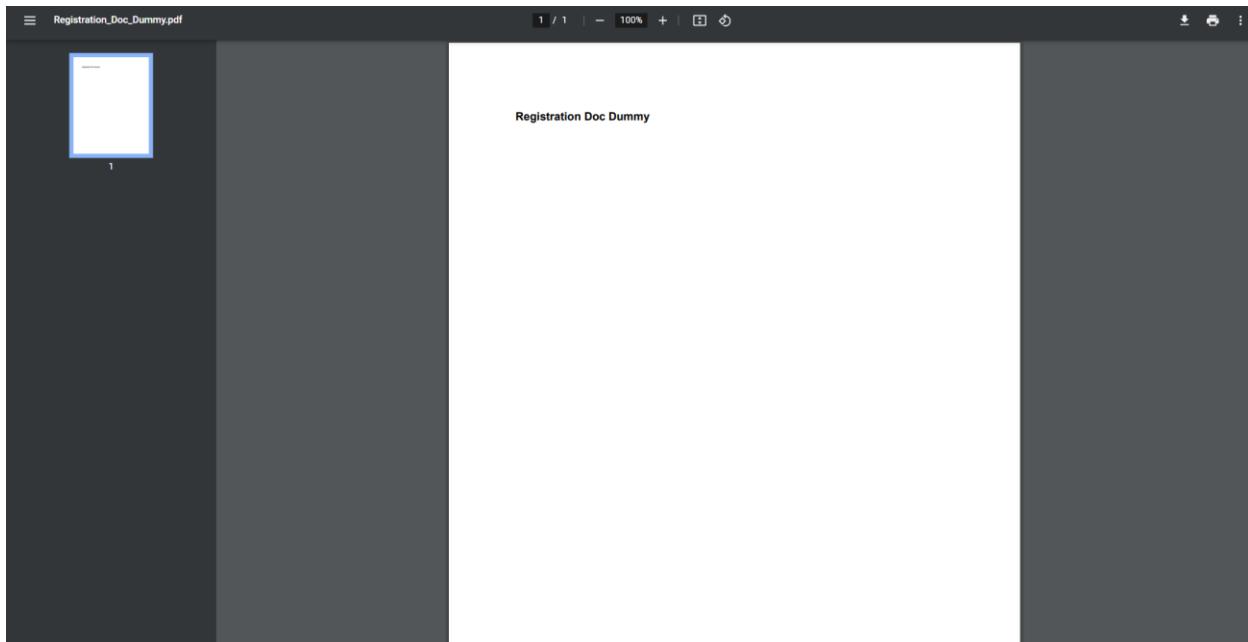


Figure 164: Screenshot of admin checking counsellor document.

The above image shows the admin viewing the registration document.

#### 4.3.12. Testing approve user

Table 54: Whitebox testing for user approval.

<b>Objective</b>	To check if the admin can approve user.
<b>Action</b>	Login to admin panel and click on account model from the side navigation.
<b>Expected Result</b>	Admin should be able to approve user. User should be able to login without verifying their Gmail.
<b>Actual Result</b>	Admin was able to approve user. The user was able to login without verifying Gmail account.
<b>Conclusion</b>	Test Successful.

Select account to change								
<input type="button" value="ADD ACCOUNT +"/> <input type="text"/> <input type="button" value="Search"/>								
Action:	EMAIL	FIRST NAME	LAST NAME	USERNAME	DATE JOINED	LAST LOGIN	USER TYPE	IS ACTIVE
<input type="checkbox"/>	test@gardian.com	Guardian_f	L_G	test_guardian	April 14, 2022, 6:21 a.m.	April 14, 2022, 6:21 a.m.	admin	<input checked="" type="checkbox"/>
<input type="checkbox"/>	guardian@gardian.com	guar	dian	guardian	April 16, 2022, 3:13 p.m.	April 25, 2022, 8:45 a.m.	admin	<input checked="" type="checkbox"/>
<input type="checkbox"/>	rrs0053@my.londonmet.ac.uk	Rishi	Shrestha	rrs0053	April 17, 2022, 12:28 a.m.	April 25, 2022, 9:18 a.m.	pharmacist	<input checked="" type="checkbox"/>
<input type="checkbox"/>	manishstha338@gmail.com	Manish	Shrestha	manishstha338	April 22, 2022, 7:36 a.m.	April 25, 2022, 5:09 a.m.	counsellor	<input checked="" type="checkbox"/>
<input type="checkbox"/>	gurun.123@gma.com	Rishi	Shrestha	gurun.123	April 24, 2022, 4:15 p.m.	April 24, 2022, 4:16 p.m.	pharmacist	<input checked="" type="checkbox"/>
<input type="checkbox"/>	gurungmandip111@gmail.com	Mandip	Gurung	gurungmandip111	April 25, 2022, 5:54 a.m.	April 25, 2022, 5:57 a.m.	customer	<input checked="" type="checkbox"/>
<input type="checkbox"/>	raiparasmani101@gmail.com	Paras	Rai	raiparasmani101	April 25, 2022, 6:04 a.m.	April 25, 2022, 6:11 a.m.	pharmacist	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Kcanish@gmail.com	Anish	KC	Kcanish	April 25, 2022, 6:16 a.m.	April 25, 2022, 8:39 a.m.	counsellor	<input checked="" type="checkbox"/>
<input type="checkbox"/>	mangu@gmail.com	Manju	Shrestha	mangu	April 25, 2022, 10:27 a.m.	April 25, 2022, 10:27 a.m.	customer	<input type="checkbox"/>

9 accounts

Figure 165: Screenshot of admin viewing user details.

The above image shows the admin viewing the user accounts. The highlighted box shows the unverified user.

Change account

**mangu@gmail.com**

HISTORY

Password: algorithm: pbkdf2\_sha256 iterations: 260000 salt: N37LwY\*\*\*\*\* hash: ryHcv1\*\*\*\*\*  
Raw passwords are not stored, so there is no way to see this user's password, but you can change the password using this form.

First name:	Manju
Last name:	Shrestha
Username:	mangu
Email:	mangu@gmail.com
User type:	customer
<input type="checkbox"/> Is active	<input checked="" type="checkbox"/>
<input type="checkbox"/> Is admin	<input type="checkbox"/>
<input type="checkbox"/> Is staff	<input type="checkbox"/>
<input type="checkbox"/> Is superuser	<input type="checkbox"/>

**Delete** **Save and add another** **Save and continue editing** **SAVE**

Figure 166: Screenshot of admin approving user.

The above image shows the admin viewing the details of user named manju. The highlighted box shows the user is going to check the check box.

Change account  
mangu@gmail.com

algorithm: pbkdf2\_sha256 iterations: 260000 salt: N37LwY\*\*\*\*\* hash: ryHcv1\*\*\*\*\*

Raw passwords are not stored, so there is no way to see this user's password, but you can change the password using [this form](#).

First name:	Manju
Last name:	Shrestha
Username:	mangu
Email:	mangu@gmail.com
User type:	customer

Is active

Is admin

Is staff

Is superuser

[Delete](#) [Save and add another](#) [Save and continue editing](#) [SAVE](#)

Figure 167: Screenshot of admin approving user.

The above image shows the admin viewing the details of user named manju. The highlighted box shows the user has check the check box.

The account \*mangu@gmail.com\* was changed successfully.

Select account to change

Action:   0 of 9 selected

	EMAIL	FIRST NAME	LAST NAME	USERNAME	DATE JOINED	LAST LOGIN	USER TYPE	IS ACTIVE
<input type="checkbox"/>	test@guardian.com	Guardian_f	L_G	test_guardian	April 14, 2022, 6:21 a.m.	April 14, 2022, 6:21 a.m.	admin	<input checked="" type="checkbox"/>
<input type="checkbox"/>	guardian@guardian.com	guar	dian	guardian	April 16, 2022, 3:13 p.m.	April 25, 2022, 8:45 a.m.	admin	<input checked="" type="checkbox"/>
<input type="checkbox"/>	rrs0053@my.londonmet.ac.uk	Rishi	Shrestha	rrs0053	April 17, 2022, 12:28 a.m.	April 25, 2022, 9:18 a.m.	pharmacist	<input checked="" type="checkbox"/>
<input type="checkbox"/>	manishstha338@gmail.com	Manish	Shrestha	manishstha338	April 22, 2022, 7:36 a.m.	April 25, 2022, 5:09 a.m.	counsellor	<input checked="" type="checkbox"/>
<input type="checkbox"/>	gurun.123@gma.com	Rishi	Shrestha	gurun.123	April 24, 2022, 4:15 p.m.	April 24, 2022, 4:16 p.m.	pharmacist	<input checked="" type="checkbox"/>
<input type="checkbox"/>	gurungmandip111@gmail.com	Mandip	Gurung	gurungmandip111	April 25, 2022, 5:54 a.m.	April 25, 2022, 5:57 a.m.	customer	<input checked="" type="checkbox"/>
<input type="checkbox"/>	raiparasmani101@gmail.com	Paras	Rai	raiparasmani101	April 25, 2022, 6:04 a.m.	April 25, 2022, 6:11 a.m.	pharmacist	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Kcanish@gmail.com	Anish	KC	Kcanish	April 25, 2022, 6:16 a.m.	April 25, 2022, 8:39 a.m.	counsellor	<input checked="" type="checkbox"/>
<input type="checkbox"/>	<b>mangu@gmail.com</b>	Manju	Shrestha	<b>mangu</b>	April 25, 2022, 10:27 a.m.	April 25, 2022, 10:28 a.m.	customer	<input checked="" type="checkbox"/>

9 accounts

Figure 168: Screenshot of success after approving user.

The above image shows the admin updating the user account successfully.

#### 4.3.13. Testing add Medicine

Table 55: Whitebox testing for add medicine.

<b>Objective</b>	To check if the added medicine is stored in database.
<b>Action</b>	Login as a pharmacist and fill out add product form. Login as admin and check medicine model. Login as user and view store page.
<b>Expected Result</b>	The added product should be able to be viewed by the pharmacist on view product page, user on store page and admin on the medicine model.
<b>Actual Result</b>	The added product was displayed on view added product page, on store page as well as by the admin on model.
<b>Conclusion</b>	Test Successful.

Figure 169: Screenshot of pharmacist adding product.

The above image shows the pharmacy adding product/medicine named Cough Syrup.

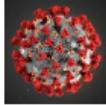
PRODUCT	CATAGORY	PER-PRICE	UPDATE PRODUCT	REMOVE PRODUCT
 Paracetamol Alternative Name: name1	Liquid	Rs: 123	<button>Update</button>	<button>Remove</button>
 Paracetamol2 Alternative Name: name1	Liquid	Rs: 123	<button>Update</button>	<button>Remove</button>
 Cough Syrup Alternative Name: name1	Liquid	Rs: 150	<button>Update</button>	<button>Remove</button>

Figure 170: Screenshot of pharmacist viewing recently added product.

The above image shows the recently added medicine.

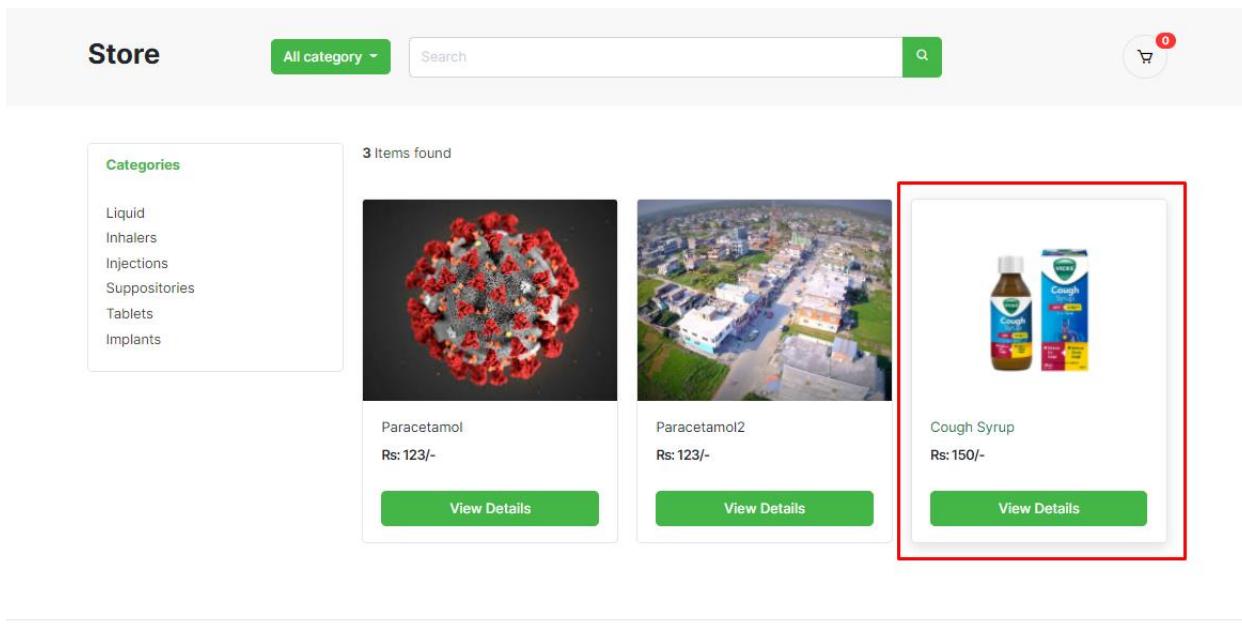
Change add\_product

Cough Syrup

Product name:	Cough Syrup	HISTORY
Slug:	cough-syrup	
Image:	Currently: photos/product_images/Vicks_AU_Cough_2in1_Syrup_front.webp	
Cost:	150	
Stock:	10	
Dose child:	15ml	
Dose adult:	30ml	
Category:	Liquid	 
Contraindiction:	Lorem Ipsum has been the industry's standard.	
Indication:	Lorem Ipsum has been the industry's standard.	
Special precautions:	Lorem Ipsum has been the industry's standard.	
Adverse effect:	Lorem Ipsum has been the industry's standard.	
Pharmacy name id:	Test Pharmacy nepal	 

Figure 171: Screenshot of admin viewing product.

The above image shows the admin viewing the product.



[Terms and conditions](#)

Copyright © 2020 .All rights reserved by Gardian.

Figure 172: Screenshot of user viewing product.

The above image shows the user viewing the product.

#### 4.3.14. Testing add Blog

Table 56: Whitebox testing for add blog.

<b>Objective</b>	To check if the added blog is stored in database.
<b>Action</b>	Login as a counsellor and fill out add blog form. Login as admin and check blog model. Login as user and view blog page.
<b>Expected Result</b>	The added blog should be able to be viewed by the counsellor on view blog page, user on blog page and admin on the blog model.
<b>Actual Result</b>	The added blog was displayed on view added blog page, on store page as well as by the admin on blog model.
<b>Conclusion</b>	Test Successful.

### Add Blog

Title:

The Healthiest People in the World Don't Go to the Gym

Content:

If you want to be as healthy as possible, there are no treadmills or weight machines required.

Don't just take my word for it — look to the longest-lived people in the world for proof.

People in the world's Blue Zones— the places around the world with the highest life expectancy — don't pump iron, run marathons or join gyms.

Instead, they live in environments that constantly nudge them into moving without even thinking about it. This means that they grow gardens, walk throughout the day, and minimize mechanical conveniences for house and yard work.

Powered by Froala Characters : 551

Image:

Choose File No file chosen

**Add Blog**

**Clear**

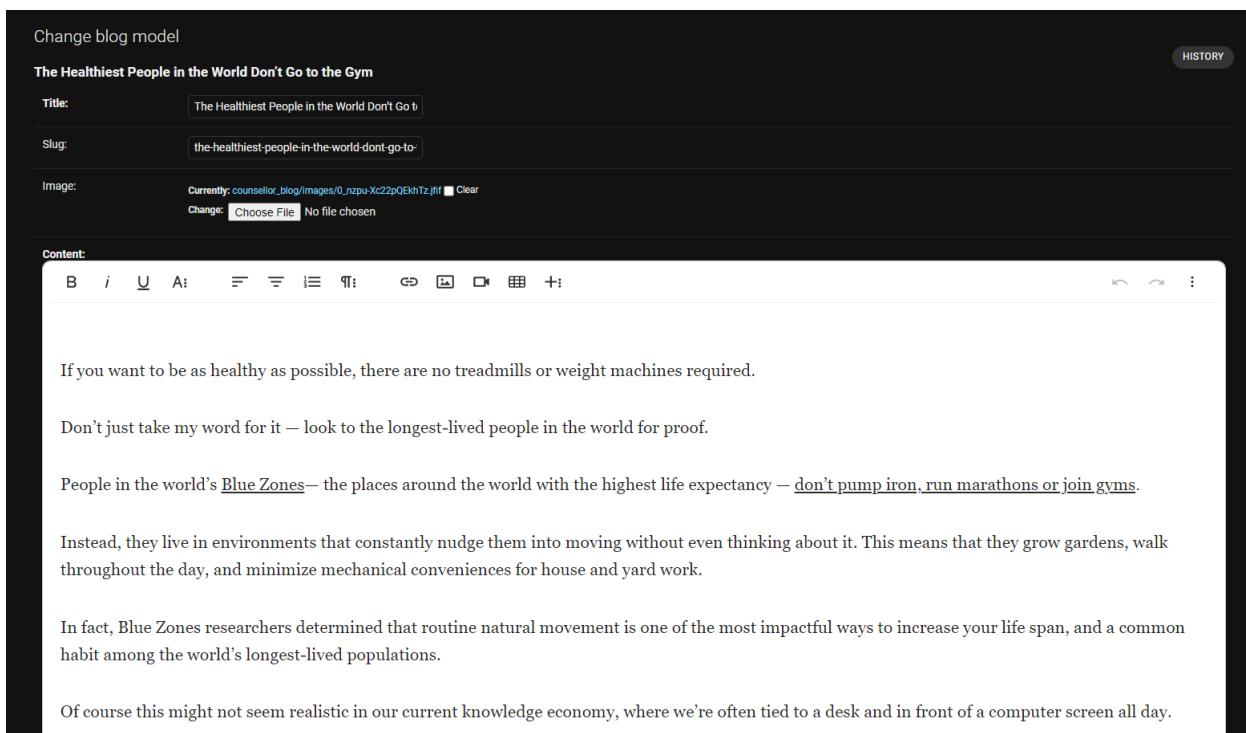
Figure 173: Screenshot of counsellor adding blog.

The above image shows the pharmacy adding blog named 'The Healthiest People in the world don't go to the Gym'.

TITLE	DATE	UPDATE	REMOVE
 The Healthiest People in the World Don't Go to the Gym Alternative Name: name1	April 25, 2022, 10:35 a.m.	<button>Update</button>	<button>Remove</button>

Figure 174: Screenshot of recently added blog viewed as a counsellor.

The above image shows the recently added blog.



Change blog model

**The Healthiest People in the World Don't Go to the Gym**

Title: The Healthiest People in the World Don't Go to the Gym

Slug: the-healthiest-people-in-the-world-don-t-go-to-

Image: Currently: counsellor\_blog/images/0\_nzpu-Xc22pQEkh1z.jfif Clear Change Choose File No file chosen

Content:

If you want to be as healthy as possible, there are no treadmills or weight machines required.

Don't just take my word for it — look to the longest-lived people in the world for proof.

People in the world's Blue Zones— the places around the world with the highest life expectancy — don't pump iron, run marathons or join gyms.

Instead, they live in environments that constantly nudge them into moving without even thinking about it. This means that they grow gardens, walk throughout the day, and minimize mechanical conveniences for house and yard work.

In fact, Blue Zones researchers determined that routine natural movement is one of the most impactful ways to increase your life span, and a common habit among the world's longest-lived populations.

Of course this might not seem realistic in our current knowledge economy, where we're often tied to a desk and in front of a computer screen all day.

Figure 175: Screenshot of admin viewing blog.

The above image shows the admin viewing the blog.

## All Blogs



**Manish Shrestha** April 25, 2022, 5 a.m.

[Read](#)



**Manish Shrestha** April 25, 2022, 5:04 a.m.

[Read](#)



**Teen Counselling Center** April 25, 2022, 10:35 a.m.

[Read](#)

[1](#) [2](#) [3](#) [4](#) [→](#)

Figure 176: Screenshot of user viewing blog.

The above image shows the user viewing the blog.

### 4.3.15. Testing Payment

Table 57: Whitebox testing for payment.

<b>Objective</b>	To check if the PayPal payment is successful.
<b>Action</b>	Navigate to the checkout page and click on PayPal payment method.
<b>Expected Result</b>	A PayPal model should pop up. User should login into PayPal to pay the amount. User should be redirected to home page and success message should be displayed.
<b>Actual Result</b>	A PayPal model is popped up. User was redirected to home page.
<b>Conclusion</b>	Test Successful.

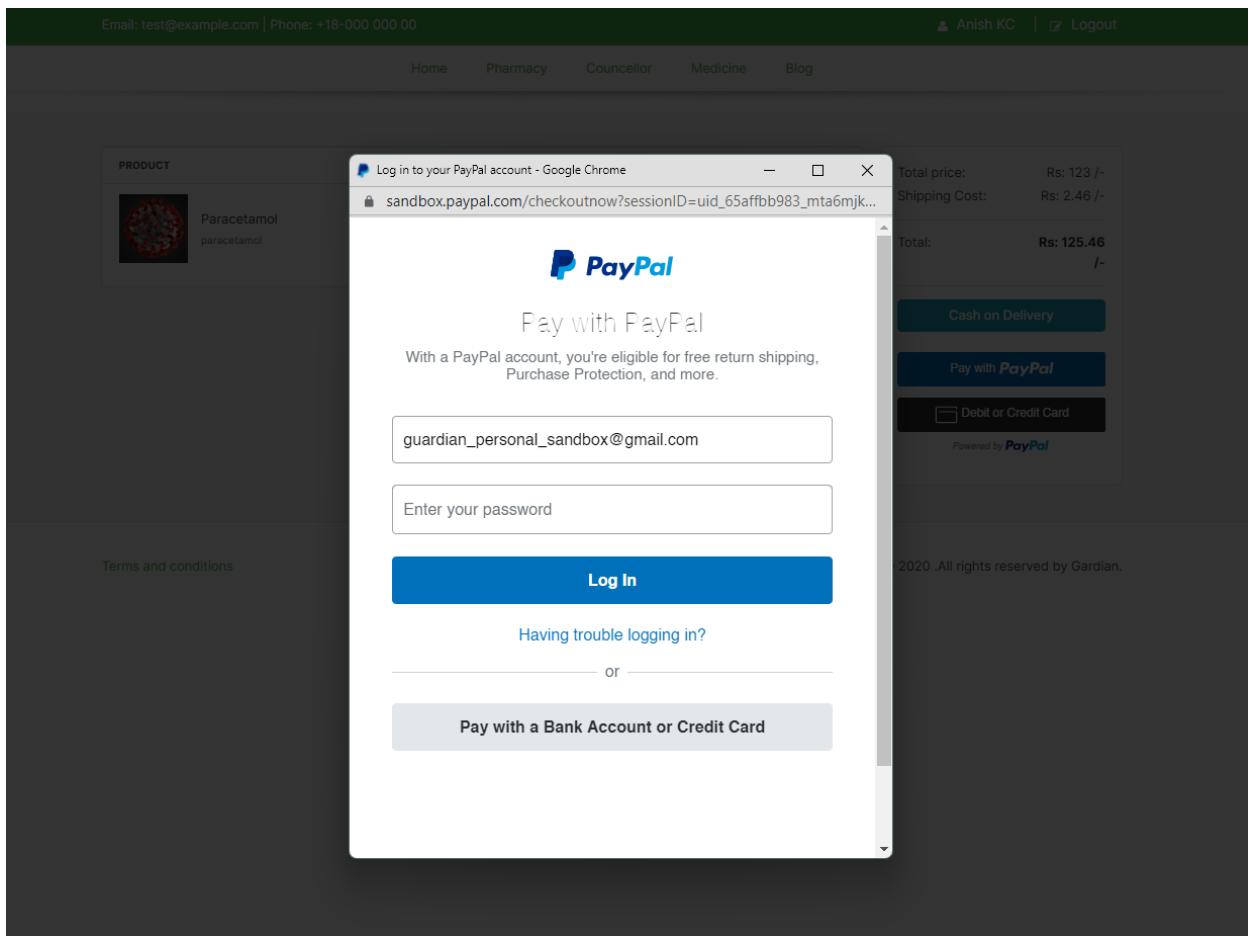


Figure 177: Screenshot of PayPal login.

The above image shows the PayPal Login popped up after clicking pay with PayPal Button.

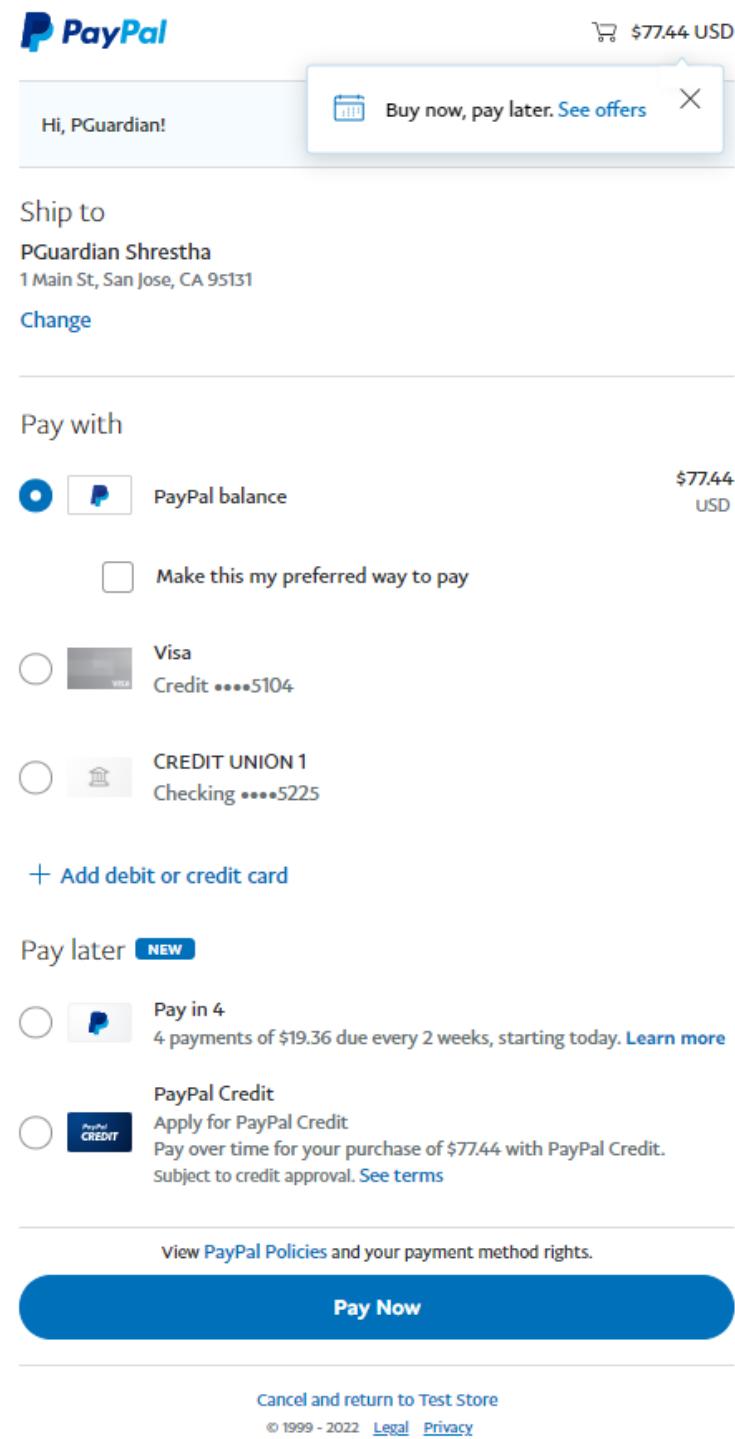


Figure 178: Screenshot of payment made with PayPal.

The above image shows the user paying via PayPal.

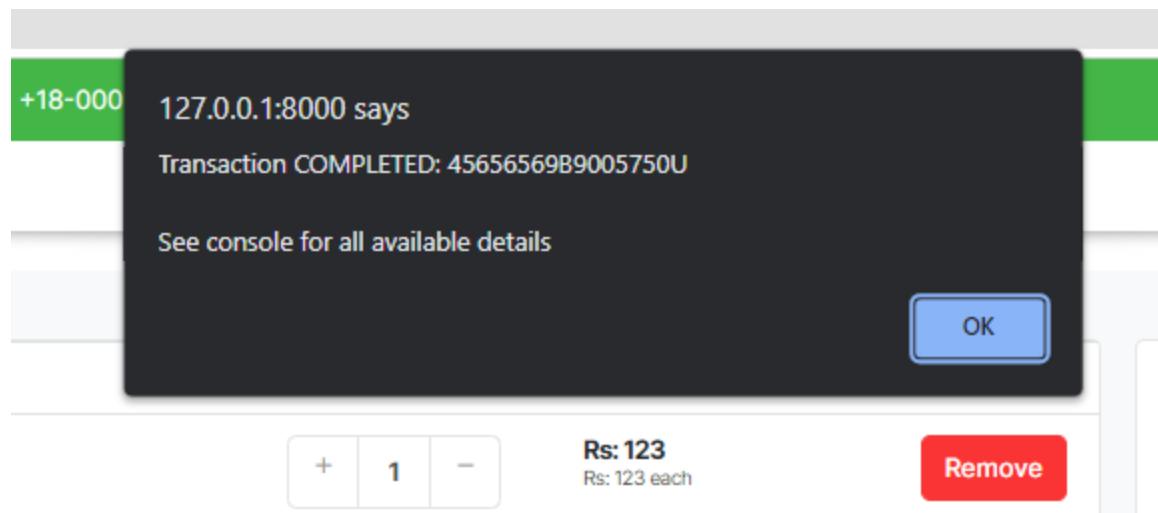


Figure 179: Screenshot of message alert after successful payment.

The above image shows the successful payment.

#### 4.3.16. Validation Testing

##### 4.3.16.1. Login validation testing

Table 58: Validation testing for user login.

<b>Objective</b>	To test if an error message is displayed when user tries to login unverified or login with bad credential.
<b>Action</b>	Fill up the login form with random email and password and click login button.
<b>Expected Result</b>	The error message should be displayed.
<b>Actual Result</b>	The error message was displayed.
<b>Conclusion</b>	Test Successful.

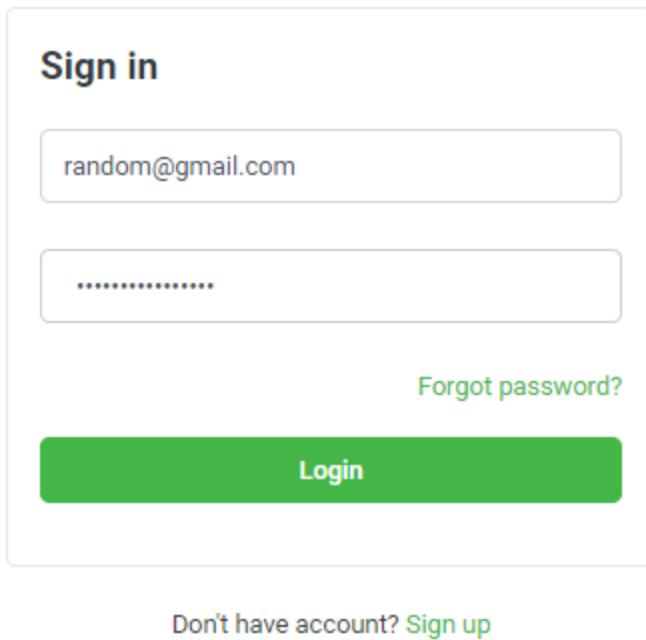


Figure 180: Screenshot of login page with different credential.

The above image shows user logging in with an email address random@gmail.com.

The screenshot shows a login interface with a pink error message box at the top containing the text "Error: Invalid email or password" and a close button. Below the message box is the title "Sign in". There are two input fields: "Email Address" and "Password". To the right of the "Forgot password?" link is a green "Login" button. At the bottom left, there is a link "Don't have account? Sign up".

*Figure 181: Screenshot of unsuccessful login.*

The above image indicates the entered credentials are invalid.

#### 4.3.16.2. Register validation testing

*Table 59: Validation testing for user registration.*

<b>Objective</b>	To test if the error message is displayed when a user tries to log in with an already registered email.
<b>Action</b>	Fill up the registration details and click the “Register” button
<b>Expected Result</b>	The error message should be displayed.
<b>Actual Result</b>	The error message was displayed.
<b>Conclusion</b>	Test Successful.

## Sign up

First name                      Last name

Recreate                        Shrestha

Email                            Who are you?

rrs0053@my.londonmet.ac.uk    Customer

Create password                Conform password

Password                        Confirm Password

• Account with this Email already exists.

I agree the [term and condition](#).

[Register](#)

Have an account? [Log In](#)

Figure 182: Screenshot of registration user with similar email address.

The above image shows the user registration page where the already registered email is used to register a user.

## 4.4. Integration Testing

### 4.4.1. Testing Create Operation of the medicine.

*Table 60: Integration testing to create operation of medicine.*

<b>Objective</b>	To test if pharmacist can add a new medicine.
<b>Action</b>	Login with pharmacist account and navigate to its dashboard to add product.
<b>Expected Result</b>	Pharmacist should be able to add a new medicine. The medicine should be stored in a database. The new medicine should be displayed to the admin. The new medicine should be displayed to the user.
<b>Actual Result</b>	Pharmacist was able to add a new product. The medicine was stored in a database. The new medicine was displayed to the admin. The new medicine was displayed to the user
<b>Conclusion</b>	Test Successful.

**Add Product**

Name	Alternative Name	
Cough Syrup	Khoki	
Cost	Stock	
150	10	
Image		
<input type="button" value="Choose File"/> Vicks_AU_Cough_2in1_Syrup_front.webp		
Category*	Usual Dose for child	Usual Dose for Adult
Liquid	15ml	30ml
Contraindiction	Indication	
Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled		
Special Precautions	Adverse Effect	
Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled		

Figure 183: Screenshot of adding product by pharmacist.

Here the pharmacist is adding product.

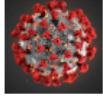
PRODUCT	CATEGORY	PER-PRICE	UPDATE PRODUCT	REMOVE PRODUCT
 Paracetamol Alternative Name: name1	Liquid	Rs: 123	<input type="button" value="Update"/>	<input type="button" value="Remove"/>
 Paracetamol2 Alternative Name: name1	Liquid	Rs: 123	<input type="button" value="Update"/>	<input type="button" value="Remove"/>
 Cough Syrup Alternative Name: name1	Liquid	Rs: 150	<input type="button" value="Update"/>	<input type="button" value="Remove"/>

Figure 184: Screenshot of products added by pharmacist.

The product added on previous image is highlighted here.

The screenshot shows a form titled "Change add\_product" for a "Cough Syrup" product. The form includes fields for Product name (Cough Syrup), Slug (cough-syrup), Image (currently photos/product\_images/Vicks\_AU\_Cough\_2in1\_Syrup\_front.webp, Change, Choose File, No file chosen), Cost (150), Stock (10), Dose child (15ml), Dose adult (30ml), Category (Liquid), Contraindiction (Lorem Ipsum has been the industry's standard), Indication (Lorem Ipsum has been the industry's standard), Special precautions (Lorem Ipsum has been the industry's standard), Adverse effect (Lorem Ipsum has been the industry's standard), and Pharmacy name id (Test Pharmacy nepal). A "HISTORY" button is visible in the top right corner.

Figure 185: Screenshot of admin viewing product.

Here the admin is viewing the added product.

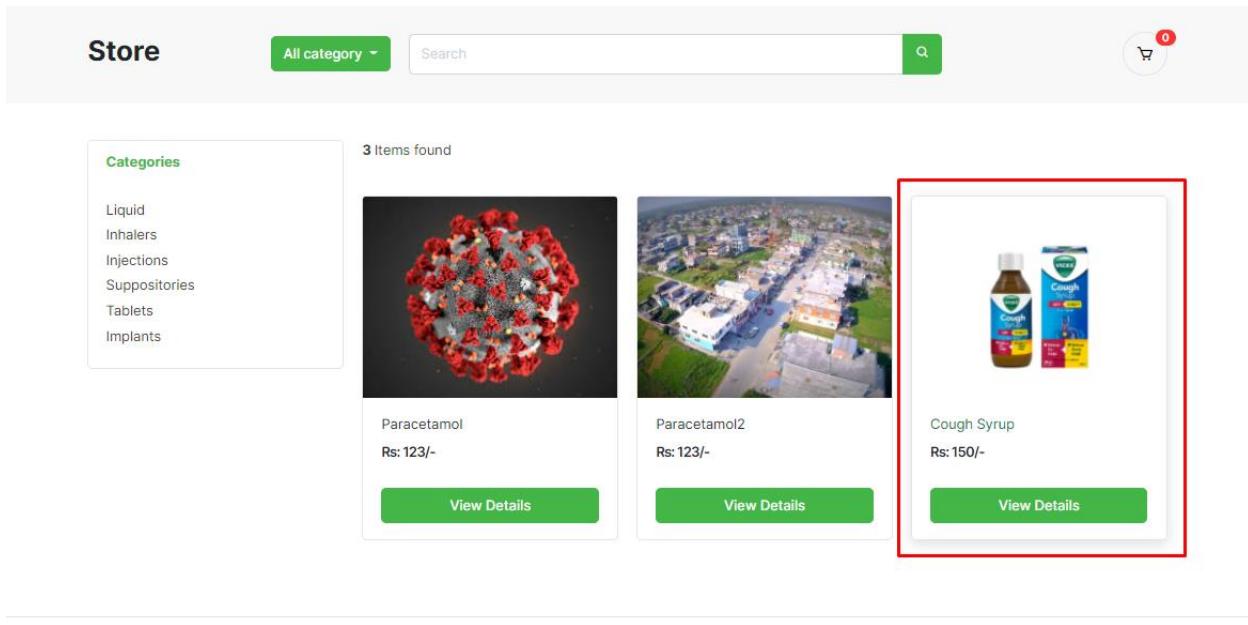


Figure 186: Screenshot of store page with products.

Here the user is viewing the product on pharmacy store.

#### 4.4.2. Testing Create Operation of the Blog

*Table 61: Integration testing to create operation of blog.*

<b>Objective</b>	To test if counsellor can add a new blog.
<b>Action</b>	Login with counsellor account and navigate to its dashboard to add blog.
<b>Expected Result</b>	Counsellor should be able to add a new medicine. The blog should be stored in a database. The new blog should be displayed to the admin. The new blog should be displayed to the user.
<b>Actual Result</b>	Counsellor was able to add a new product. The blog was stored in a database. The new blog was displayed to the admin. The new blog was displayed to the user
<b>Conclusion</b>	Test Successful.

**Add Blog**

Title:

Content:

If you want to be as healthy as possible, there are no treadmills or weight machines required.

Don't just take my word for it — look to the longest-lived people in the world for proof. People in the world's Blue Zones— the places around the world with the highest life expectancy — don't pump iron, run marathons or join gyms.

Instead, they live in environments that constantly nudge them into moving without even thinking about it. This means that they grow gardens, walk throughout the day, and minimize mechanical conveniences for house and yard work.

Powered by Froala

Characters : 551

Image:

 No file chosen

**Add Blog**

**Clear**

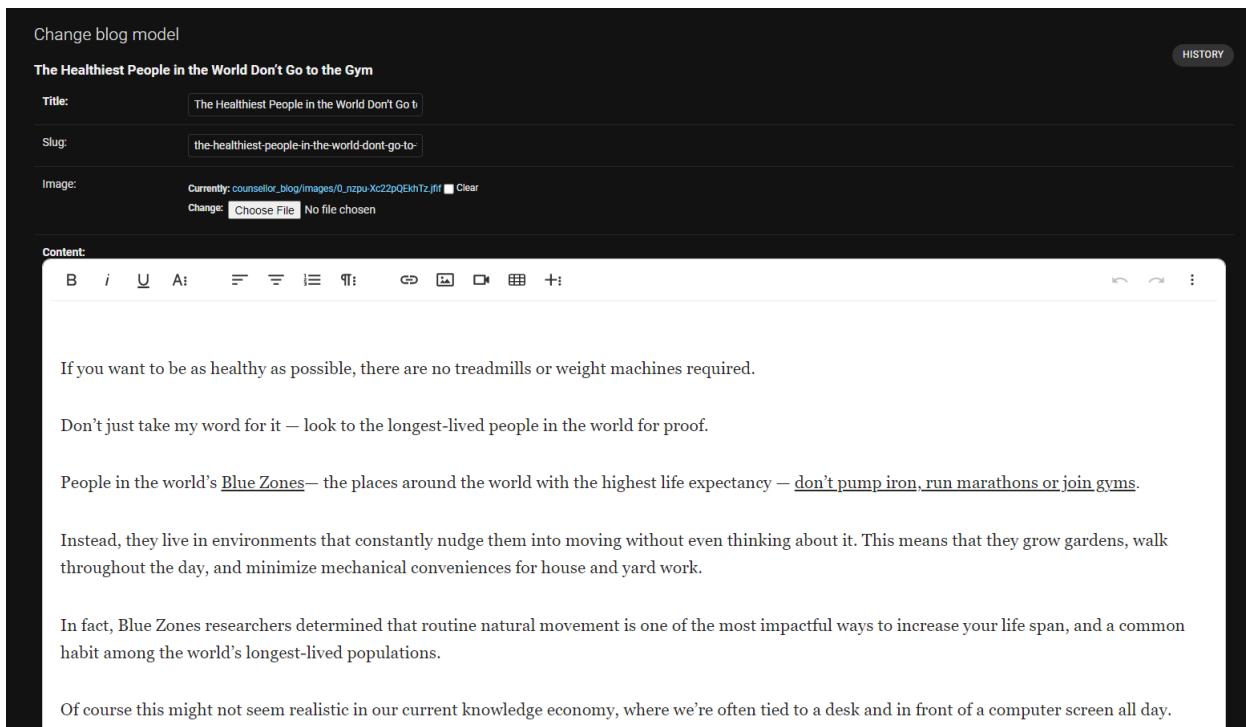
Figure 187: Screenshot of blog being added by counsellor.

Here the counsellor is adding a blog.

TITLE	DATE	UPDATE	REMOVE
 The Healthiest People in the World Don't Go to the Gym Alternative Name: name1	April 25, 2022, 10:35 a.m.	<b>Update</b>	<b>Remove</b>

Figure 188: Screenshot of added blogs by counsellor.

The recently added blog is viewed by counsellor.



*Figure 189: Screenshot of admin viewing blog.*

The blog added on previous image is viewed on above image by admin.

## All Blogs

Manish Shrestha-April 25, 2022, 5 a.m.  
[Read](#)

Manish Shrestha-April 25, 2022, 5:04 a.m.  
[Read](#)

1 2 3 4 →

Teen Counselling Center-April 25, 2022, 10:55 a.m.  
[Read](#)

*Figure 190: Screenshot of user viewing all blogs.*

The user is viewing the recently added blog.

#### 4.4.3. Testing Read Operation of the Medicine

Table 62: Integration testing to read operation of medicine.

<b>Objective</b>	To test if admin and user can view all products.
<b>Action</b>	Navigate to the admin dashboard and click Medicine model to view the medicines Navigate to the user dashboard and click "Store" to view the medicines
<b>Expected Result</b>	Admin and users should be able to view all medicines.
<b>Actual Result</b>	Admin and users can view all medicines.
<b>Conclusion</b>	Test Successful.

Change add\_product

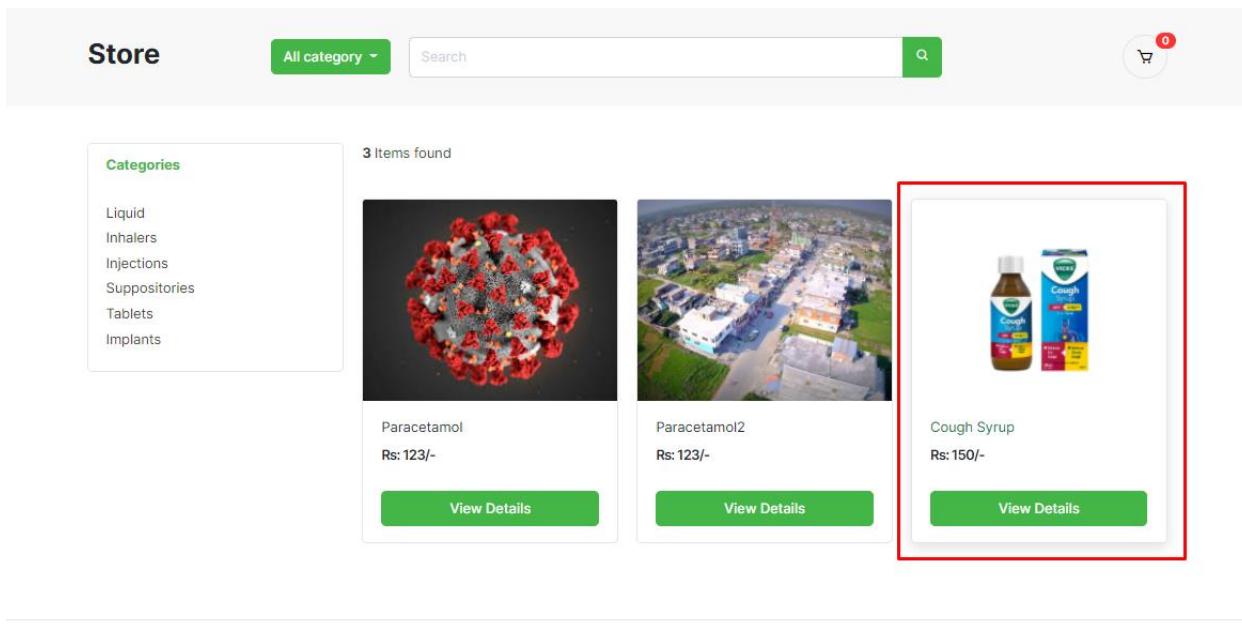
Cough Syrup

HISTORY

Product name:	Cough Syrup
Slug:	cough-syrup
Image:	Currently: photos/product_images/Vicks_AU_Cough_2in1_Syrup_front.webp Change: Choose File No file chosen
Cost:	150
Stock:	10
Dose child:	15ml
Dose adult:	30ml
Category:	Liquid <input type="button" value="edit"/> <input type="button" value="add"/>
Contraindication:	Placeholder text: Lorem Ipsum has been the industry's standard
Indication:	Placeholder text: Lorem Ipsum has been the industry's standard
Special precautions:	Placeholder text: Lorem Ipsum has been the industry's standard
Adverse effect:	Placeholder text: Lorem Ipsum has been the industry's standard
Pharmacy name id:	Test Pharmacy nepal <input type="button" value="edit"/> <input type="button" value="add"/>

Figure 191: Screenshot of admin viewing product/medicine.

Here, the admin is viewing the product/Medicine.



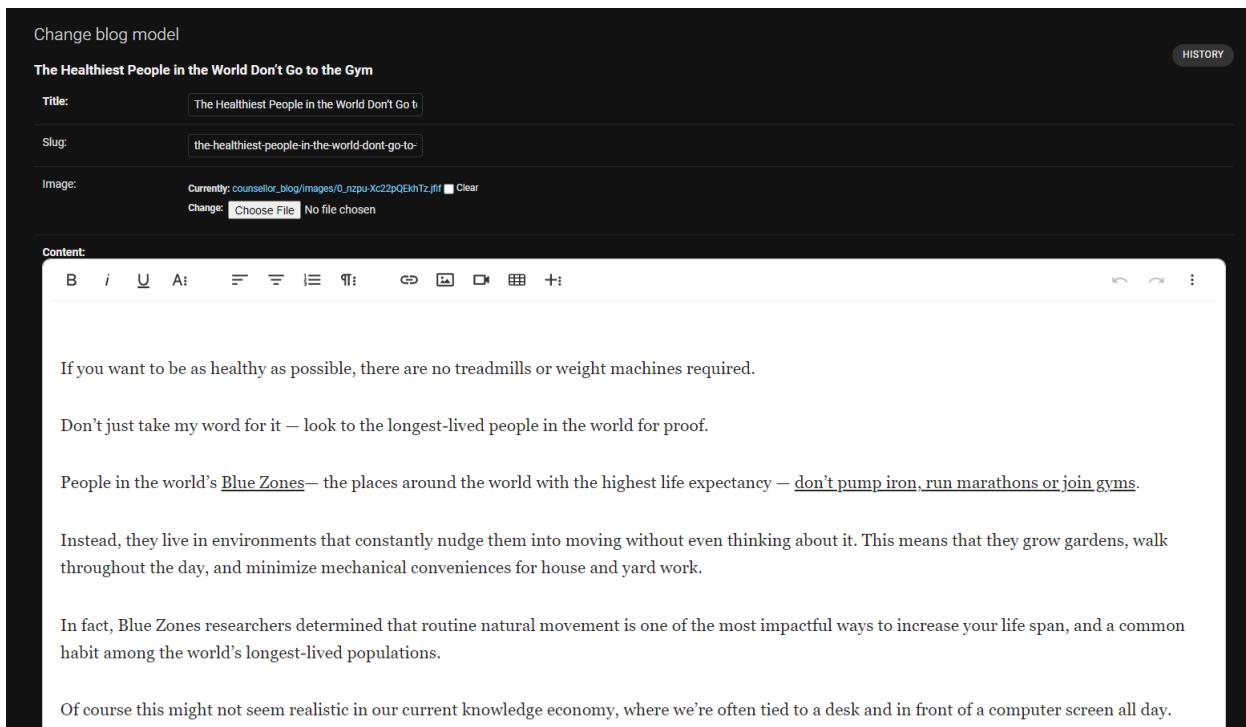
*Figure 192: Screenshot of user viewing medicine.*

Here the user is viewing the product.

#### 4.4.4. Testing Read Operation of the Blog

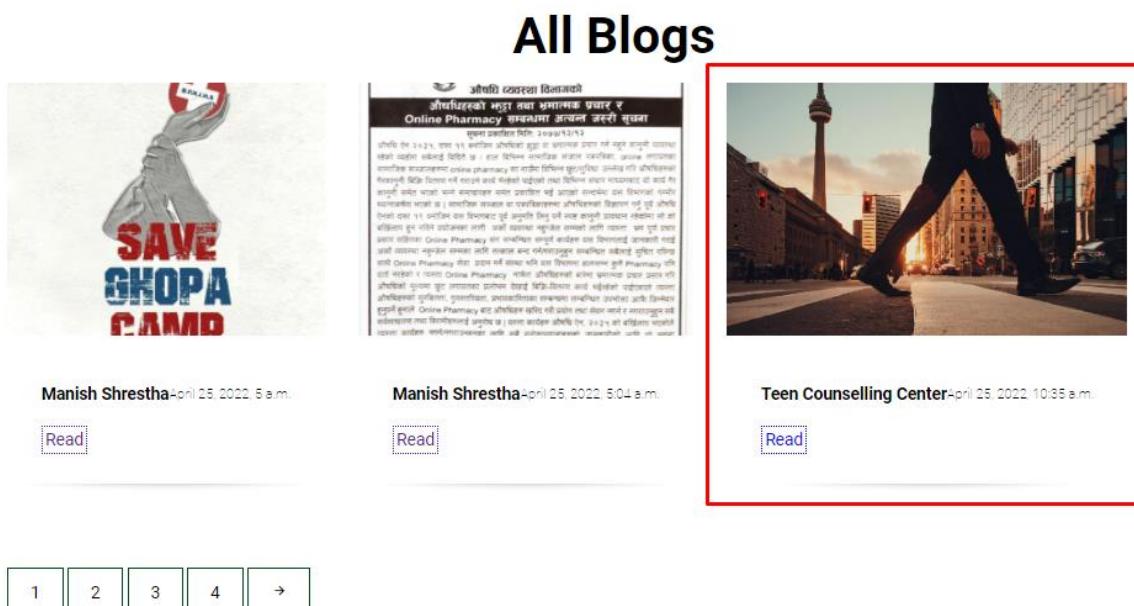
*Table 63: Integration testing to read operation of blog.*

<b>Objective</b>	To test if admin and user can view all blogs.
<b>Action</b>	<p>Navigate to the admin dashboard and click blog model to view the blogs</p> <p>Navigate to the user dashboard and click “Blog” to view the blogs.</p>
<b>Expected Result</b>	Admin and users should be able to view all blog.
<b>Actual Result</b>	Admin and users can view all blog.
<b>Conclusion</b>	Test Successful.



*Figure 193: Screenshot of admin viewing blog.*

Here the admin is viewing the blog.



*Figure 194: Screenshot of user viewing blogs.*

Here the user is viewing the blogs.

#### 4.4.1. Testing Delete Operation of the Medicine

Table 64: Integration testing to remove operation of medicine.

<b>Objective</b>	To test if pharmacist can delete a medicine.
<b>Action</b>	Login with pharmacist account and navigate to its dashboard to view added medicine.
<b>Expected Result</b>	<p>Pharmacist should be able to remove a new medicine.</p> <p>The medicine should be deleted from the database.</p> <p>The deleted medicine should be removed from the admin dashboard.</p> <p>The deleted medicine should be removed from the user view.</p>
<b>Actual Result</b>	<p>Pharmacist was able to add a new product.</p> <p>The medicine was removed from the database.</p> <p>The deleted product was removed from admin dashboard.</p> <p>The deleted product was removed from the user view.</p>
<b>Conclusion</b>	Test Successful.

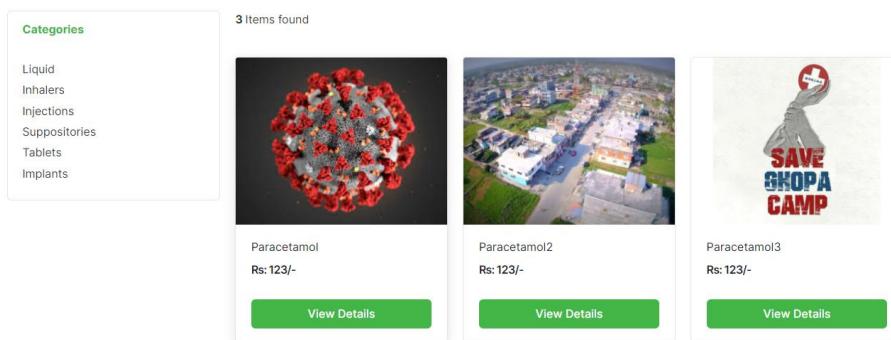


Figure 195: Screenshot of store page.

Image above was taken before removing product.

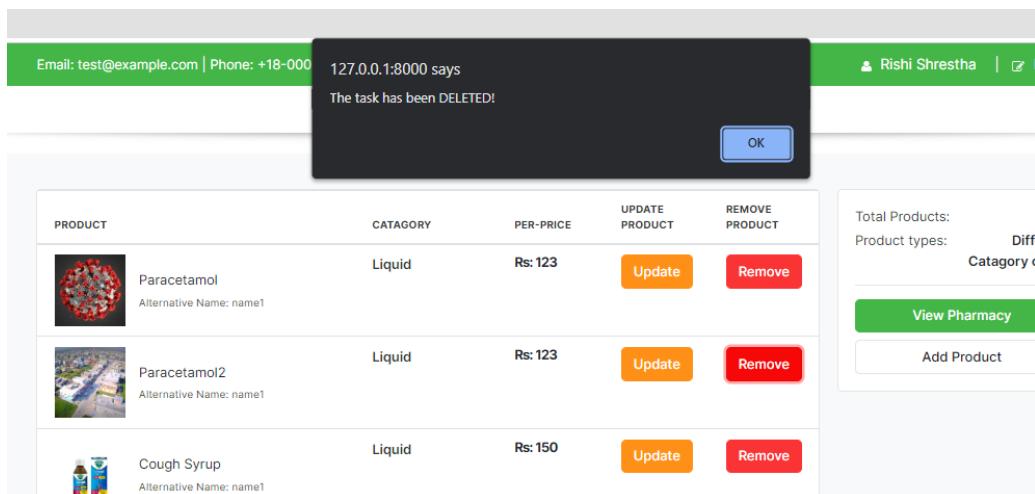


Figure 196: Screenshot of product being removed from the store.

Here the product is being removed.

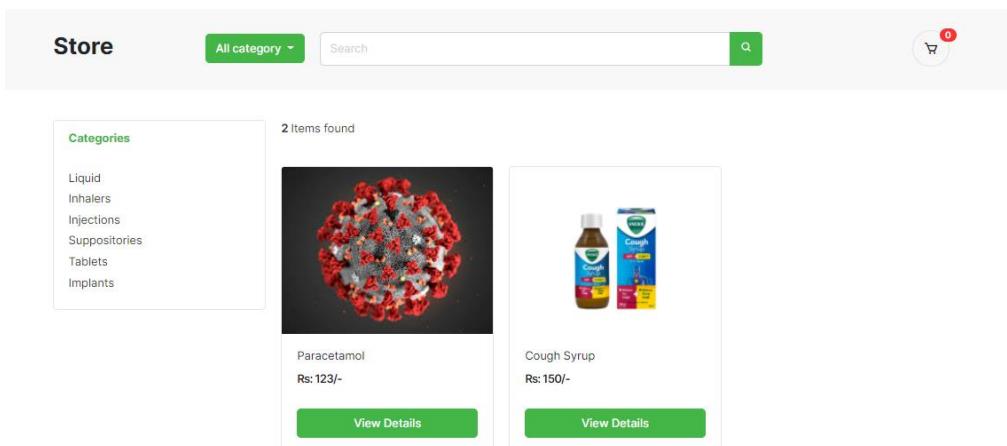


Figure 197: Screenshot of store page after removing products.

Image above was taken after removing product.

Select add_product to change							ADD ADD_PRODUCT +	
Action:	PRODUCT NAME	SLUG	IMAGE	COST	STOCK	CATEGORY	CREATED DATE	MODIFIED DATE
<input type="checkbox"/>	Cough Syrup	cough-syrup	photos/product_images/Vicks_AU_Cough_2in1_Syrup_front.webp	150	10	Liquid	April 25, 2022, 10:43 a.m.	April 25, 2022, 10:43 a.m.
<input type="checkbox"/>	Paracetamol	paracetamol	photos/product_images/images.jfif	123	12	Liquid	April 17, 2022, 12:30 a.m.	April 17, 2022, 12:30 a.m.
2 add_products								

Figure 198: Screenshot of admin page showing products.

The removed product is also removed form admin panel.

#### 4.4.2. Testing Delete Operation of the Blog

Table 65: Integration testing to remove operation of blog.

<b>Objective</b>	To test if counsellor can delete a blog.
<b>Action</b>	Login with counsellor account and navigate to its dashboard to view added blog.
<b>Expected Result</b>	<p>Counsellor should be able to remove a new medicine.</p> <p>The blog should be deleted from the database.</p> <p>The deleted blog should be removed from the admin dashboard.</p> <p>The deleted blog should be removed from the user view.</p>
<b>Actual Result</b>	<p>Counsellor was able to add a new product.</p> <p>The blog was removed from the database.</p> <p>The deleted blog was removed from admin dashboard.</p> <p>The deleted blog was removed from the user view.</p>
<b>Conclusion</b>	Test Successful.

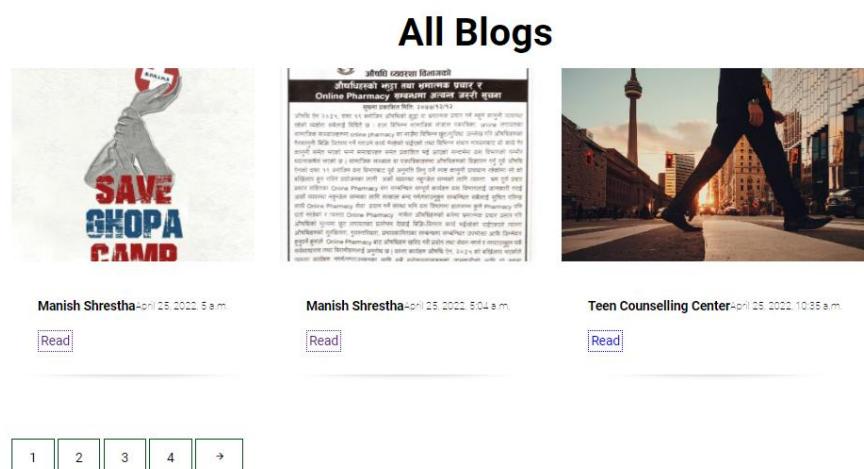


Figure 199: Screenshot of blog page.

Image above was taken before removing blog.

TITLE	DATE	UPDATE	REMOVE
 The Healthiest People in the World Don't Go to the Gym Alternative Name: name1	April 25, 2022, 10:35 a.m.	<a href="#">Update</a>	<a href="#">Remove</a>

Figure 200: Screenshot of blog added by a counsellor.

Image above was taken before removing blog.

TITLE	DATE	UPDATE	REMOVE

Figure 201: Screenshot of empty table representing zero blog.

Image above was taken after removing the blog.

## All Blogs



**प्रति दिन खाना**  
गुडियार्स वर्ष का खाना प्राप्त है  
Online Pharmacy खाना अपने जर्सी सहज

खाना लेने का एक बड़ा फायदा है। यह आपको जल्दी खाना लेने की अवसरा देता है। और आप अपने खाना का बहुत अच्छा इंजिनियरिंग कर सकते हैं। यह आपको अपने खाना का बहुत अच्छा इंजिनियरिंग कर सकते हैं। यह आपको अपने खाना का बहुत अच्छा इंजिनियरिंग कर सकते हैं। यह आपको अपने खाना का बहुत अच्छा इंजिनियरिंग कर सकते हैं। यह आपको अपने खाना का बहुत अच्छा इंजिनियरिंग कर सकते हैं। यह आपको अपने खाना का बहुत अच्छा इंजिनियरिंग कर सकते हैं। यह आपको अपने खाना का बहुत अच्छा इंजिनियरिंग कर सकते हैं। यह आपको अपने खाना का बहुत अच्छा इंजिनियरिंग कर सकते हैं। यह आपको अपने खाना का बहुत अच्छा इंजिनियरिंग कर सकते हैं। यह आपको अपने खाना का बहुत अच्छा इंजिनियरिंग कर सकते हैं। यह आपको अपने खाना का बहुत अच्छा इंजिनियरिंग कर सकते हैं। यह आपको अपने खाना का बहुत अच्छा इंजिनियरिंग कर सकते हैं। यह आपको अपने खाना का बहुत अच्छा इंजिनियरिंग कर सकते हैं।

Manish Shrestha-apr/25/2022, 5 a.m.  
[Read](#)

**प्रति दिन खाना**  
गुडियार्स वर्ष का खाना प्राप्त है  
Online Pharmacy खाना अपने जर्सी सहज

खाना लेने का एक बड़ा फायदा है। यह आपको जल्दी खाना लेने की अवसरा देता है। और आप अपने खाना का बहुत अच्छा इंजिनियरिंग कर सकते हैं। यह आपको अपने खाना का बहुत अच्छा इंजिनियरिंग कर सकते हैं। यह आपको अपने खाना का बहुत अच्छा इंजिनियरिंग कर सकते हैं। यह आपको अपने खाना का बहुत अच्छा इंजिनियरिंग कर सकते हैं। यह आपको अपने खाना का बहुत अच्छा इंजिनियरिंग कर सकते हैं। यह आपको अपने खाना का बहुत अच्छा इंजिनियरिंग कर सकते हैं। यह आपको अपने खाना का बहुत अच्छा इंजिनियरिंग कर सकते हैं। यह आपको अपने खाना का बहुत अच्छा इंजिनियरिंग कर सकते हैं। यह आपको अपने खाना का बहुत अच्छा इंजिनियरिंग कर सकते हैं। यह आपको अपने खाना का बहुत अच्छा इंजिनियरिंग कर सकते हैं। यह आपको अपने खाना का बहुत अच्छा इंजिनियरिंग कर सकते हैं।

Manish Shrestha-apr/25/2022, 5:04 a.m.  
[Read](#)

1 2 3 4 →

Figure 202: Screenshot of blog page.

Here the removed blog is also removed from the user view.

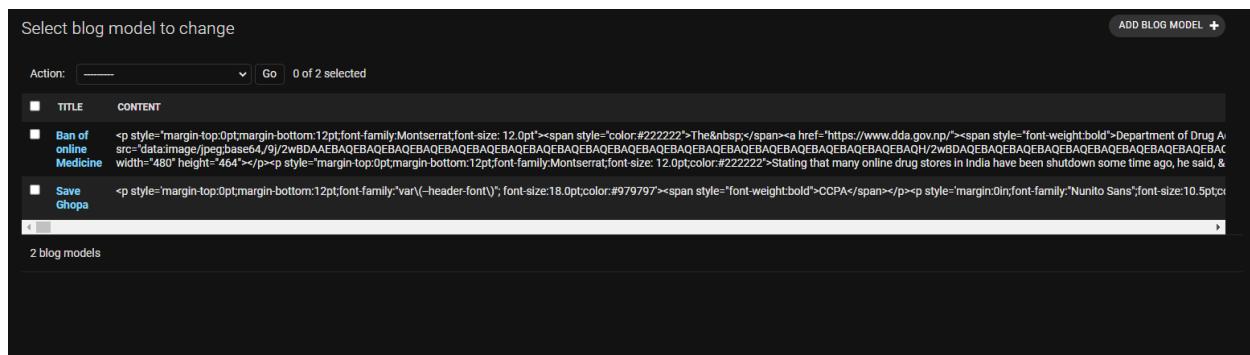


Figure 203: Screenshot of admin viewing blogs.

Here the blog is removed from the admin panel.

## 4.5. System Testing

### 4.5.1. Run the application in different browsers.

Table 66: System testing 1.

<b>Objective</b>	To test if the system runs on chrome and Firefox browser.
<b>Action</b>	Run the system and open chrome and run <a href="http://127.0.0.1:8000/">http://127.0.0.1:8000/</a> Open Firefox and run <a href="http://127.0.0.1:8000/">http://127.0.0.1:8000/</a>
<b>Expected Result</b>	The home page must be open on both the browser.
<b>Actual Result</b>	Home page was opened on both browsers.
<b>Conclusion</b>	Test Successful.

```
PS D:\FYP\code\fyp> & D:/FYP/code/venv/Scripts/activate.ps1
(venv) PS D:\FYP\code\fyp> py manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
April 26, 2022 - 09:49:43
Django version 3.2.9, using settings 'fpo.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

Figure 204: Starting the application.

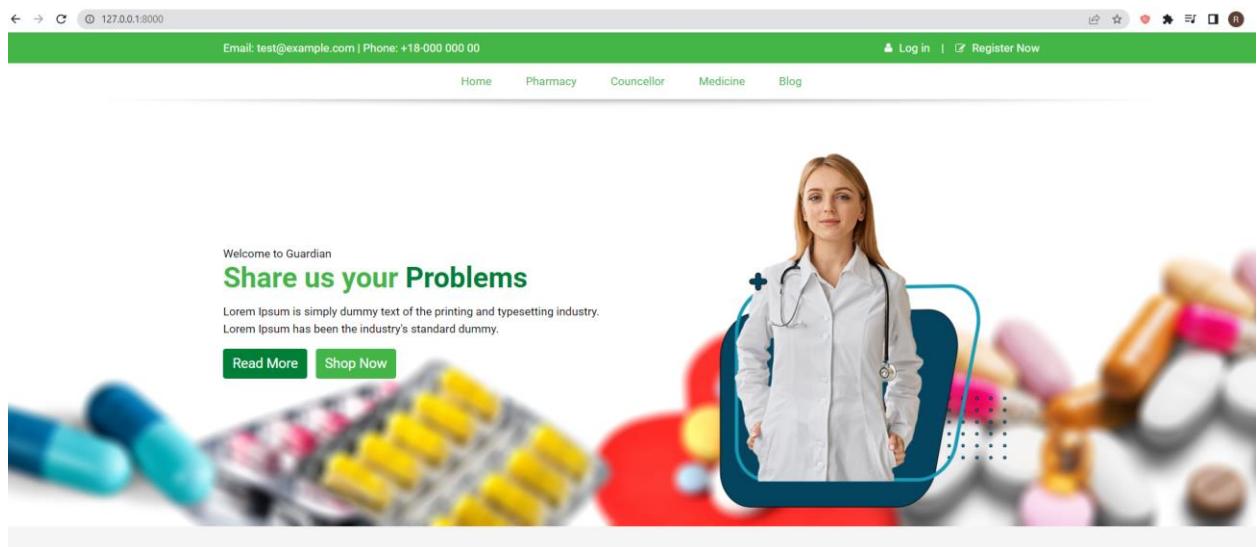


Figure 205: Home page viewed on chrome.

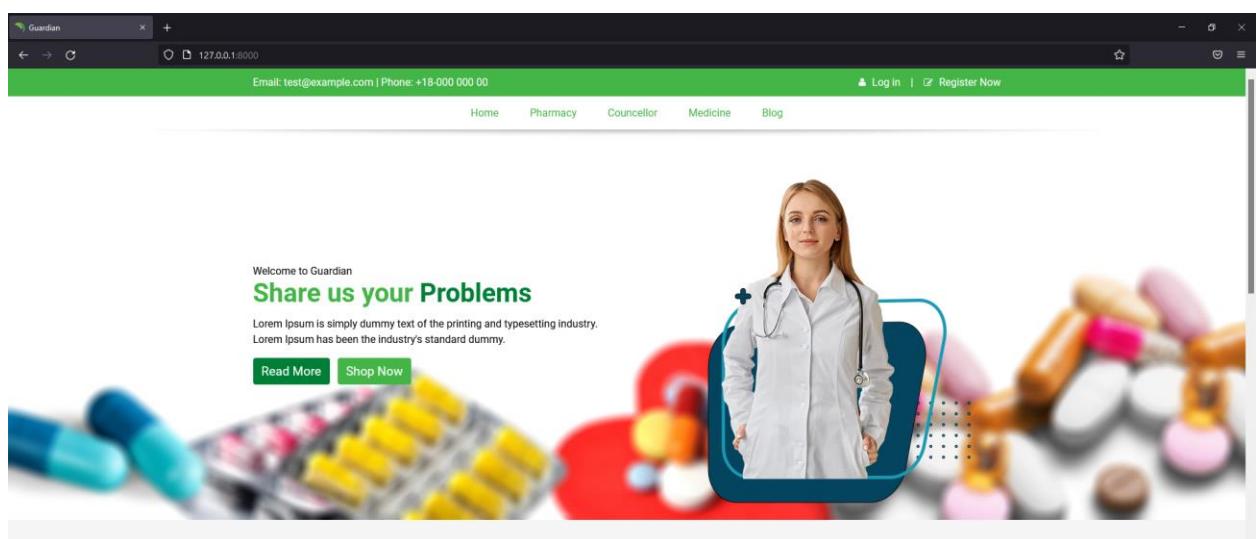


Figure 206: Home page viewed on Firefox.

#### 4.5.2. Run the application in online terminal for deployment.

Table 67: System Testing 2

<b>Objective</b>	To run the application in online terminal for deployment.
<b>Action</b>	Upload the file repository on GitHub and clone it on PythonAnywhere and run the program. Run <a href="http://guardianfyp.pythonanywhere.com/">http://guardianfyp.pythonanywhere.com/</a> in the browser.
<b>Expected Result</b>	The home page is expected to launch in the browser.
<b>Actual Result</b>	The home page is launched in the browser.
<b>Conclusion</b>	Test Successful.

The screenshot shows the PythonAnywhere dashboard. At the top, there's a navigation bar with links for Send feedback, Forums, Help, Blog, Account, and Log out. Below that is the Python logo and the text "python anywhere". To the right, there are links for Dashboard, Consoles, Files, Web, Tasks, and Databases. The main area is titled "Dashboard" and "Welcome, guardianfyp". It displays four sections: "Recent Consoles" (with two entries: "Bash console 24021271" and "Bash console 24021157"), "Recent Files" (with five entries: base.html, home.html, settings.py, guardianfyp\_pythonanywhere\_com\_ws..., and requirements.txt), "Recent Notebooks" (with a message: "Your account does not support Jupyter Notebooks. Upgrade your account to get access!"), and "All Web apps" (with a link to guardianfyp.pythonanywhere.com). At the bottom, there are buttons for "+ Open another file" and "Browse files". A footer at the very bottom contains copyright information: "Copyright © 2011-2022 PythonAnywhere LLP — Terms — Privacy & Cookies".

Figure 207: PythonAnywhere dashboard.

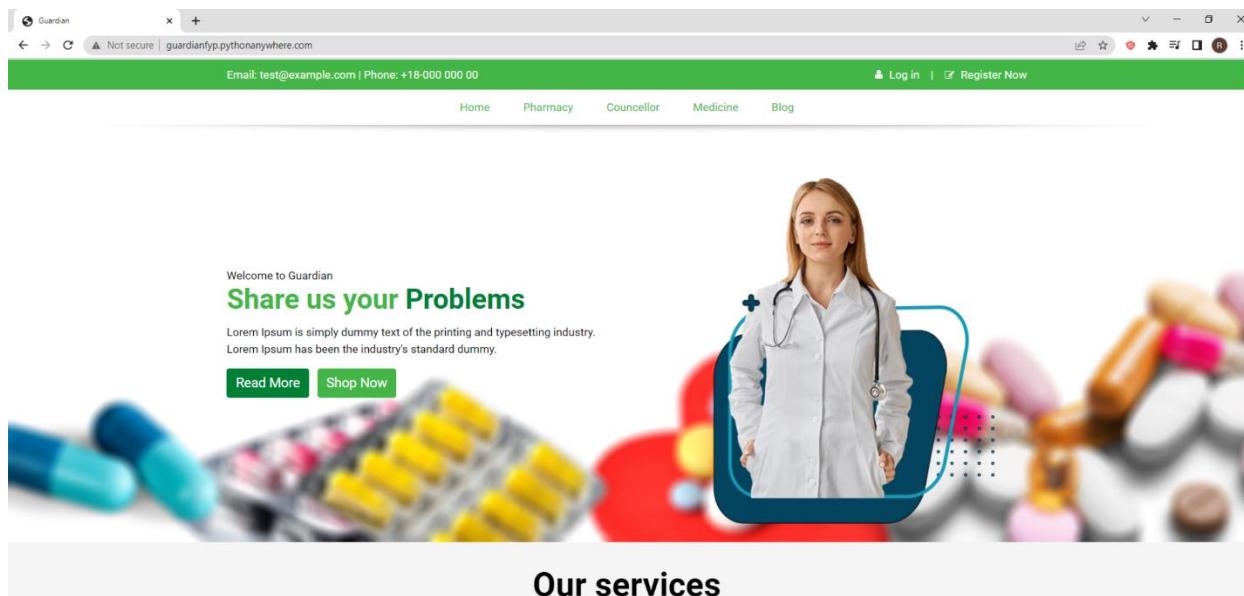


Figure 208: Homepage viewed after deployment.

#### 4.6. Critical Analysis

As the testing (white box, black box, integration, client evaluation, and system testing), it was found that all of the tests were successful. All feasible testing was carried out at this stage of the testing process, with each step of testing being explained with either snippet of code, graphical representations, and process elaboration. The system's performance was excellent, and the application was simple to use. For invalid and incorrect inputs, a good user experience was provided by prompting users with an appropriate message.

The overall analysis of the testing carried out as per the test cases the test result was true for all the test case carried out which signifies that the features are working as intended, the overall performance of the system was excellent similarly the UI rendered, and overall design perfectly matched with the project theme. even though there were some problems were encountered with the processing speed which will be dealt accordingly in coming days. likewise, as per the form testing, it seems to need few securities related validation which will be dealt accordingly in coming days. Similarly, there is not any issue in rendering views as per the routing. Also, the API has been working great with fine result.

## 5. Conclusion

The project has been successfully completed on time and the learning experience has been awesome throughout the development of the system. The project main aim to provide an online store where customers can place their orders and to book counsellor has been successfully fulfilled and objective like testing and client review, using different API has been well fulfilled and a complete web app named 'Guardian' has been successfully developed integrating all the features and functionality outlined before. The recent unpredicted pandemic has caused a drastic change in the world. It has jolted the economy, lifestyle, and health care system worldwide. Lockdown and mobility restrictions caused inconvenience in various sectors. Understanding the risk of physical contact during covid time and the demanding need of people to have online presence of doctors, pharmacists, a project named 'Guardian' is introduced.

The projects serve as an internet hub for pharmacist, counsellor, and their parents. The customer can upload their prescription sheet and order the medicine. It also helps them to select, order and pay online or offline. Also, it informs patients (clients) about their medicines they are taking. For the development of the system database like PostgreSQL and SQLite has been used with VS code as a code editor. The backend is done using python programming language using Django framework. The projects are well compared with other similar projects and the features has been critically evaluated. Similarly different methodology was implemented like waterfall, extreme programming but Scrum has been selected. Hence, overall development of the system is based on the scrum methodology as it satisfies the need and requirements of the projects to be developed. All the needed design has been developed properly like sequence, communication, use case, ERD and so on has been designed nicely. The projects testing has been properly done with proper test case and the testing included white-box testing, black-box testing and system testing. In the same way, survey has also been conducted among the users and the surveys include pre-survey and post-survey form. After the user feedback and analysing it, the project has been updated accordingly.

Overall, the process of developing the system and learning experience has been awesome. Doing all the component of the system by own research and plan, it helped

me a lot to increase and boost my research and critical thinking. Also, it helped me to increase my expertise related to different tools and technologies such as Django.

## 5.1. Issues

### 5.1.1. Legal Issues

The medicine sold are not expired and is not over-priced. All the medicine are kept and stored in proper environment needed for the medicine to remain in good condition. Decision Also, any component of the system or any blogs are not copied from any source and the system is free from trademark issue and copyright infringement. All the work done using the system and everything present in the system is solely copyright free. Also, the system does not falsify the data and protect right to privacy of the customer. In the same way, all the tools and technologies used are open source and free to use. As per the previous notice published on Kathmandu post, the government of the Nepal has banned in selling the drugs through online prescription, hence after the band if any pharmacy sells through online then it will raise the legal issue in action. Although the system proposed and developed is also related to online prescription and online counselling and health check-up, it does not raise any legal issue as it is for educational purpose only and for now, there is not any intention to sell medicine using this system. If in coming future, the system plans to sell online then the system needs to have document from the government bodies which will surely be done if gone online. Overall, the system does not raise any issues.

### 5.1.2. Social Issues

The issue or any activities that is against the social norms and values and influence the greater number of people in the society is called a social issue influence greater number of people in the society in wrong manner. The developed project is intended to develop and work for the welfare of the people health and spread awareness about the health facility. The developed project is reliable, and the counsellor and pharmacist associated with the system are all well qualified. So, the medicine provided by the store is authentic and is provided only by the certified personnel. The system does not promote any political or any other activities that involved hatred or illegal substance.

Also, analysing the result of the surveyed from, there was not any complain about anything, and all the response were positive, which signifies there is not any issue related to social.

### **5.1.3. Ethical Issues**

The issues that are against the society standard morality of the society is called as ethical issue. The developed system does not bring any ethical issues as the system is monitored and will be upgraded timely. All the data are securely kept and is not misused by any means. The data are not sold to anyone, nor any negligence is done in securing it, which would be ethically very wrong to do so. The system does not interfere in any customer related post and getting like vendor. Also, the system does not interfere any logging controller. Any update, information available are timely present.

## **5.2. Advantages**

There are many advantages of using this application. Some of its advantages are listed below:

- The developed system helps to increase awareness about the importance of the health.
- Any user can book counsellor and books an appointment / order prescription online.
- This application provides a platform for the pharmacy to sell its product online.
- This application has an authentication phase where only a user with a valid email can use the features of the application.
- Also, there is facility of online payment through PayPal.

### 5.3. Limitations

This application has certain drawbacks in addition to its benefits. These limitations are merely issues for the time being, but eventually, they will be addressed in the future. The limitations of the application are as follows:

- The application works only when internet service is available.
- The government of Nepal have put banned on selling and buying medicine online.
- The application lacks a feature to track user order.
- There is no feature to reply to a review.

### 5.4. Future Work

The functions and features of the application are completed as per the proposal. However, this application can be improved further, and new features can be added to the application. Some of the features that can be added to enhance this application are as follows:

- Implementing reply feature for comment and review
- Order tracking
- Prescription reading
- Implementing Chat system
- Enhance Multi-store platform
- Host on trustworthy hosting platform.

## 6. References

- Agilealliance.org, 2021. *Extreme Programming.* [Online] Available at: <https://www.agilealliance.org/membership> [Accessed 13 11 2021].
- Biscobing, J., 2021. *TechTarget.* [Online] Available at: <https://www.techtarget.com/searchdatamanagement/definition/entity-relationship-diagram> [Accessed 16 11 2021].
- Dhani Pharmacy, 2021. *Dhani Pharmacy.* [Online] Available at: <https://www.dhani.com/pharmacy-store/> [Accessed 21 10 2021].
- Digite, 2021. *Scrum.* [Online] Available at: <https://www.digite.com/agile/scrum-methodology/> [Accessed 11 10 2021].
- Geeks for Geeks, 2019. *Introduction to Visual Studio.* [Online] Available at: <https://www.geeksforgeeks.org/introduction-to-visual-studio/> [Accessed 20 11 2021].
- Geeks for Geeks, 2021. *Django Project MVT Structure.* [Online] Available at: <https://www.geeksforgeeks.org/django-project-mvt-structure/> [Accessed 16 11 2021].
- Hamilton, T., 2022. *Guru99.* [Online] Available at: <https://www.guru99.com/software-testing-introduction-importance.html> [Accessed 1 03 2022].
- Lewis, S., 2020. *TechTarget.* [Online] Available at: <https://www.techtarget.com/searchsoftwarequality/definition/waterfall-model> [Accessed 23 11 2021].

Nepmeds, 2021. *Nepmeds the health partner.* [Online] Available at: <https://www.nepmeds.com.np/> [Accessed 10 11 2021].

Netmeds, 2021. *Netmeds pharmacy.* [Online] Available at: <https://www.netmeds.com/> [Accessed 13 11 2021].

News Team, 2021. *ktm2day.* [Online] Available at: <https://www.ktm2day.com/govt-bans-online-pharmacy-nepal/> [Accessed 12 02 2022].

oracle, 2021. *OCI.* [Online] Available at: <https://www.oracle.com/database/what-is-database/> [Accessed 24 11 2021].

PostgreSQL, 2021. *PostgreSQL Tutorial.* [Online] Available at: <https://www.postgresqltutorial.com/postgresql-getting-started/what-is-postgresql/#:~:text=PostgreSQL%20is%20an%20advanced%2C%20enterprise,by%20the%20open%2Dsource%20community.> [Accessed 23 11 2021].

WHO, 2022. *WHO.* [Online] Available at: <https://www.who.int/news/item/22-04-2022-who-recommends-highly-successful-covid-19-therapy-and-calls-for-wide-geographical-distribution-and-transparency-from-originator> [Accessed 14 01 2022].

## 7. Appendix

### 7.1. Pre-Survey

#### 7.1.1. Pre-Survey Form

##### Guardian

Guardian is an web application that connects pharmacies and counsellors to their customers. It aims to reduce different types of difficulties faced by them during the uncertain situations.

\* Required

1. Email \*

---

2. Name \*

---

3. Have you ever felt the need of an online platform to order medicines and to connect with your counsellor simultaneously during the lockdown? \*

Mark only one oval.

 Yes No

4. Have you ever felt difficulties in understanding the prescriptions given by doctors and invoice given by pharmacist? \*

Mark only one oval.

 Yes No

5. How useful do you think an AI used in an webapp to digitalize your prescription, will be? \*

Mark only one oval.

1	2	3	4	5	
low	<input type="radio"/> High				

Thank you for giving your valuable time.

6. Do you prefer ordering medicines online? \*

Mark only one oval.

Yes  
 No

7. Do you prefer booking therapy session online? \*

Mark only one oval.

Yes  
 No

8. Rate the usefulness of this project in our daily life. \*

Mark only one oval.

1	2	3	4	5	
less	<input type="radio"/> High				

The above two figures are the question asked for the normal users via google form.

## Guardian

Guardian is an web application that connects pharmacies and counsellors to their customers. It aims to reduce different types of difficulties faced by them during the uncertain situations.

\* Required

1. Email \*

---

2. Name \*

---

3. Who are you? \*

*Mark only one oval.*

Doctor

Nurse

Pharmacist

Counsellor

Other: \_\_\_\_\_

4. What is the name of your pharmacy?(If you are an pharmacist.)

---

5. Have you ever used an application to order medicines? \*

*Mark only one oval.*

Yes

No

6. If Yes, Can you name the app name?

---

7. Have you ever faced the need of an online platform to order medicines and to connect with your counsellor simultaneously? \*

*Mark only one oval.*

Yes

No

8. Do you think every patients can understand the prescription? \*

*Mark only one oval.*

Yes

No

9. Have you ever felt difficulties in understanding the prescriptions? \*

*Mark only one oval.*

Yes

No

10. How useful do you think an Artificial Intelligence used in an webapp to digitalize your prescription, will be? \*

*Mark only one oval.*

1      2      3      4      5

Low                     High

11. Do you prefer ordering medicines online? \*

*Mark only one oval.*

Yes

No

12. Do you think finding counsellor via this application will be helpful? \*

*Mark only one oval.*

Yes

No

13. Do you prefer booking therapy session online? \*

*Mark only one oval.*

Yes

No

14. Rate the usefulness of this project in our daily life. \*

Mark only one oval.

1      2      3      4      5

less                        High

15. If you have any suggestion regarding the application, please feel free to write them below.

---

---

---

---

Thank you for giving your valuable time.

---

This content is neither created nor endorsed by Google.

Google Forms

The above four images are the screenshot of google form which was shared between the frontline workers.

### 7.1.2. Sample of Filled Pre-Survey Forms

#### Guardian

Guardian is an web application that connects pharmacies and counsellors to their customers. It aims to reduce different types of difficulties faced by them during the uncertain situations.

Email \*

chintankarki@gmail.com

Name \*

Chintan Karki

Have you ever felt the need of an online platform to order medicines and to connect with your counsellor simultaneously during the lockdown? \*

Yes

No

Have you ever felt difficulties in understanding the prescriptions given by doctors and invoice given by pharmacist? \*

Yes

No

How useful do you think an AI used in an webapp to digitalize your prescription, will be? \*

1      2      3      4      5

low                                    High

Thank you for giving your valuable time.

Do you prefer ordering medicines online? \*

Yes  
 No

Do you prefer booking therapy session online? \*

Yes  
 No

Rate the usefulness of this project in our daily life. \*

1      2      3      4      5

less                                    High

The above two images are the survey sample filled by a user.

## Guardian

Guardian is an web application that connects pharmacies and counsellors to their customers. It aims to reduce different types of difficulties faced by them during the uncertain situations.

Email \*

Madav98@gmail.com

Name \*

Dr. Madhav regmi

Who are you? \*

- Doctor
- Nurse
- Pharmacist
- Counsellor
- Other: \_\_\_\_\_

What is the name of your pharmacy?(If you are an pharmacist.)

Ariaka medical hall

Have you ever used an application to order medicines? \*

- Yes  
 No

If Yes, Can you name the app name?

---

Have you ever faced the need of an online platform to order medicines and to connect with your counsellor simultaneously? \*

- Yes  
 No

Do you think every patients can understand the prescription? \*

- Yes  
 No

Have you ever felt difficulties in understanding the prescriptions? \*

- Yes  
 No

How useful do you think an Artificial Intelligence used in an webapp to digitalize your prescription, will be? \*



Do you prefer ordering medicines online? \*

- Yes  
 No

Do you think finding counsellor via this application will be helpful? \*

- Yes  
 No

Do you prefer booking therapy session online? \*

Yes

No

Rate the usefulness of this project in our daily life. \*

1

2

3

4

5

less

High

If you have any suggestion regarding the application, please feel free to write them below.

---

Thank you for giving your valuable time.

This form was created inside of Itahari International College.

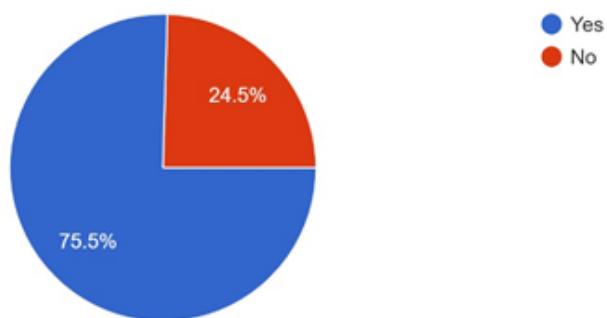
Google Forms

The above images are the answer given by a pharmacist.

### 7.1.3. Pre-Survey Result

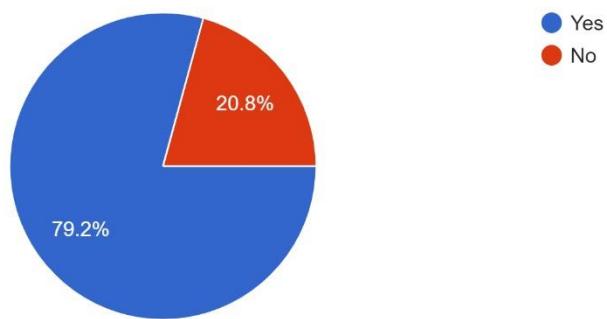
Have you ever been in such a situation where you felt the need of online platform for/your pharmacy?

53 responses

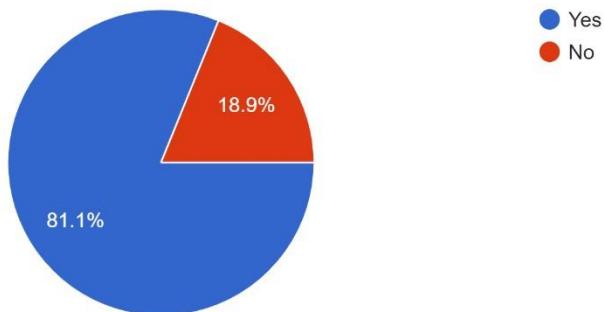


Have you ever been in such a situation where you felt the need of online platform to find out counsellor/Reach out to people(as counsellor)?

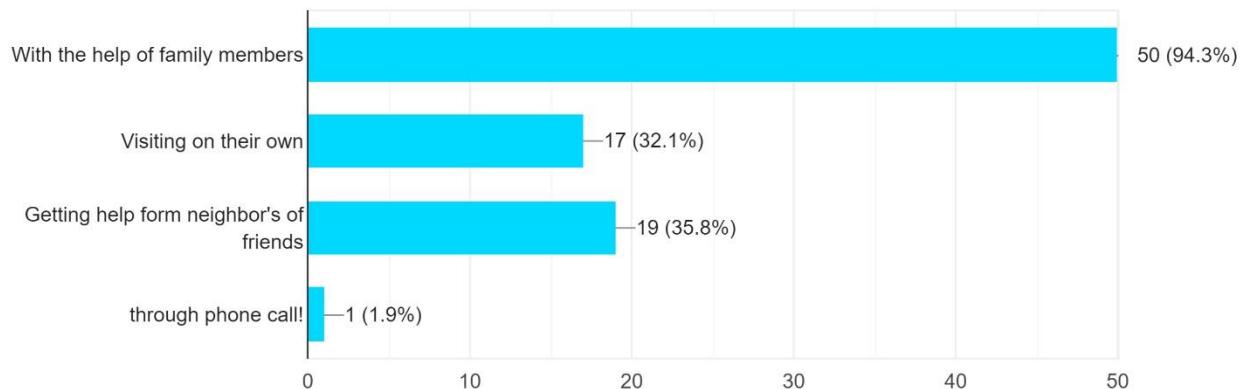
53 responses



When we were grounded because of covid lockdown do you think people needed someone to cheer them up like counsellors with whom they can ...ow they felt and release all their hidden words ?  
53 responses

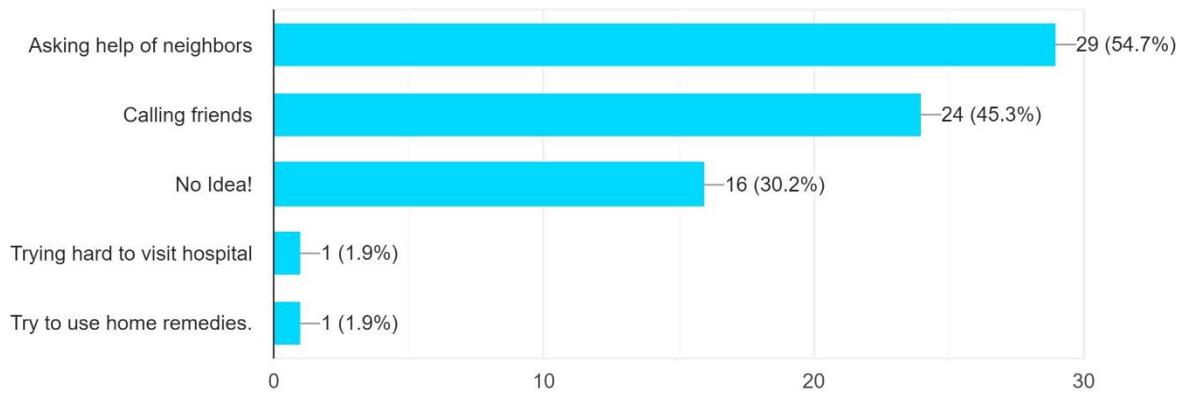


Do you have any idea how aged people buy medicines or have a therapy sessions?  
53 responses



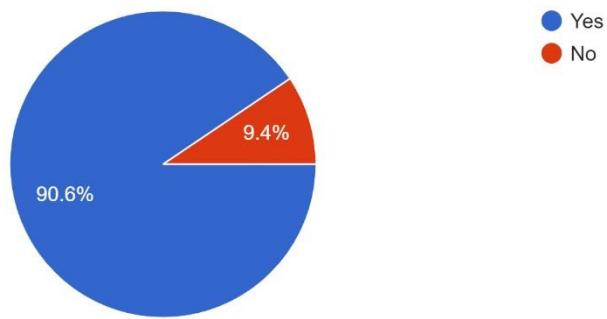
If you are home alone and you got sick then how will you buy medicines?

53 responses



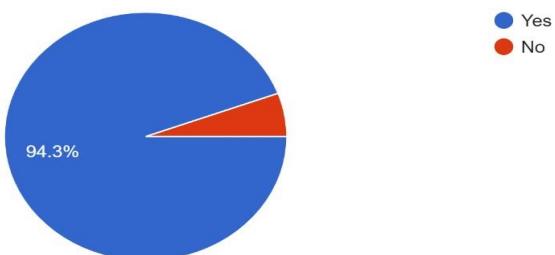
Have you ever felt difficulties in understanding the prescriptions given by doctors and invoice given by pharmacist?

53 responses



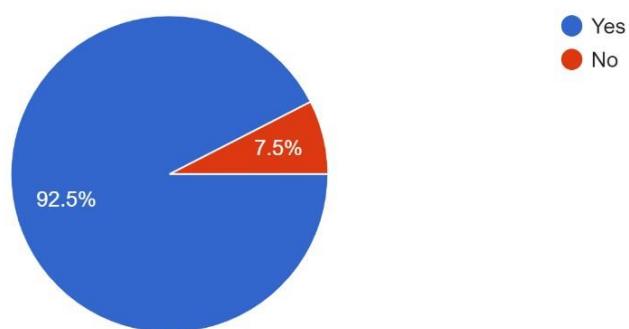
Do you think an online pharmacy with mentioned features would be better to solve the problems (mentioned above) ?

53 responses



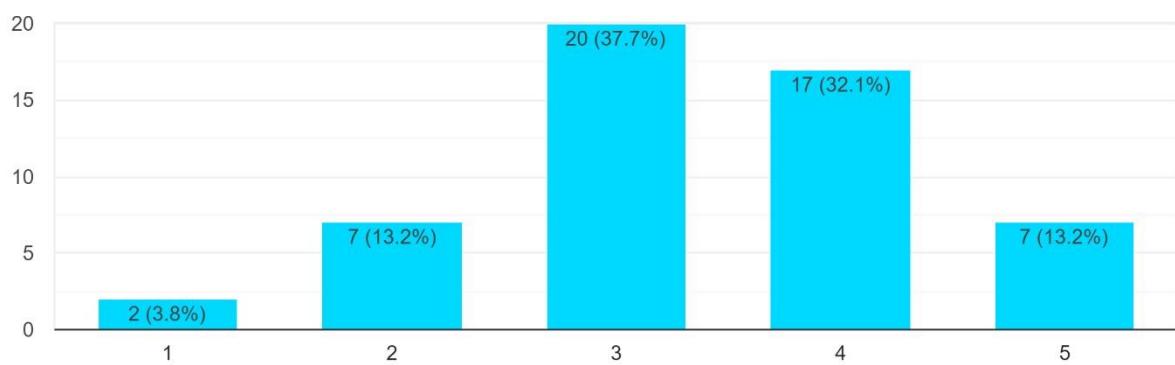
Do you think an online Counsellor with mentioned features would be better to solve the problems (mentioned above) ?

53 responses



How useful do you think an AI used in an webapp to digitalize your prescription, will be?

53 responses



## 7.2. Post-survey

### 7.2.1. Post Survey Form

#### Guardian

Before filling this form i request you to visit the site: <http://guardianfyp.pythonanywhere.com/>

\* Required

1. Name \*

---

2. Do you find this application reliable and trustworthy? \*

*Mark only one oval.*

- Yes
- No
- Maybe

3. For what purpose do you think you will use this application for? \*

*Check all that apply.*

- Booking Counsellor.
- To buy medicine
- To sell medicine
- To share prescription
- Reading blog

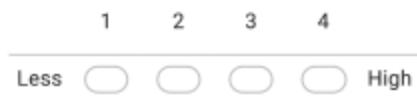
4. Select the feature you found useful in the application. \*

*Check all that apply.*

- Buying Medicine
- Options for making Payment
- Contacting counsellor
- Tracking nearby pharmacies
- Selling medicine
- Sharing prescription
- Reading Blog
- Adding Blogs
- Giving Reviews

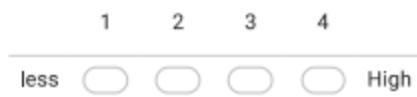
5. Select the usefulness of the project. \*

*Mark only one oval.*



6. How likely are you going to use this application? \*

*Mark only one oval.*



The above two images are the screenshot taken from the post survey form.

### 7.2.2. Sample of filed Post Survey Form

#### Guardian

Before filling this form i request you to visit the site: <http://guardianfyp.pythonanywhere.com/>

Name \*

Sandip Regmi

Do you find this application reliable and trustworthy? \*

- Yes
- No
- Maybe

For what purpose do you think you will use this application for? \*

- Booking Counsellor.
- To buy medicine
- To sell medicine
- To share prescription
- Reading blog

Select the feature you found useful in the application. \*

- Buying Medicine
- Options for making Payment
- Contacting counsellor
- Tracking nearby pharmacies
- Selling medicine
- Sharing prescription
- Reading Blog
- Adding Blogs
- Giving Reviews

Select the usefulness of the project. \*

1

2

3

4

Less



High

How likely are you going to use this application? \*

1

2

3

4

less



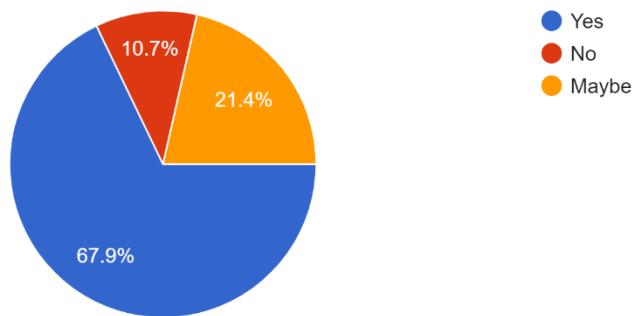
High

The above image shows the post survey filled by a user named Sandip Regmi.

### 7.2.3. Post Survey Result

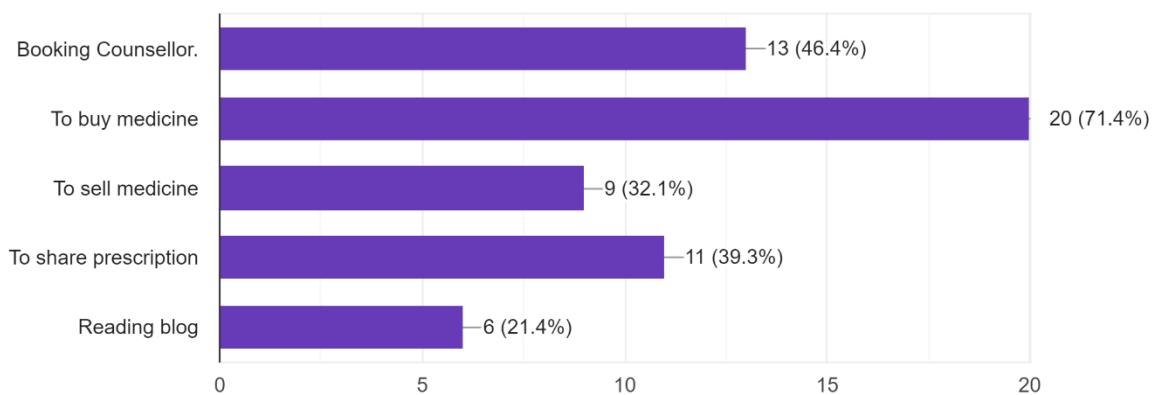
Do you find this application reliable and trustworthy?

28 responses



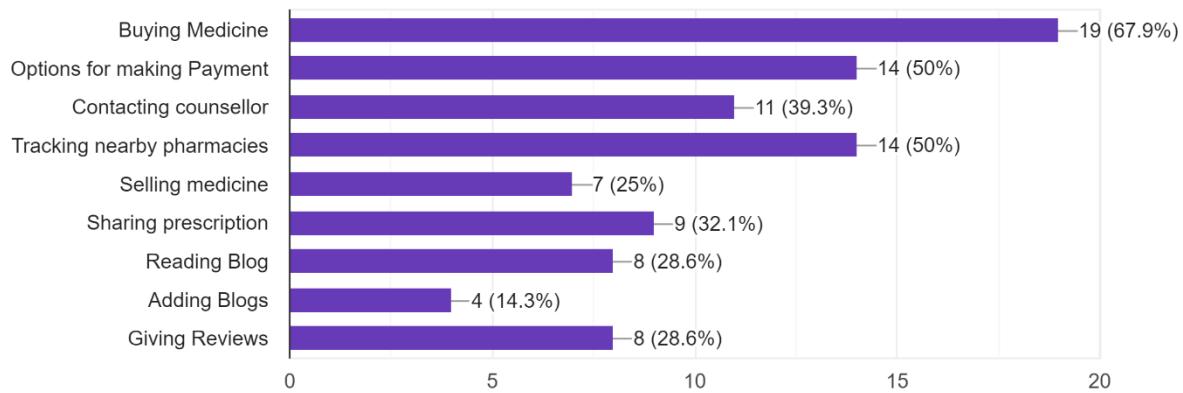
For what purpose do you think you will use this application for?

28 responses



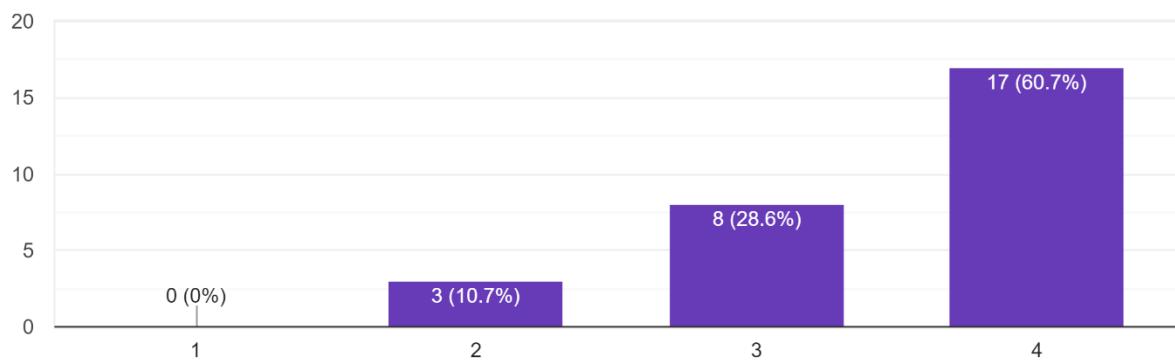
Select the feature you found useful in the application.

28 responses



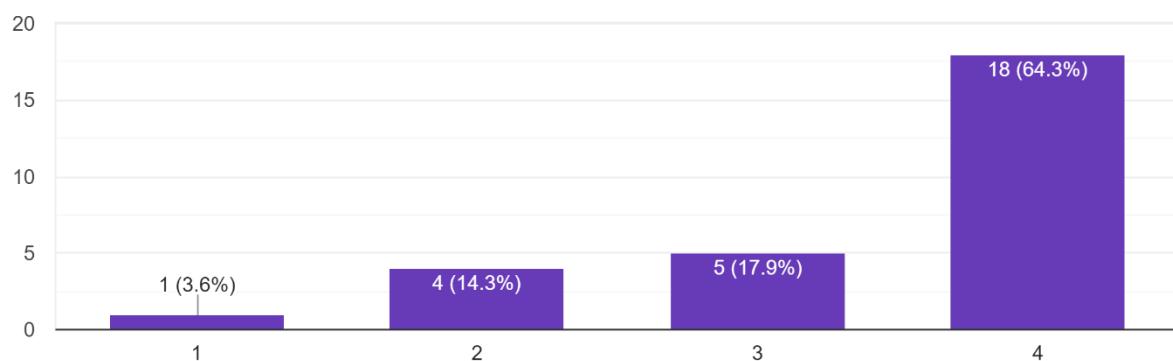
Select the usefulness of the project.

28 responses



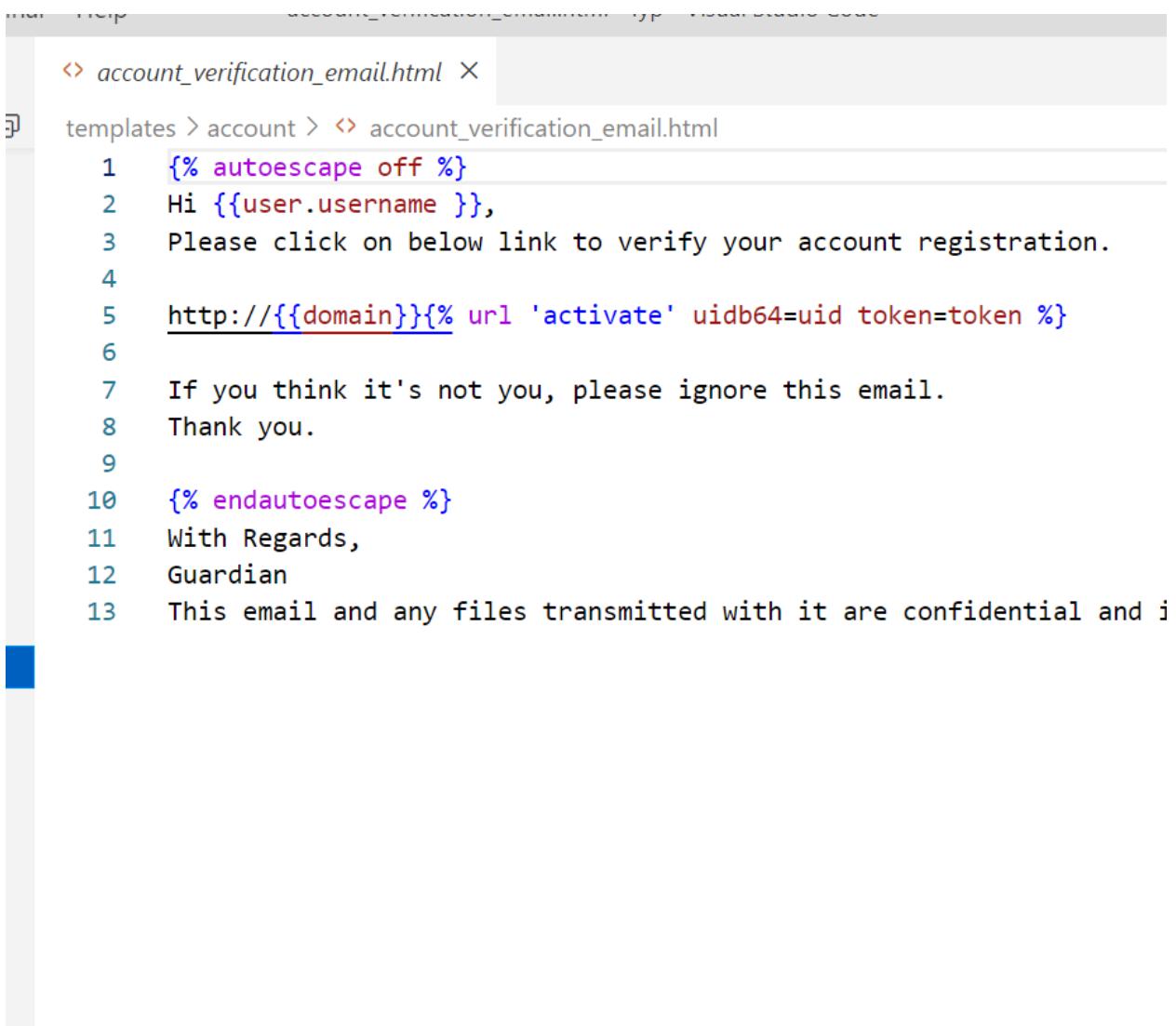
How likely are you going to use this application?

28 responses



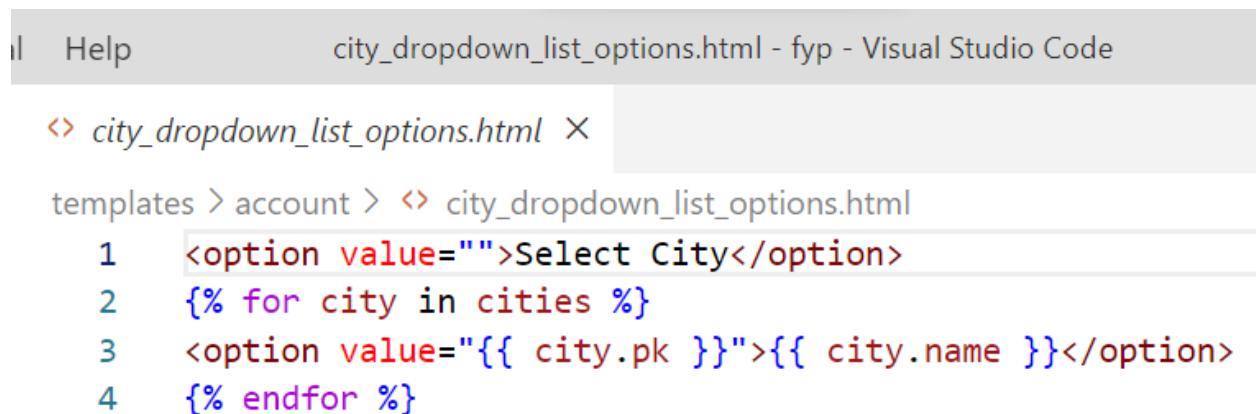
### 7.3. Sample Codes

#### 7.3.1. Code of User interface



The screenshot shows a code editor window in Visual Studio Code. The title bar says "account\_verification\_email.html". The file path is "templates > account > account\_verification\_email.html". The code itself is:

```
1  {% autoescape off %}  
2  Hi {{user.username }},  
3  Please click on below link to verify your account registration.  
4  
5  http://{{domain}}{% url 'activate' uidb64=uid token=token %}  
6  
7  If you think it's not you, please ignore this email.  
8  Thank you.  
9  
10 {% endautoescape %}  
11 With Regards,  
12 Guardian  
13 This email and any files transmitted with it are confidential and i
```



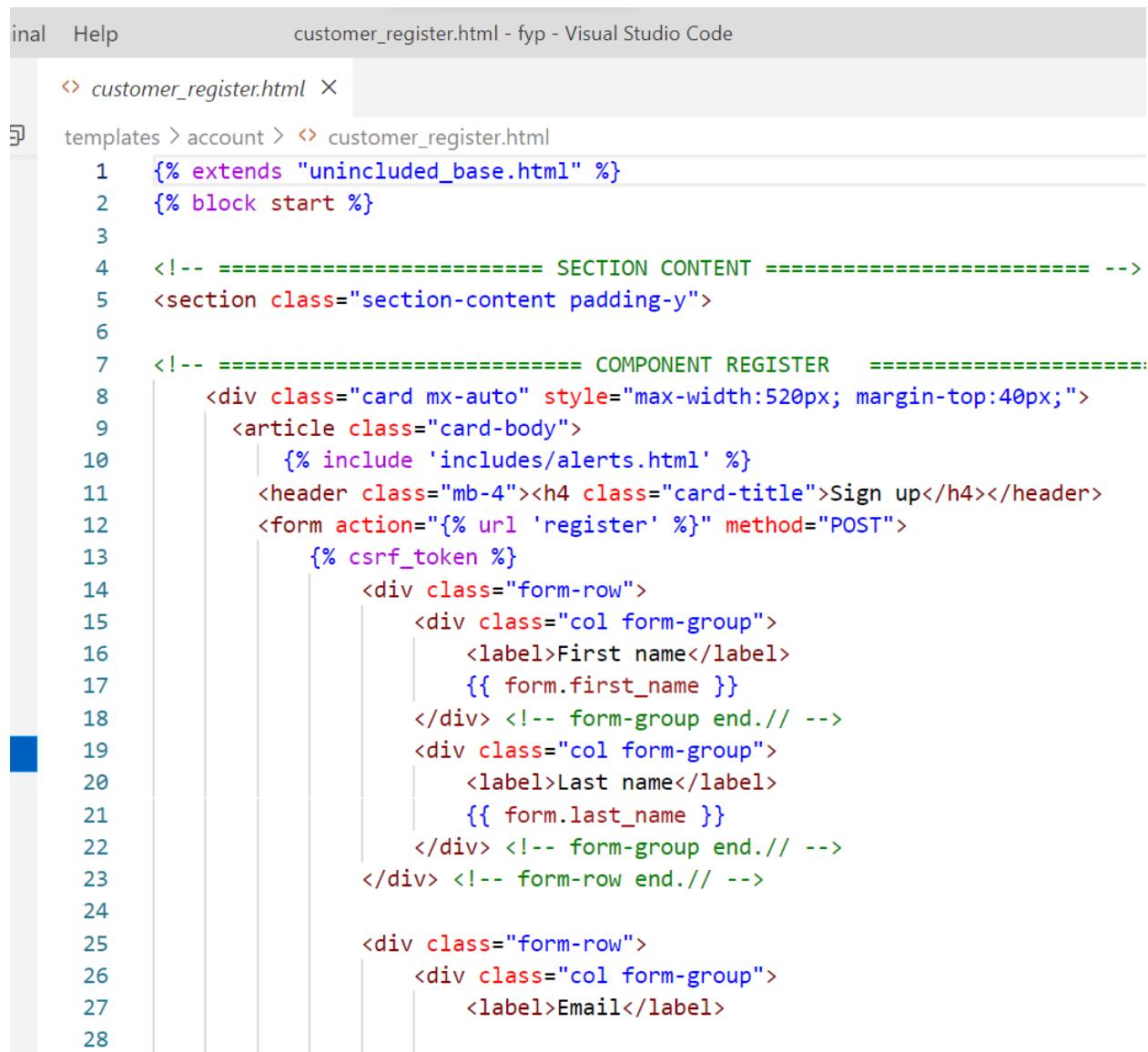
A screenshot of the Visual Studio Code interface. The title bar shows "city\_dropdown\_list\_options.html - fyp - Visual Studio Code". Below the title bar, the file path "templates > account > city\_dropdown\_list\_options.html" is displayed. The main editor area contains the following code:

```
1 <option value="">Select City</option>
2 {% for city in cities %}
3   <option value="{{ city.pk }}">{{ city.name }}</option>
4 {% endfor %}
```

```
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <link rel="shortcut icon" href="../images/favicon.ico" type="image/x-icon">
8      <meta http-equiv="X-UA-Compatible" content="ie=edge">
9      <title>Guardian</title>
10     <!-- jQuery -->
11     <script src="../js/jquery-2.0.0.min.js" type="text/javascript"></script>
12
13     <!-- Bootstrap4 files-->
14     <script src="../js/bootstrap.bundle.min.js" type="text/javascript"></script>
15     <link href="../css/bootstrap.css" rel="stylesheet" type="text/css"/>
16
17     <!-- Font awesome 5 -->
18     <link href="fonts/fontawesome/css/all.min.css" type="text/css" rel="stylesheet">
19
20     <!-- custom style -->
21     <link href="../css/ui.css" rel="stylesheet" type="text/css"/>
22     <!-- <link href="css/responsive.css" rel="stylesheet" media="only screen and (max-width: 1200px)" -->
23
24     <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/boxicons@latest/css/boxicons.min.css"/>
25     <link rel="stylesheet" type="text/css" href="https://cdn.jsdelivr.net/npm/slick-carousel@1.8.1/slick/slick.css"/>
26
27     <!-- custom javascript -->
28     <script src="../js/script.js" type="text/javascript"></script>
29
30     <script type="text/javascript">
31         // some script

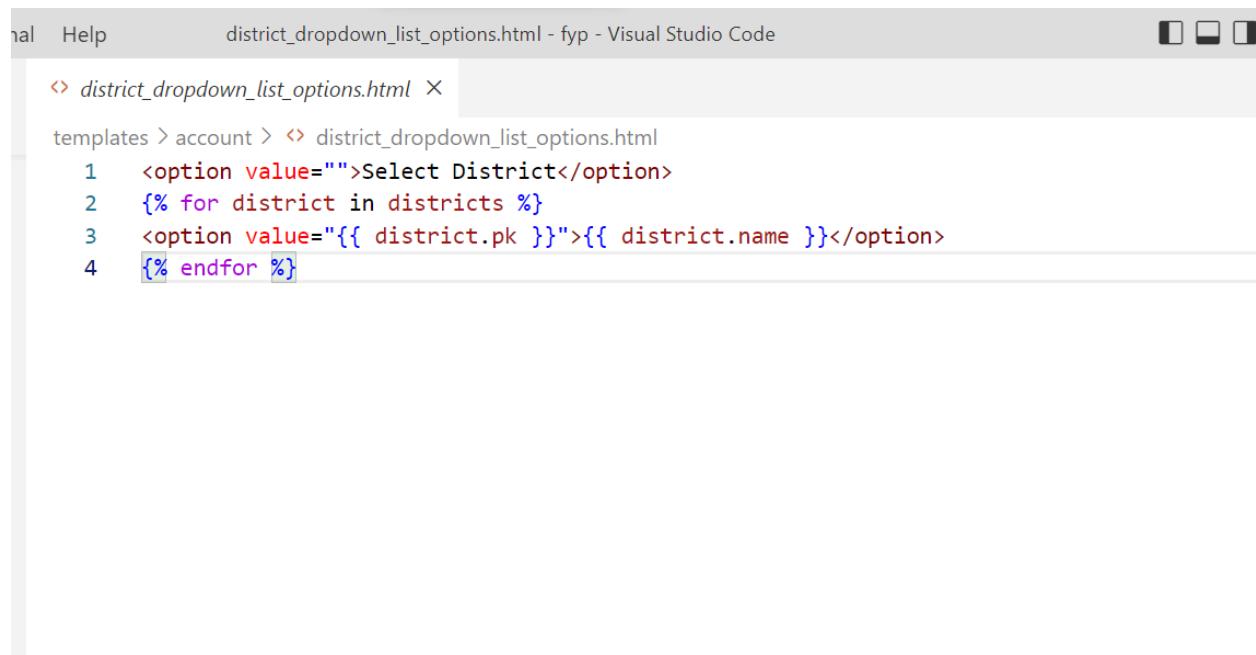
```

```
Help counsellorregister.html - fyp - Visual Studio Code
<> counsellorregister.html X
templates > account > <> counsellorregister.html
1  {% extends "unincluded_base.html" %} 
2  {% block start %}
3      <style>
4          textarea{
5              overflow: auto;
6              resize: vertical;
7              display: block;
8              width: 100%;
9              height: calc(1.5em + 0.9rem + 5rem);
10             padding: 0.45rem 0.85rem;
11             font-size: 1rem;
12             font-weight: 400;
13             line-height: 1.5;
14             color: #495057;
15             background-color: #fff;
16             background-clip: padding-box;
17             border: 1px solid #ced4da;
18             border-radius: 0.37rem;
19             transition: border-color 0.15s ease-in-out, box-shadow 0.15s ease-in-out;
20         }
21     </style>
22     <!-- ===== SECTION CONTENT ===== -->
23     <section class="section-content padding-y bg">
24         <div class="container">
25             <br>
26             <br>
27             <div class="row align-items-center">
28                 <article class="col-md-1">
29                     </article>
```



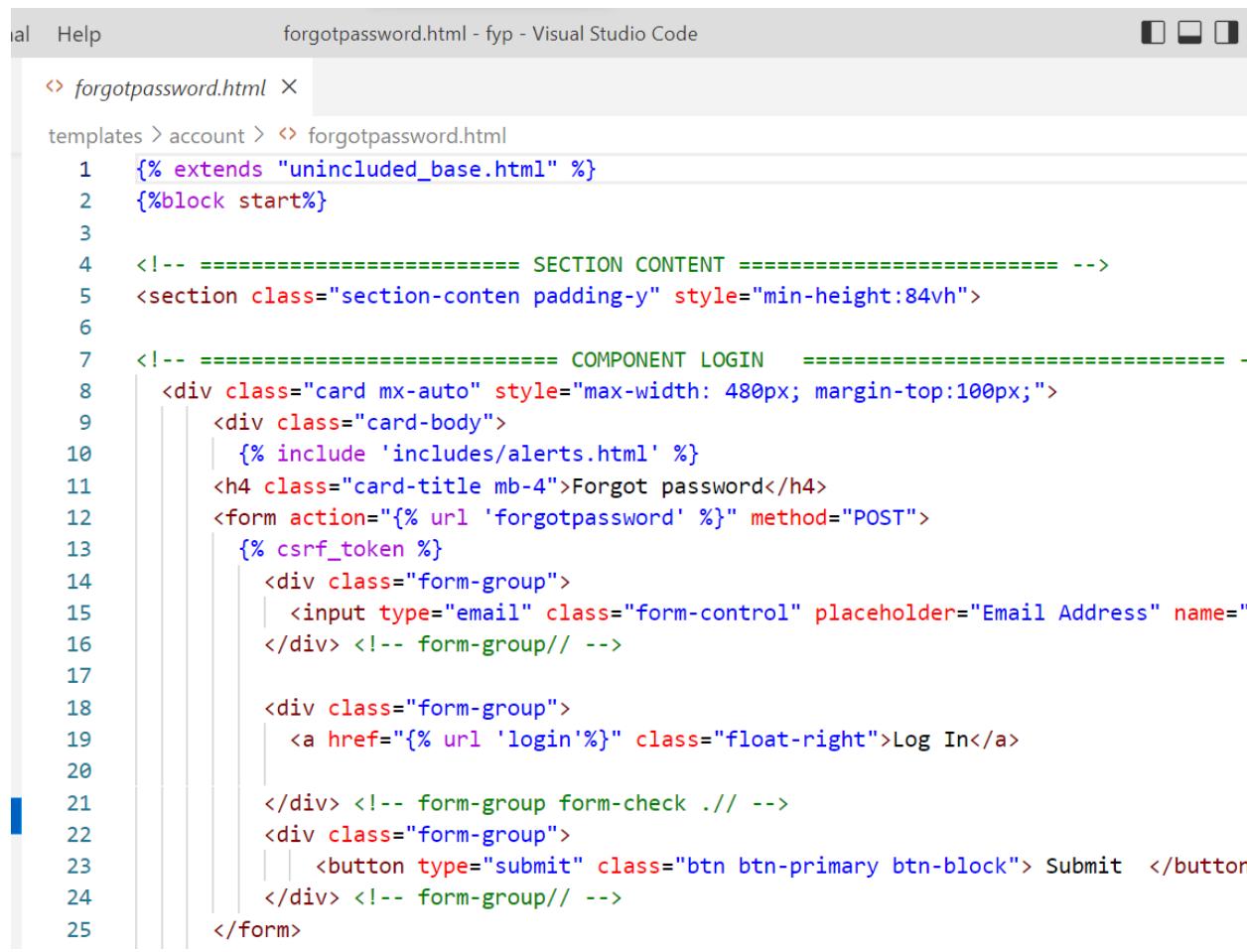
The screenshot shows a Visual Studio Code window with the title bar "customer\_register.html - fyp - Visual Studio Code". The file path in the sidebar is "templates > account > customer\_register.html". The code itself is a Jinja template for a sign-up form. It starts with an extends statement to "unincluded\_base.html" and defines a block start. The template then includes a section for "SECTION CONTENT" and a component for "COMPONENT REGISTER". It features a card with a header "Sign up", a form with CSRF token, and fields for first name, last name, and email.

```
1  {% extends "unincluded_base.html" %}  
2  {% block start %}  
3  
4  <!-- ===== SECTION CONTENT ===== -->  
5  <section class="section-content padding-y">  
6  
7  <!-- ===== COMPONENT REGISTER ===== -->  
8  <div class="card mx-auto" style="max-width:520px; margin-top:40px;">  
9    <article class="card-body">  
10      {% include 'includes/alerts.html' %}  
11      <header class="mb-4"><h4 class="card-title">Sign up</h4></header>  
12      <form action="{% url 'register' %}" method="POST">  
13        {% csrf_token %}  
14        <div class="form-row">  
15          <div class="col form-group">  
16            <label>First name</label>  
17            {{ form.first_name }}  
18          </div> <!-- form-group end.-->  
19          <div class="col form-group">  
20            <label>Last name</label>  
21            {{ form.last_name }}  
22          </div> <!-- form-group end.-->  
23        </div> <!-- form-row end.-->  
24  
25        <div class="form-row">  
26          <div class="col form-group">  
27            <label>Email</label>
```



A screenshot of Visual Studio Code showing the file `district_dropdown_list_options.html`. The code is a Django template snippet:

```
<option value="">Select District</option>
{% for district in districts %}
<option value="{{ district.pk }}">{{ district.name }}</option>
{% endfor %}
```



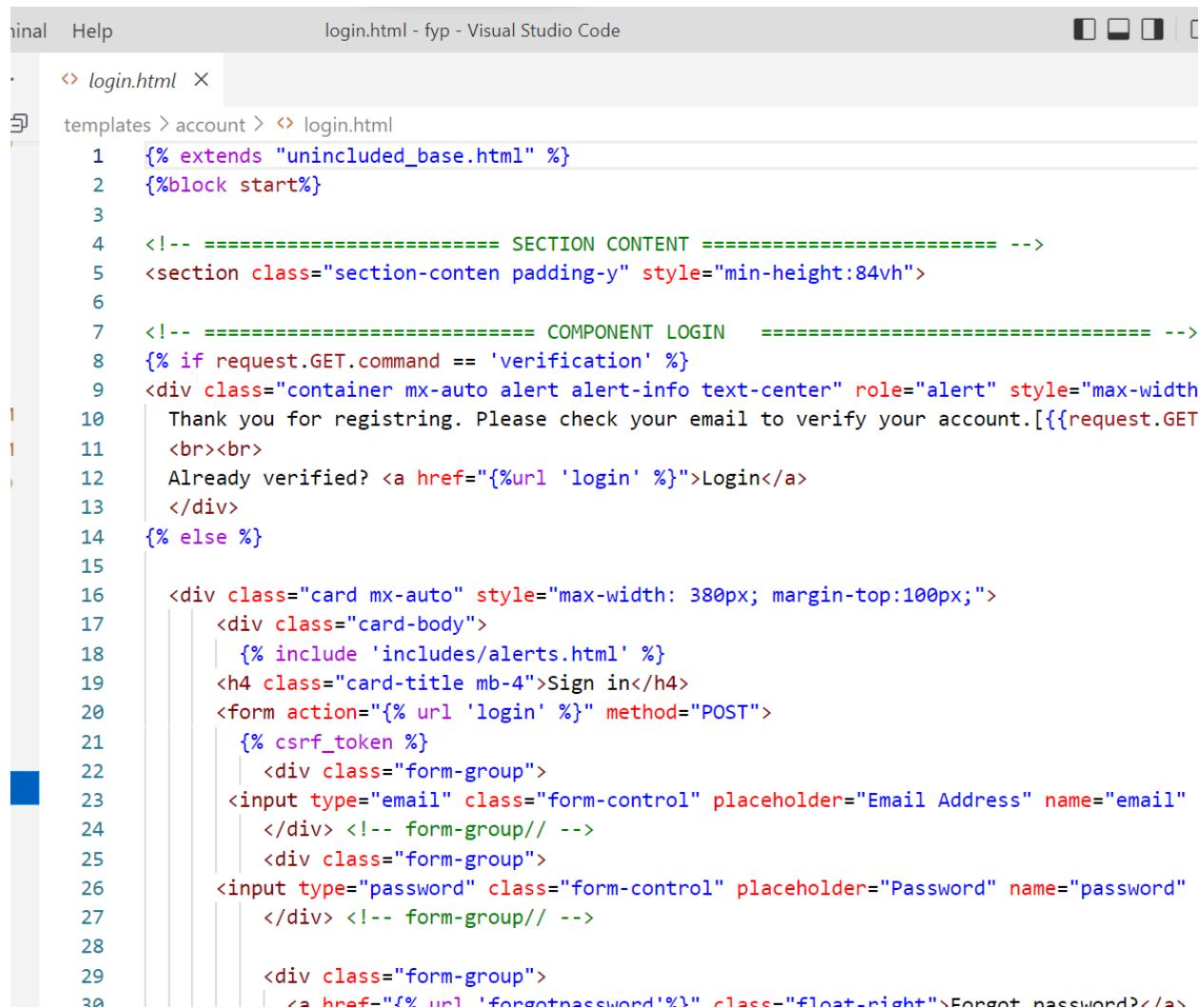
A screenshot of Visual Studio Code showing the file `forgotpassword.html`. The code is a Django template snippet:

```
{% extends "unincluded_base.html" %}
{%block start%}

<section class="section-content padding-y" style="min-height:84vh">

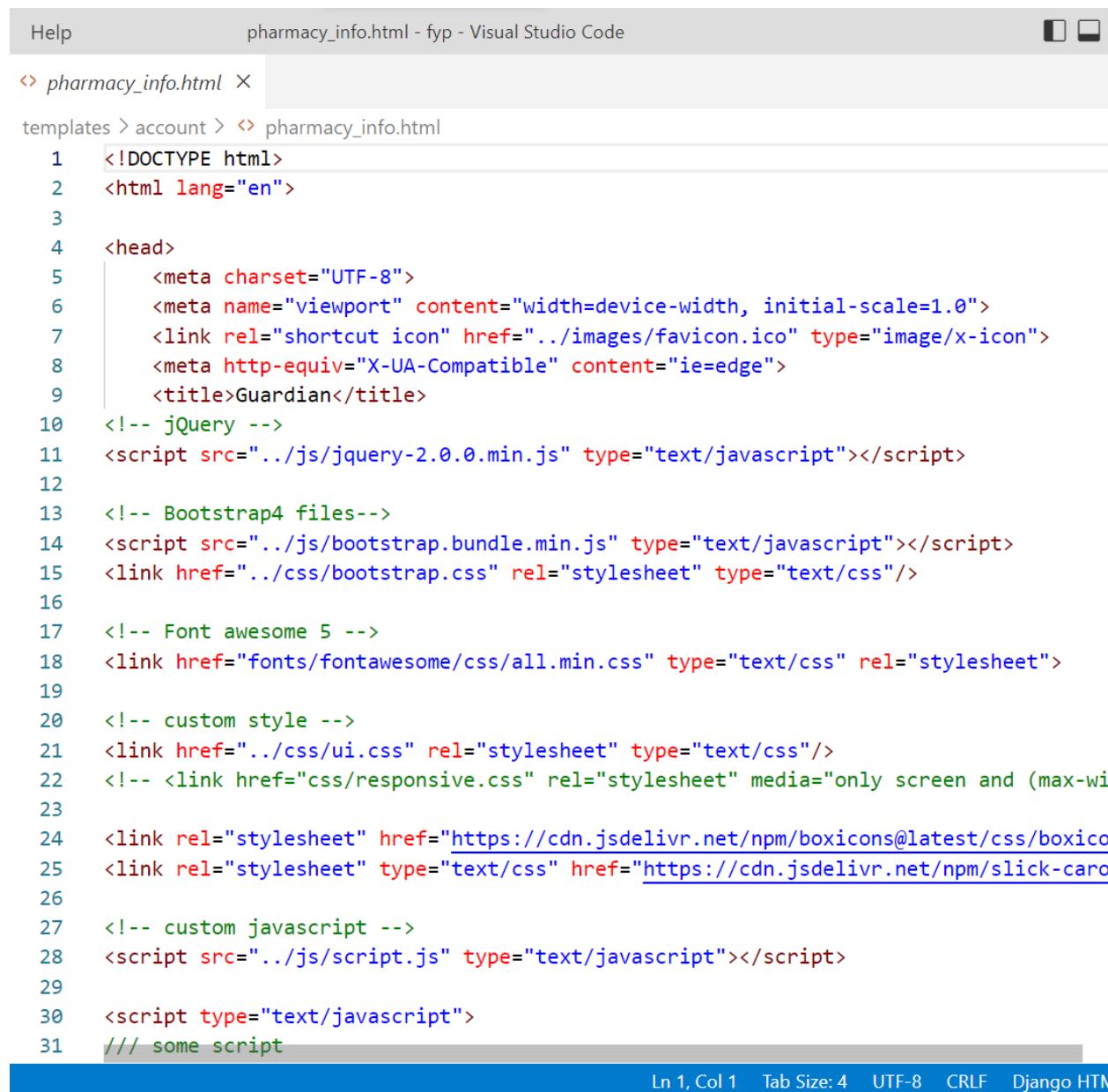
<!-- ===== COMPONENT LOGIN =====
&lt;div class="card mx-auto" style="max-width: 480px; margin-top:100px;"&gt;
    &lt;div class="card-body"&gt;
        {% include 'includes/alerts.html' %}
        &lt;h4 class="card-title mb-4"&gt;Forgot password&lt;/h4&gt;
        &lt;form action="{% url 'forgotpassword' %}" method="POST"&gt;
            {% csrf_token %}
            &lt;div class="form-group"&gt;
                &lt;input type="email" class="form-control" placeholder="Email Address" name="email"&gt;
            &lt;/div&gt; &lt;!-- form-group// --&gt;

            &lt;div class="form-group"&gt;
                &lt;a href="{% url 'login'%}" class="float-right"&gt;Log In&lt;/a&gt;
            &lt;/div&gt; &lt;!-- form-group form-check .// --&gt;
            &lt;div class="form-group"&gt;
                &lt;button type="submit" class="btn btn-primary btn-block"&gt; Submit &lt;/button&gt;
            &lt;/div&gt; &lt;!-- form-group// --&gt;
        &lt;/form&gt;</pre>
```



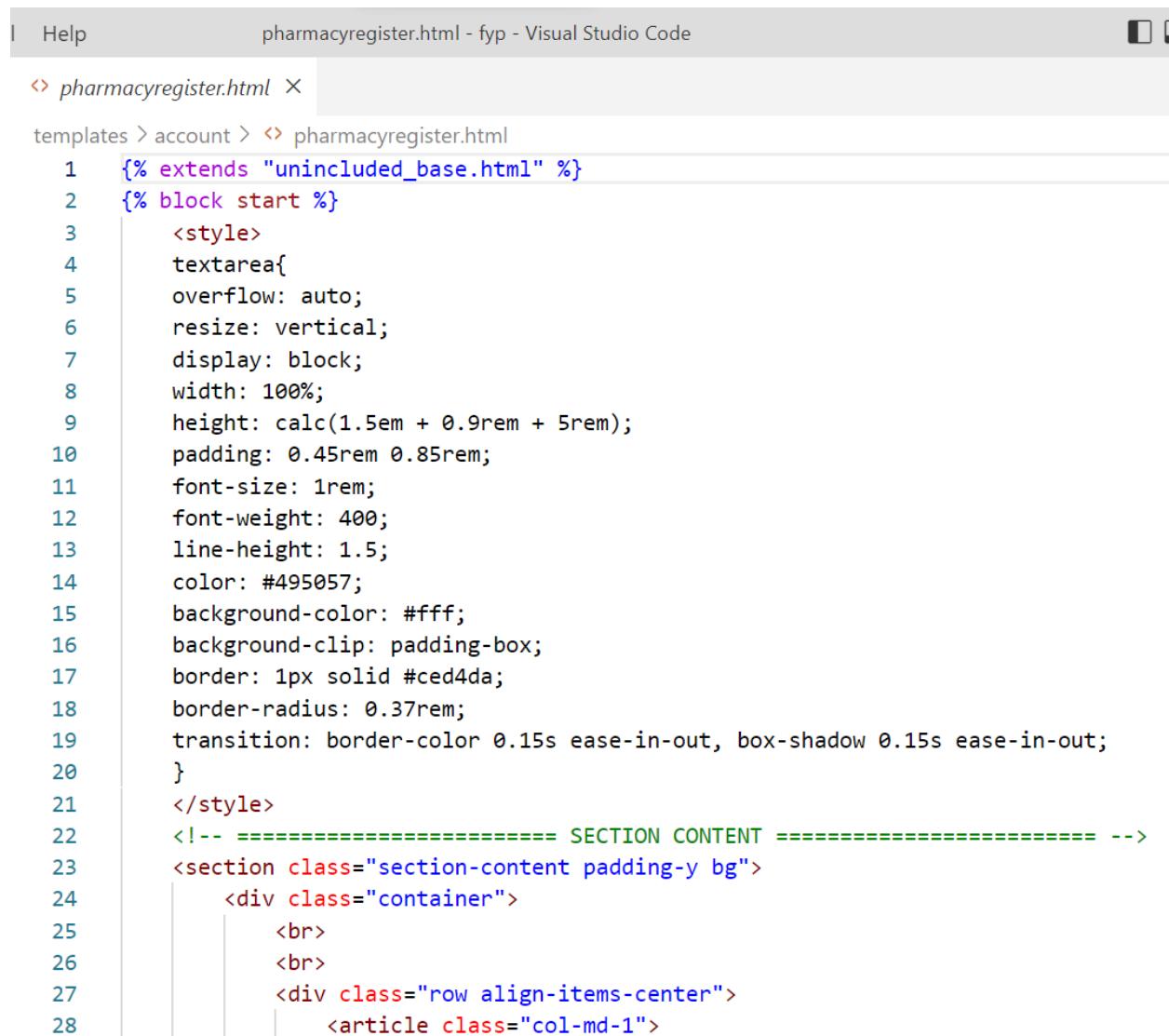
The screenshot shows a Visual Studio Code window with the title "login.html - fyp - Visual Studio Code". The code editor displays the "login.html" file, which is a template for a login page. The file uses Jinja2 syntax with blocks and includes. It includes sections for account verification and a sign-in form. The code is well-formatted with line numbers and color-coded syntax highlighting.

```
1  {% extends "unincluded_base.html" %}          login.html
2  {%block start%}
3
4  <!-- ===== SECTION CONTENT ===== -->
5  <section class="section-content padding-y" style="min-height:84vh">
6
7  <!-- ===== COMPONENT LOGIN ===== -->
8  {% if request.GET.command == 'verification' %}
9  <div class="container mx-auto alert alert-info text-center" role="alert" style="max-width:
10     Thank you for registering. Please check your email to verify your account.[{{request.GET
11     <br><br>
12     Already verified? <a href="{%url 'login' %}">Login</a>
13     </div>
14     {% else %}
15
16     <div class="card mx-auto" style="max-width: 380px; margin-top:100px;">
17       <div class="card-body">
18         {% include 'includes/alerts.html' %}
19         <h4 class="card-title mb-4">Sign in</h4>
20         <form action="{% url 'login' %}" method="POST">
21           {% csrf_token %}
22             <div class="form-group">
23               <input type="email" class="form-control" placeholder="Email Address" name="email"
24               </div> <!-- form-group// -->
25               <div class="form-group">
26                 <input type="password" class="form-control" placeholder="Password" name="password"
27                 </div> <!-- form-group// -->
28
29               <div class="form-group">
30                 <a href="{% url 'forgetpassword' %}" class="float-right">Forgot password?</a>
```



The screenshot shows a Visual Studio Code interface with the following details:

- Title Bar:** Help | pharmacy\_info.html - fyp - Visual Studio Code
- File Path:** templates > account > pharmacy\_info.html
- Code Content:** The code is an HTML file with various meta tags, scripts, and links. It includes jQuery, Bootstrap4 files, Font awesome 5, custom styles, and external CSS and JS files from cdn.jsdelivr.net.
- Code Lines:** Numbered from 1 to 31, showing the structure of the HTML document.
- Status Bar:** Ln 1, Col 1 | Tab Size: 4 | UTF-8 | CRLF | Django HTM

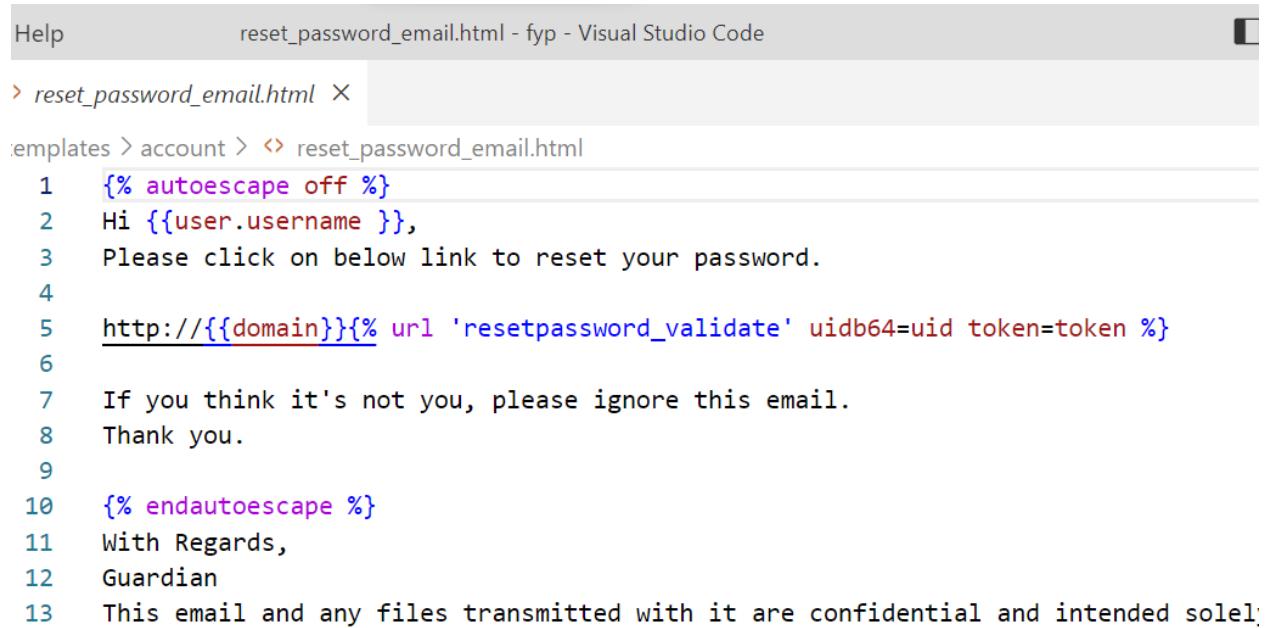


```
pharmacyregister.html - fyp - Visual Studio Code

<pharmacyregister.html>

templates > account > <pharmacyregister.html>

1  {% extends "unincluded_base.html" %} 
2  {% block start %} 
3      <style>
4          textarearea{
5              overflow: auto;
6              resize: vertical;
7              display: block;
8              width: 100%;
9              height: calc(1.5em + 0.9rem + 5rem);
10             padding: 0.45rem 0.85rem;
11             font-size: 1rem;
12             font-weight: 400;
13             line-height: 1.5;
14             color: #495057;
15             background-color: #fff;
16             background-clip: padding-box;
17             border: 1px solid #ced4da;
18             border-radius: 0.37rem;
19             transition: border-color 0.15s ease-in-out, box-shadow 0.15s ease-in-out;
20         }
21     </style>
22     <!-- ===== SECTION CONTENT ===== -->
23     <section class="section-content padding-y bg">
24         <div class="container">
25             <br>
26             <br>
27             <div class="row align-items-center">
28                 <article class="col-md-1">
```



The screenshot shows a Visual Studio Code window with the title bar "Help" and "reset\_password\_email.html - fyp - Visual Studio Code". The file path "templates > account > reset\_password\_email.html" is visible in the top left. The code editor contains the following content:

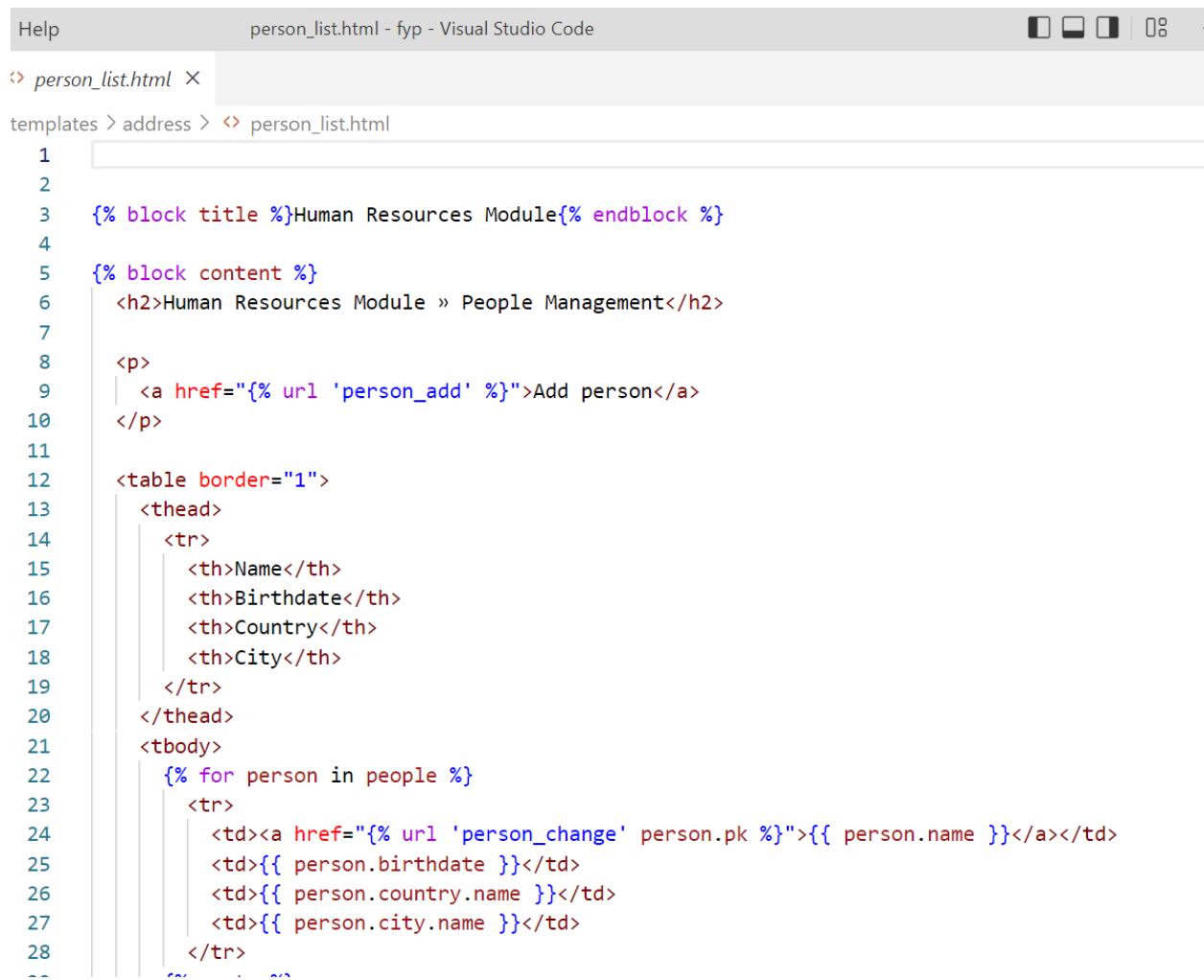
```
1  {% autoescape off %}  
2  Hi {{user.username }},  
3  Please click on below link to reset your password.  
4  
5  http://{{domain}}{% url 'resetpassword\_validate' uidb64=uid token=token %}  
6  
7  If you think it's not you, please ignore this email.  
8  Thank you.  
9  
10 {% endautoescape %}  
11 With Regards,  
12 Guardian  
13 This email and any files transmitted with it are confidential and intended solel
```

```
terminal Help             resetpassword.html - fyp - Visual Studio Code
...  ⌂ resetpassword.html ×
templates > account > ⌂ resetpassword.html
1  {% extends "unincluded_base.html" %}           M
2  {%block start%}                                M
3
4  <!-- ===== SECTION CONTENT ===== -->          ●
5  <section class="section-content padding-y" style="min-height:84vh">
6
7  <!-- ===== COMPONENT LOGIN ===== -->
8  <div class="card mx-auto" style="max-width: 380px; margin-top:100px;">
9    <div class="card-body">
10   |   {% include 'includes/alerts.html' %}
11   |   <h4 class="card-title mb-4">Reset Password</h4>
12   |   <form action="{% url 'resetpassword' %}" method="POST">
13   |     |   {% csrf_token %}
14   |     |   <div class="form-group">
15   |     |     <input type="password" class="form-control" placeholder="Password" name="p" style="width: 100%; height: 40px; border-radius: 5px; border: 1px solid #ccc; padding: 5px; margin-bottom: 10px; font-size: 14px; font-weight: bold; color: #333; transition: border-color 0.3s ease;"/>
16   |     |     |   <!-- form-group// -->
17   |     |     <div class="form-group">
18   |     |       <input type="password" class="form-control" placeholder="Conform Password" style="width: 100%; height: 40px; border-radius: 5px; border: 1px solid #ccc; padding: 5px; margin-bottom: 10px; font-size: 14px; font-weight: bold; color: #333; transition: border-color 0.3s ease;"/>
19   |     |       |   <!-- form-group// -->
20
21  |   <!-- form-group form-check //-->
22  |   |   <div class="form-group">
23  |   |     |   <button type="submit" class="btn btn-primary btn-block"> Reset </b>
24  |   |     |   </div> <!-- form-group// -->
25  |   |   </form>
26  |   |   </div> <!-- card-body// -->
27  |   |   </div> <!-- card .// -->
28  |   <!-- ===== COMPONENT LOGIN END.// ===== -->
29
30
31  </section>
```

A screenshot of Visual Studio Code showing the file `person_form.html`. The code is a Django template for a person form. It includes an `<h2>Person Form</h2>`, a `<form>` block with `method="post"`, `id="personForm"`, and `data-cities-url="{% url 'ajax_load_cities' %}"`. The form contains a csrf token, a table with form data, a save button, and a link to the person changelist. Below the form is a script that uses jQuery to handle a change event on the country select dropdown. The script sends an AJAX request to the specified URL with the selected country ID as data. The success callback updates the city dropdown with the returned data.

```
1  final Help person_form.html - fyp - Visual Studio Code
2  <!-- person_form.html -->
3  templates > address > <!-- person_form.html -->
4
5  1  {% block content %}
6  2    <h2>Person Form</h2>
7  3
8  4    <form method="post" id="personForm" data-cities-url="{% url 'ajax_load_cities' %}" novalidate>
9  5      {% csrf_token %}
10  6      <table>
11  7        {{ form.as_table }}
12  8      </table>
13  9      <button type="submit">Save</button>
14 10      <a href="{% url 'person_changelist' %}">Nevermind</a>
15 11    </form>
16 12
17 13    <script src="https://code.jquery.com/jquery-3.3.1.min.js"></script>
18 14    <script>
19 15      $("#id_country").change(function () {
20 16        var url = $("#personForm").attr("data-cities-url");
21 17        var countryId = $(this).val();
22 18
23 19        $.ajax({
24 20          url: url,
25 21          data: {
26 22            'country': countryId
27 23          },
28 24          success: function (data) {
29 25            $("#id_city").html(data);
30 26          }
31 27        });
32 28      });
33 29    </script>
34 30
35 31  
```

Ln 1, Col 1 Spaces: 2 UTF-8 CRLF Django HTML kite: ready



The screenshot shows a Visual Studio Code interface with the following details:

- Title Bar:** Help | person\_list.html - fyp - Visual Studio Code | Close, Minimize, Maximize, Exit
- File Path:** templates > address > person\_list.html
- Code Content:** A Django template named person\_list.html. The code uses Python-like syntax for blocks and variables. It includes a title block, a content block with an h2 header and a link to add a person, and a table block displaying a list of people with columns for Name, Birthdate, Country, and City. Each row contains a link to change the person's information.

```
1
2
3  {% block title %}Human Resources Module{% endblock %}
4
5  {% block content %}
6      <h2>Human Resources Module » People Management</h2>
7
8      <p>
9          <a href="{% url 'person_add' %}">Add person</a>
10
11     <table border="1">
12         <thead>
13             <tr>
14                 <th>Name</th>
15                 <th>Birthdate</th>
16                 <th>Country</th>
17                 <th>City</th>
18             </tr>
19         </thead>
20         <tbody>
21             {% for person in people %}
22                 <tr>
23                     <td><a href="{% url 'person_change' person.pk %}">{{ person.name }}</a></td>
24                     <td>{{ person.birthdate }}</td>
25                     <td>{{ person.country.name }}</td>
26                     <td>{{ person.city.name }}</td>
27                 </tr>
28             {% endfor %}
29         </tbody>
30     </table>
31 
```



The screenshot shows a Visual Studio Code interface with a single tab open for 'base\_site.html'. The code editor displays the following Python template code:

```
1  {% extends "admin/base_site.html" %}  
2  
3  {% load static %}  
4  
5  {% block extrahead %}  
6  |    <link rel="shortcut icon" href="{% static 'images/favicon.ico' %}" />  
7  {% endblock %}
```

```
Help          blog_detail.html - fyp - Visual Studio Code

blog_detail.html ×

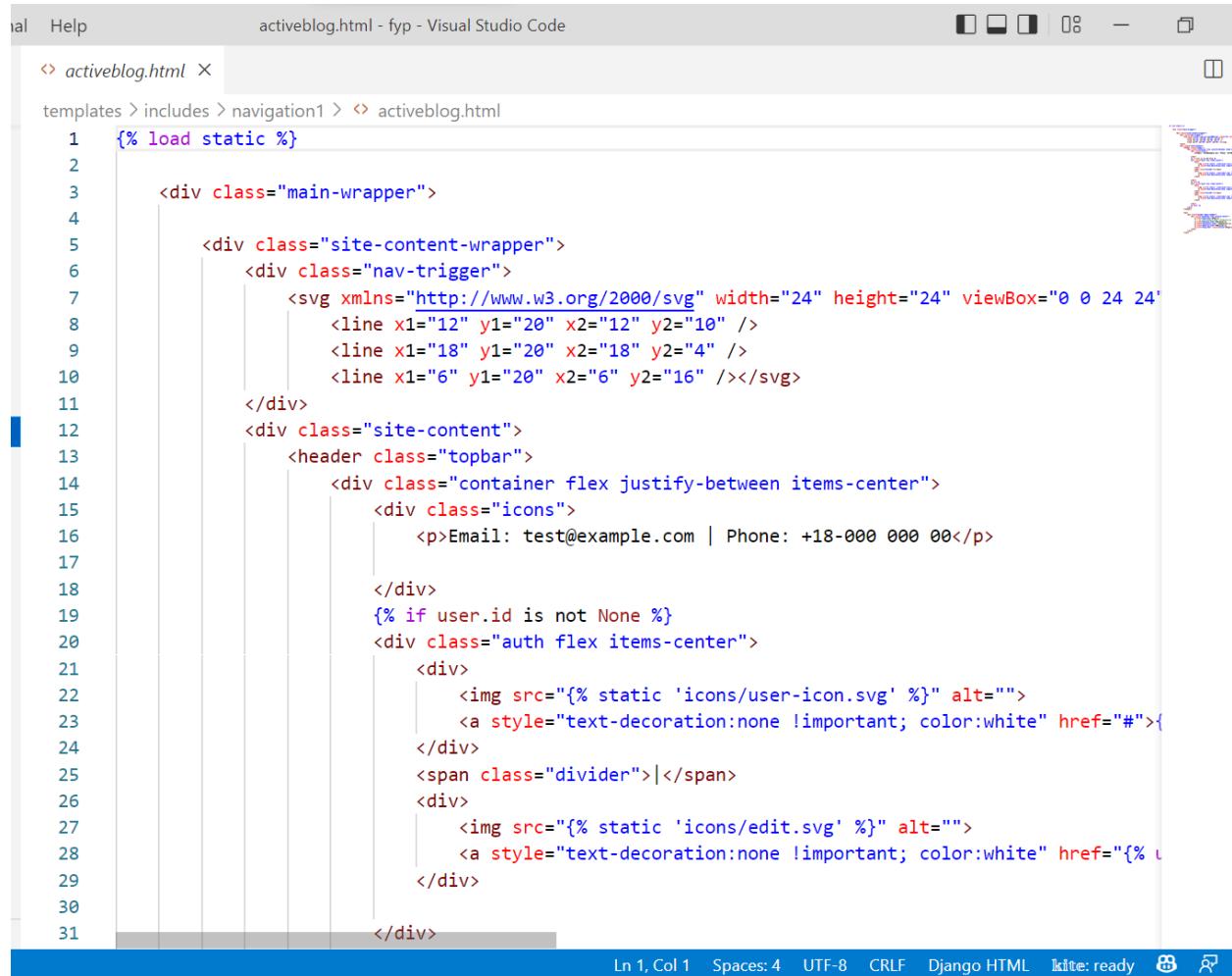
templates > blog > blog_detail.html
1  {% extends "base2.html" %} 
2  {% load static %} 
3  {% block start %} 
4      <section>
5          <div class="container">
6              <div class="card mb-3">
7                  
9                      <div class="text-center">
10                         <h5 class="card-title">{{ blog_obj.title }}<
11                         </div>
12                         <div class="d-flex justify-content-around">
13                             <div>Author: {{ blog_obj.counsellor_name_id }}<
14                             <div>Added on: {{ blog_obj.created_at }}</di
15                         </div>
16                         <p class="card-text">
17                             {{ blog_obj.content|safe }}<
18                         </p>
19                         <p class="card-text">
20                             <small class="text-muted">{{ blog_obj.create
21                         </p>
22                         </div>
23                     </div>
24                 </div>
--> ..
```



```
minal Help book_meeting.html - fyp - Visual Studio Code
.. <> book_meeting.html M X
templates > booking > <> book_meeting.html
45     <div class="form-group col-md-12">
46         <label>Description</label>
47         {{ form.meeting_description }}
48         <small class="form-text text-muted" style="color:black!important">
49             {{ form.non_field_errors }}</small>
50     </div>
51     <div class="form-group">
52         {{ form.non_field_errors }}</div>
53
54     <div class="form-group">
55         <button type="submit" value="Submit" class="btn btn-primary btn-block">Submit</button>
56     </div>
57
58     </form>
59 </article>
60 </div>
61 </section>
62
63
64 {% endblock %}
```

The screenshot shows a Visual Studio Code window with the title bar "Help" and "dashboardcounsellor.html - fyp - Visual Studio Code". The file content is a Django template for a dashboard counsellor page. The code includes HTML structure, CSS classes, and Django templating syntax. A comment block is present between lines 6 and 13.

```
1  {% extends "base2.html" %}          dashboardcounsellor.html M X
2  {% load static %}                  templates > counsellor > < dashboardcounsellor.html
3  {% block start %}
4      <section class="section-content padding-y bg">
5          <div class="container">
6              <!-- ===== COMPONENT 1 ===== -->
7              <div class="row">
8                  <aside class="col-lg-3">
9                      <div class="card">
10                         <div class="card-body">
11                             <a href="{% url 'dashboardcounsellor' %}" class="nav-link"
12                             <a href="{% url 'added_blog' %}" class="btn btn-primary">Add Blog</a>
13                         </div>
14                     </div>
15                 </aside>
16                 <aside class="col-lg-9">
17                     <div class="card">
18                         <div class="card-body">
19                             <h4 class="card-title mb-4">Add Blog</h4>
20                             {% include 'includes/alerts.html' %}
21                             <form action="{% url 'dashboardcounsellor' %}" method="POST"
22                                 enctype="multipart/form-data"
23                                 novalidate>
24                                 {% csrf_token %}
25                                 {{ form.media }}
26                                 {{ form.as_p }}
27
28                             <div class="form-group col-sm-3"></div>
29                             <div class="form-group col-sm-2">
```



```
1  {% load static %}

2
3      <div class="main-wrapper">
4
5          <div class="site-content-wrapper">
6              <div class="nav-trigger">
7                  <svg xmlns="http://www.w3.org/2000/svg" width="24" height="24" viewBox="0 0 24 24">
8                      <line x1="12" y1="20" x2="12" y2="10" />
9                      <line x1="18" y1="20" x2="18" y2="4" />
10                     <line x1="6" y1="20" x2="6" y2="16" /></svg>
11             </div>
12             <div class="site-content">
13                 <header class="topbar">
14                     <div class="container flex justify-between items-center">
15                         <div class="icons">
16                             <p>Email: test@example.com | Phone: +18-000 000 00</p>
17
18                         </div>
19                         {% if user.id is not None %}
20                         <div class="auth flex items-center">
21                             <div>
22                                 
23                                 <a style="text-decoration:none !important; color:white" href="#">{
24
25                                     <span class="divider">|</span>
26                                     <div>
27                                         
28                                         <a style="text-decoration:none !important; color:white" href="#">{%
29
30                                     </div>
31             </div>
```

Help activehome.html - fyp - Visual Studio Code

activehome.html X

templates > includes > navigation1 > activehome.html

```
1  {% load static %}

2
3      <div class="main-wrapper">
4
5          <div class="site-content-wrapper">
6              <div class="nav-trigger">
7                  <svg xmlns="http://www.w3.org/2000/svg" width="24" height="24" viewBox="0 0 24 24">
8                      <line x1="12" y1="20" x2="12" y2="10" />
9                      <line x1="18" y1="20" x2="18" y2="4" />
10                     <line x1="6" y1="20" x2="6" y2="16" /></svg>
11      </div>
12      <div class="site-content">
13          <header class="topbar">
14              <div class="container flex justify-between items-center">
15                  <div class="icons">
16                      <p>Email: test@example.com | Phone: +18-000 000 00</p>
17
18                  </div>
19                  {% if user.id is not None %}
20                      <div class="auth flex items-center">
21                          <div>
22                              
23                              <a style="text-decoration:none !important; color:white" href="</a>">
24
25                              <span class="divider">|</span>
26
27                              <div>
28                                  
29                                  <a style="text-decoration:none !important; color:white" href="</a>">
</div>
```

Help activepharmacy.html - fyp - Visual Studio Code

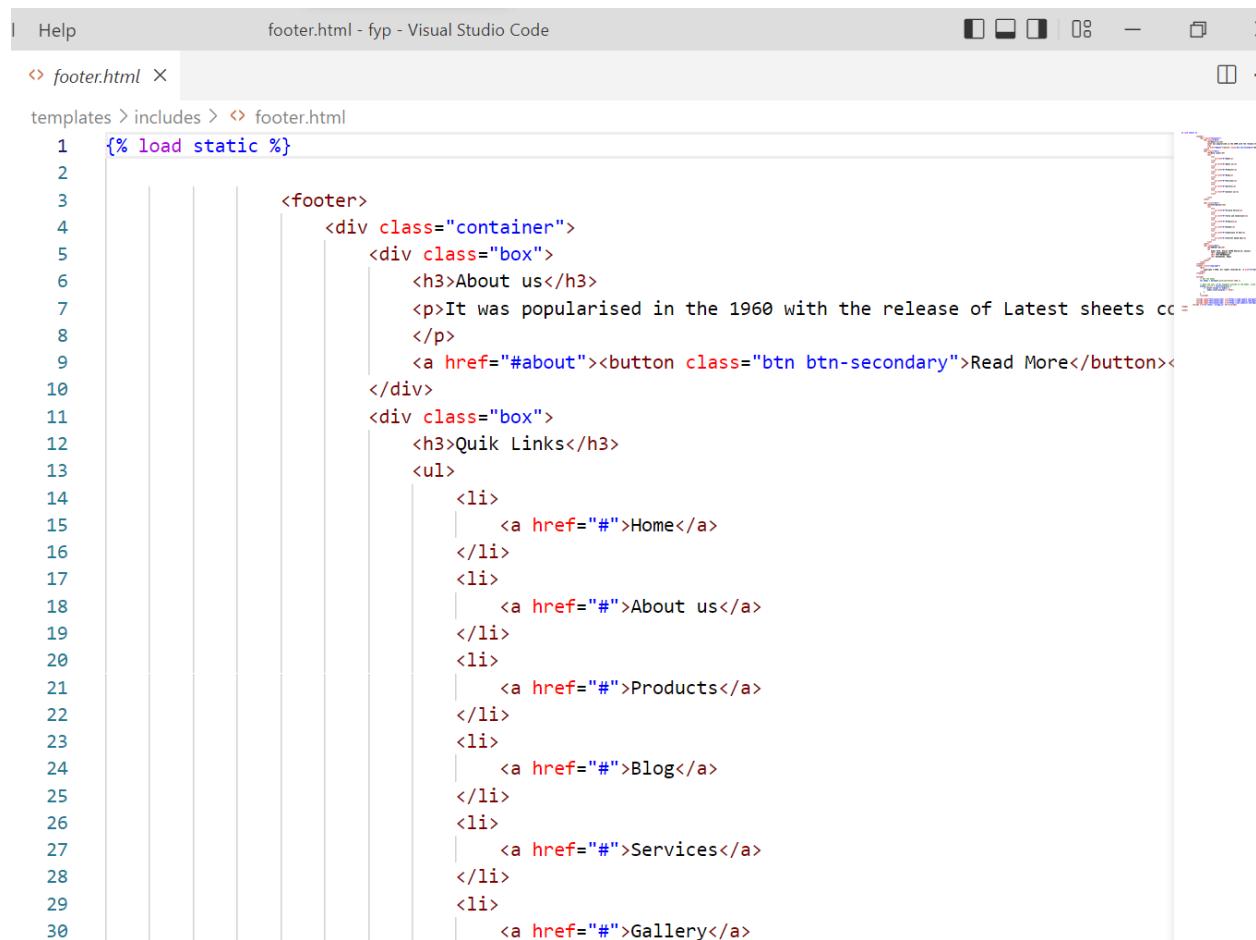
activepharmacy.html

templates > includes > navigation1 > activepharmacy.html

```
1  {% load static %}

2
3      <div class="main-wrapper">
4
5          <div class="site-content-wrapper">
6              <div class="nav-trigger">
7                  <svg xmlns="http://www.w3.org/2000/svg" width="24" height="24" viewBox="0 0 24 24">
8                      <line x1="12" y1="20" x2="12" y2="10" />
9                      <line x1="18" y1="20" x2="18" y2="4" />
10                     <line x1="6" y1="20" x2="6" y2="16" /></svg>
11
12          </div>
13          <div class="site-content">
14              <header class="topbar">
15                  <div class="container flex justify-between items-center">
16                      <div class="icons">
17                          <p>Email: test@example.com | Phone: +18-000 000 00</p>
18
19                      </div>
20                      {% if user.id is not None %}
21                      <div class="auth flex items-center">
22                          <div>
23                              
24                              <a style="text-decoration:none !important; color:white" href="</a>">
25                                  <span class="divider">|</span>
26                              <div>
27                                  
28                                  <a style="text-decoration:none !important; color:white" href="</a>">
29
30                          </div>
31                  </div>
```

Ln 14, Col 5 Spaces: 4 UTF-8 CRLF Django HTML kite: re



```
1  {% load static %}

2
3      <footer>
4          <div class="container">
5              <div class="box">
6                  <h3>About us</h3>
7                  <p>It was popularised in the 1960 with the release of Latest sheets cc =
8                  </p>
9                  <a href="#about"><button class="btn btn-secondary">Read More</button></a>
10             </div>
11             <div class="box">
12                 <h3>Quik Links</h3>
13                 <ul>
14                     <li>
15                         <a href="#">Home</a>
16                     </li>
17                     <li>
18                         <a href="#">About us</a>
19                     </li>
20                     <li>
21                         <a href="#">Products</a>
22                     </li>
23                     <li>
24                         <a href="#">Blog</a>
25                     </li>
26                     <li>
27                         <a href="#">Services</a>
28                     </li>
29                     <li>
30                         <a href="#">Gallery</a>
```



A screenshot of the Visual Studio Code interface. The title bar shows "Help" and "footer2.html - fyp - Visual Studio Code". The file path "templates > includes > footer2.html" is visible above the code editor. The code editor contains the following HTML code:

```
1 <footer class="section-footer border-top padding-y">
2   <div class="container">
3     <p class="float-md-right">
4       Copyright © 2020 .All rights reserved by Gardian
5     </p>
6     <p>
7       <a href="#">Terms and conditions</a>
8     </p>
9   </div><!-- //container -->
10 </footer>
```

Terminal Help navigation2.html - fyp - Visual Studio Code

navigation2.html

templates > includes > navigation2.html

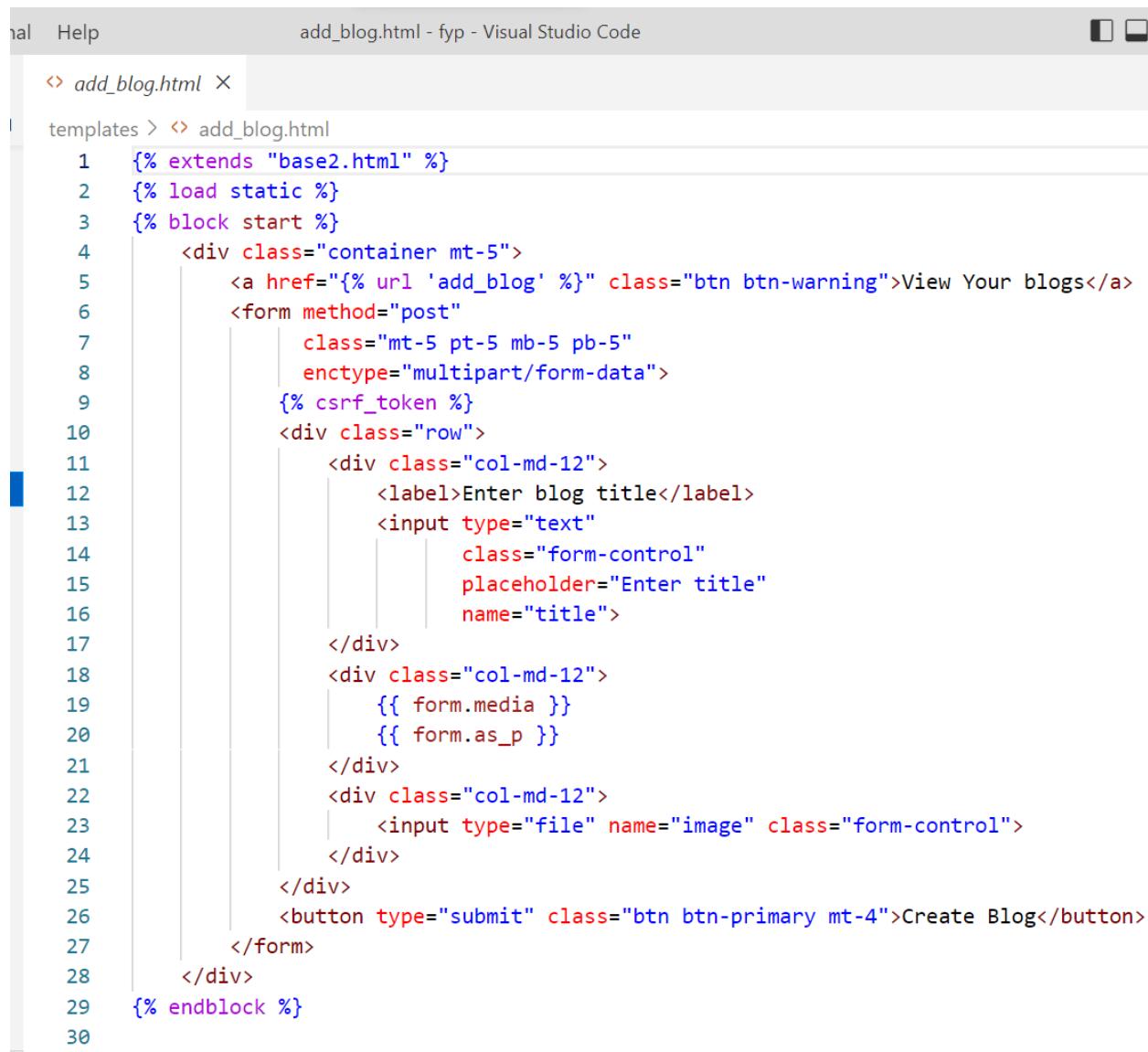
```
1  {% load static %}

2
3      <header class="topbar">
4          <div class="container flex justify-between items-center">
5              <div class="icons">
6                  <p>Email: test@example.com | Phone: +18-000 000 00</p>
7
8              </div>
9              {% if user.id is not None %}
10                 <div class="auth flex items-center">
11                     <div>
12                         
13                         <a href="#"><button> {{user.first_name}} {{user.last_name}}</button></a>
14                     </div>
15                     <span class="divider">|</span>
16                     <div>
17                         
18                         <a href="{% url 'logout' %}"><button>Logout</button></a>
19                     </div>
20
21                 </div>
22
23             {% else %}
24                 <div class="auth flex items-center">
25                     <div>
26                         
27                         <a href="{% url 'login' %}"><button> Log In</button></a>
28                     </div>
29                     <span class="divider">|</span>
30             ...
```

The screenshot shows a Visual Studio Code window with the title bar "added\_product.html - fyp - Visual Studio Code". The file path "templates > pharmacy > added\_product.html" is visible in the top left. The code editor contains a Django template with the following content:

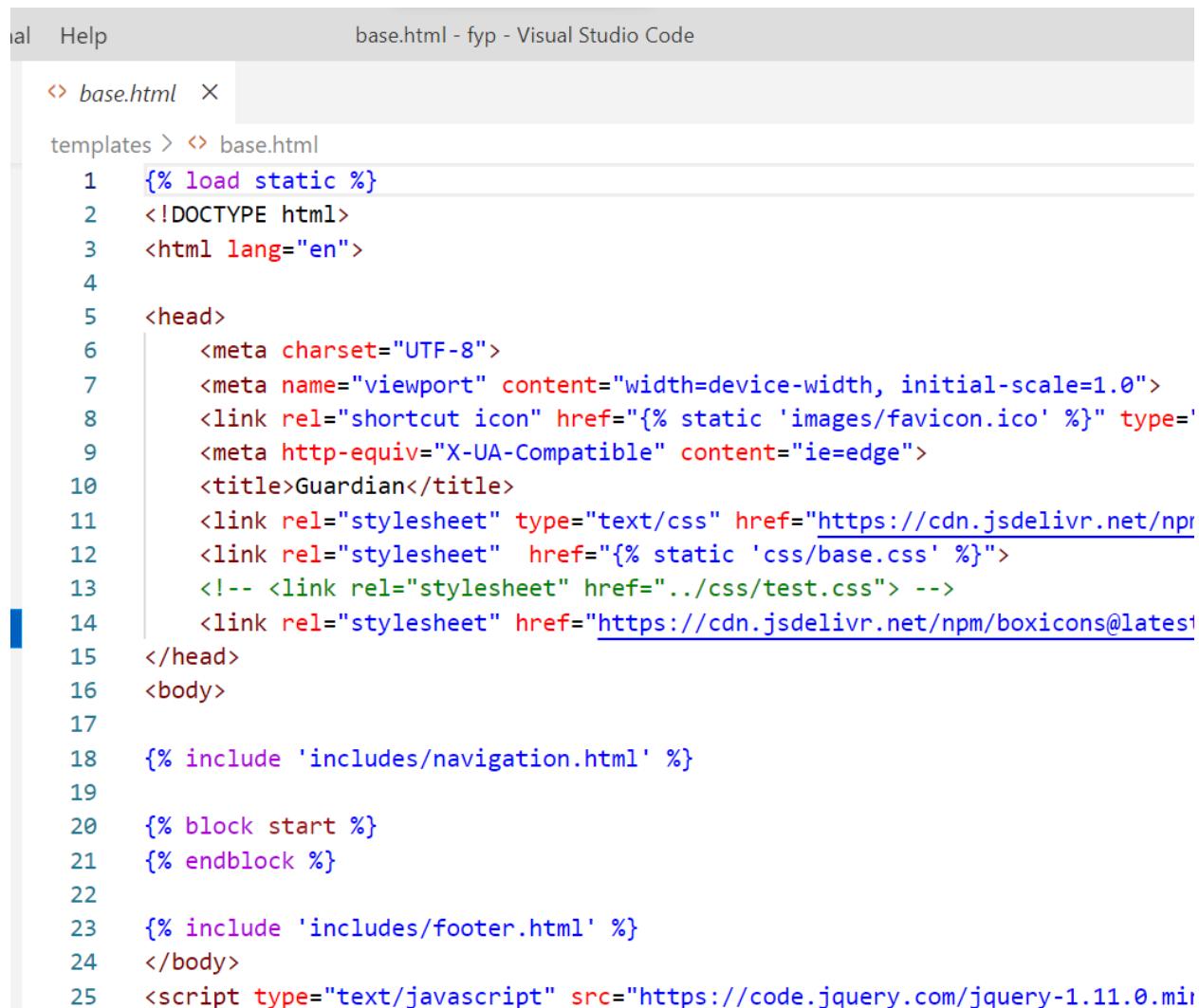
```
1  {% extends "base2.html" %}          added_product.html
2  {% load static %}                  fyp - Visual Studio Code
3  {% block start %}                Help
4
5  {% comment %} <script type="text/javascript">
6      $(document).on('click', '.delete', function(){
7          $id = $(this).attr('name');
8          $.ajax({
9              url: 'delete/' + $id,
10             type: 'POST',
11             data: {
12                 csrfmiddlewaretoken: $('input[name=csrfmiddlewaretoken]').val()
13             },
14             success: function(){
15                 Read();
16                 alert("The task has been DELETED!")
17             }
18         });
19     });
20
21 </script> {% endcomment %}
22
23
24 {% csrf_token %}
25 <section class="section-content padding-y bg">
26     <div class="container">
27
28     <!-- ===== COMPONENT 1 ===== -->
29
30     <div class="row">
31         <aside class="col-lg-9">
```

The status bar at the bottom right shows "Ln 1, Col 1" and "Spaces: 4" and "UTF-8" and "CRLF" and "Django".



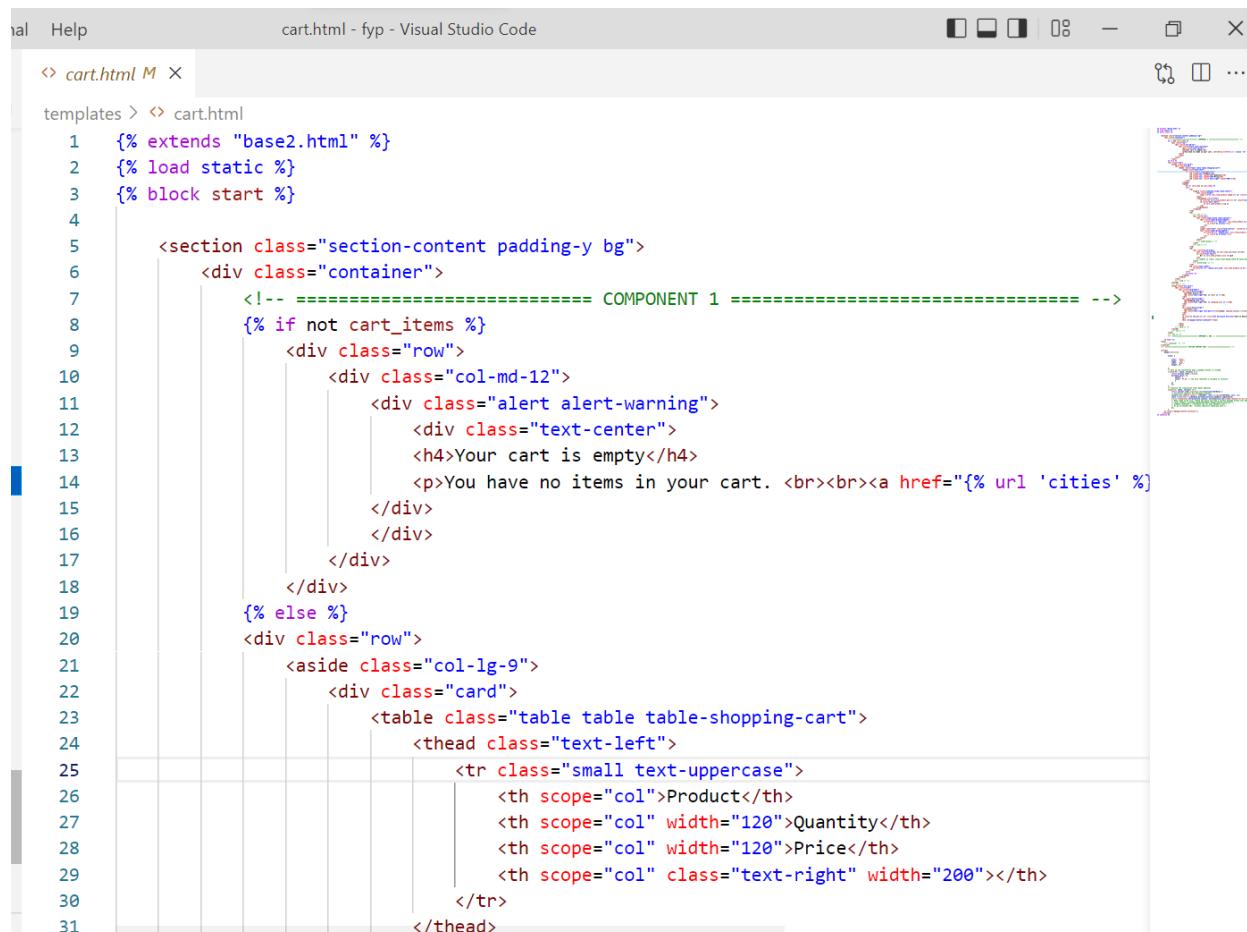
The screenshot shows a Visual Studio Code window with the file "add\_blog.html" open. The code is a Django template for adding a blog post. It includes HTML structure for a container, navigation links, a form for title and media input, and a file input for an image. The code uses Bootstrap classes for styling.

```
1  {% extends "base2.html" %} 2  {% load static %} 3  {% block start %} 4      <div class="container mt-5"> 5          <a href="{% url 'add_blog' %}" class="btn btn-warning">View Your blogs</a> 6          <form method="post" 7              class="mt-5 pt-5 mb-5 pb-5" 8              enctype="multipart/form-data"> 9              {% csrf_token %} 10             <div class="row"> 11                 <div class="col-md-12"> 12                     <label>Enter blog title</label> 13                     <input type="text" 14                         class="form-control" 15                         placeholder="Enter title" 16                         name="title"> 17                 </div> 18                 <div class="col-md-12"> 19                     {{ form.media }} 20                     {{ form.as_p }} 21                 </div> 22                 <div class="col-md-12"> 23                     <input type="file" name="image" class="form-control"> 24                 </div> 25             </div> 26             <button type="submit" class="btn btn-primary mt-4">Create Blog</button> 27         </form> 28     </div> 29  {% endblock %} 30
```



The screenshot shows a Visual Studio Code interface with the title bar "base.html - fyp - Visual Studio Code". The main editor area displays the content of the "base.html" file. The code is written in HTML and includes some Jinja2 templating syntax. It sets up the document's head with meta tags for charset, viewport, and various links to CSS and JavaScript files. It also includes a block for navigation and footer inclusion.

```
1  {% load static %}          base.html - fyp - Visual Studio Code
2  <!DOCTYPE html>
3  <html lang="en">
4
5  <head>
6      <meta charset="UTF-8">
7      <meta name="viewport" content="width=device-width, initial-scale=1.0">
8      <link rel="shortcut icon" href="{% static 'images/favicon.ico' %}" type='
9      <meta http-equiv="X-UA-Compatible" content="ie=edge">
10     <title>Guardian</title>
11     <link rel="stylesheet" type="text/css" href="https://cdn.jsdelivr.net/npm/
12     <link rel="stylesheet" href="{% static 'css/base.css' %}">
13     <!-- <link rel="stylesheet" href="../css/test.css" -->
14     <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/boxicons@lates
15 </head>
16 <body>
17
18     {% include 'includes/navigation.html' %}
19
20     {% block start %}
21     {% endblock %}
22
23     {% include 'includes/footer.html' %}
24 </body>
25 <script type="text/javascript" src="https://code.jquery.com/jquery-1.11.0.mi
```



A screenshot of Visual Studio Code showing the file `cart.html`. The code is a Django template for a shopping cart page. It includes logic to check if the cart is empty and displays a message and a table for items. The code uses Bootstrap classes for styling.

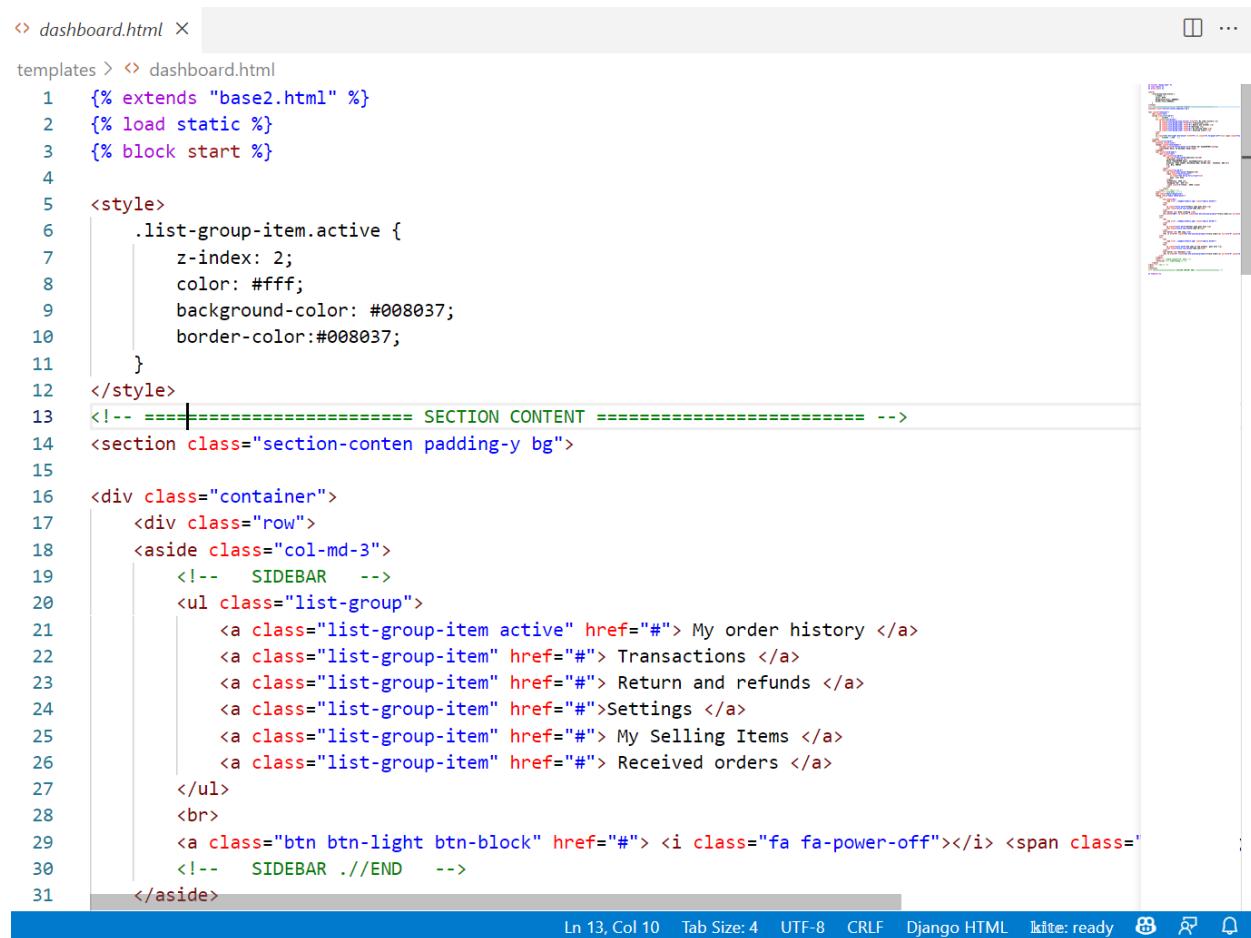
```
1  {% extends "base2.html" %}\n2  {% load static %}\n3  {% block start %}\n4\n5      <section class="section-content padding-y bg">\n6          <div class="container">\n7              <!-- ===== COMPONENT 1 ===== -->\n8              {% if not cart_items %}\n9                  <div class="row">\n10                      <div class="col-md-12">\n11                          <div class="alert alert-warning">\n12                              <div class="text-center">\n13                                  <h4>Your cart is empty</h4>\n14                                  <p>You have no items in your cart. <br><br><a href="{% url 'cities' %}</a>\n15                              </div>\n16                          </div>\n17                      </div>\n18                  </div>\n19              {% else %}\n20                  <div class="row">\n21                      <aside class="col-lg-9">\n22                          <div class="card">\n23                              <table class="table table-shopping-cart">\n24                                  <thead class="text-left">\n25                                      <tr class="small text-uppercase">\n26                                          <th scope="col">Product</th>\n27                                          <th scope="col" width="120">Quantity</th>\n28                                          <th scope="col" width="120">Price</th>\n29                                          <th scope="col" class="text-right" width="200"></th>\n30                                      </tr>\n31                                  </thead>\n32                                  <tbody>\n33                                      <tr>\n34                                          <td>\n35                                              <img alt="Product thumbnail" style="width: 100px; height: auto; border-radius: 10px;\" data-bbox="111 35 875 528">\n36                                          </td>\n37                                          <td>\n38                                              <input type="text" value="1" style="width: 100px; height: 30px; border: 1px solid #ccc; border-radius: 10px;\" data-bbox="111 35 875 528">\n39                                          </td>\n40                                          <td>\n41                                              <strong>100</strong>\n42                                          </td>\n43                                          <td style="text-align: right; vertical-align: bottom;\" data-bbox="111 35 875 528">\n44                                              <button type="button" class="btn btn-primary\" data-bbox="111 35 875 528">\n45                                                  View Details\n46                                              </button>\n47                                          </td>\n48                                      </tr>\n49                                  </tbody>\n50                              </table>\n51                          </div>\n52                      </div>\n53                  </div>\n54              {% endif %}\n55          </div>\n56      </section>\n57  {% endblock %}
```

```
Help           counsellor_individual.html - fyp - Visual Studio Code

:> counsellor_individual.html ×

templates > <> counsellor_individual.html

1   {% extends "base.html" %} 
2   {% load static %} 
3   {%-block start%} 
4 
5   <header class="banner flex items-center">
6     <div class="banner-image">
7 
8       {% comment %} 
14        <div class="container">
15          <!-- <h1 class="section-heading">{Name of Pharm
16        <div class="about-G-wrap flex">
17          <div class="flex-1">
18            
23            <h2>{{counsellor.counsellor_name}}</h2>
24            <p>
25            <span>License Number: <strong>{{counsel
26            <span>Working Hour : <strong>{{counsell
27            <span>Working Days : <strong>Sun, Mon,
28            <span>Cell Phone number : <strong>+977-
```



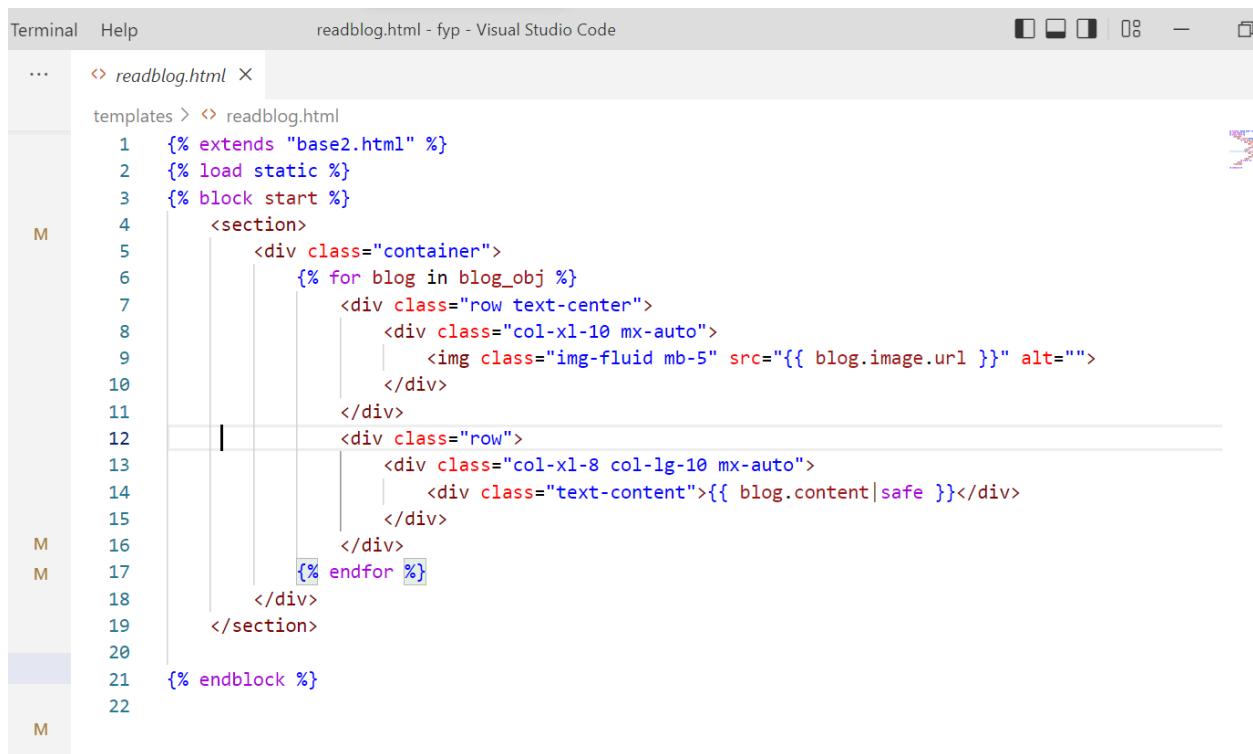
```
< dashboard.html > ...
templates > < dashboard.html
1  {% extends "base2.html" %}
2  {% load static %}
3  {% block start %}
4
5  <style>
6      .list-group-item.active {
7          z-index: 2;
8          color: #fff;
9          background-color: #008037;
10         border-color:#008037;
11     }
12 </style>
13 <!-- ===== SECTION CONTENT ===== -->
14 <section class="section-content padding-y bg">
15
16 <div class="container">
17     <div class="row">
18         <aside class="col-md-3">
19             <!-- SIDEBAR -->
20             <ul class="list-group">
21                 <a class="list-group-item active" href="#"> My order history </a>
22                 <a class="list-group-item" href="#"> Transactions </a>
23                 <a class="list-group-item" href="#"> Return and refunds </a>
24                 <a class="list-group-item" href="#">Settings </a>
25                 <a class="list-group-item" href="#"> My Selling Items </a>
26                 <a class="list-group-item" href="#"> Received orders </a>
27             </ul>
28             <br>
29             <a class="btn btn-light btn-block" href="#"> <i class="fa fa-power-off"></i> <span class='
30             <!-- SIDEBAR ./END -->
31         </aside>
```

Ln 13, Col 10 Tab Size: 4 UTF-8 CRLF Django HTML kite: ready

The screenshot shows a Visual Studio Code window with the title bar "Help pharmacy\_cities.html - fyp - Visual Studio Code". The main area displays the content of the file "pharmacy\_cities.html". The code is a Jinja2 template with the following structure:

```
1  {% extends "base2.html" %}          templates > <pharmacy_cities.html M X
2  {% load static %}                  <
3  {% block start %}                <
4  <section class="section all-products" id="products">    <
5  |   <div class="top container">      <
6  |   |   <h1>Cities</h1>        <
7  |   |   </div>                  <
8  |   </section>                <
9  <div class="container">          <
10 |   <div class="row">            <
11 |   |   <div class="card-deck">    <
12 |   |   |   {% for city in cities_all %}<
13 |   |   |   |   <div class="col-3">    <
14 |   |   |   |   |   <div class="card" style="width: 18rem; height:24rem;"> <
15 |   |   |   |   |   |    <
16 |   |   |   |   |   |   <div class="card-body"> <
17 |   |   |   |   |   |   |   <a href="{{ city.get_url }}><h5 class="card-title">{{ city.name }}</h5> </a> <
18 |   |   |   |   |   |   |   </div> <
19 |   |   |   |   |   |   <br> <
20 |   |   |   |   |   |   </div> <
21 |   |   |   |   |   |   <br> <
22 |   |   |   |   |   |   </div> <
23 |   |   |   |   |   |   <br> <
24 |   |   |   |   |   |   &#123; endfor &#125; <
25 |   |   |   |   |   </div> <
26 |   |   |   |   </div> <
27 |   |   </div> <
28 |   |   <div class="container"> <
29 |   |   |   <br> <
30 |   |   |   <nav aria-label="..."><
31 |   |   |   |   {% if cities_all.has_other_pages %}<
```

The status bar at the bottom of the code editor shows: Line 1 Col 1 Spaces: 4 LIPE-8 CRLE Dianoo HTML Write ready.

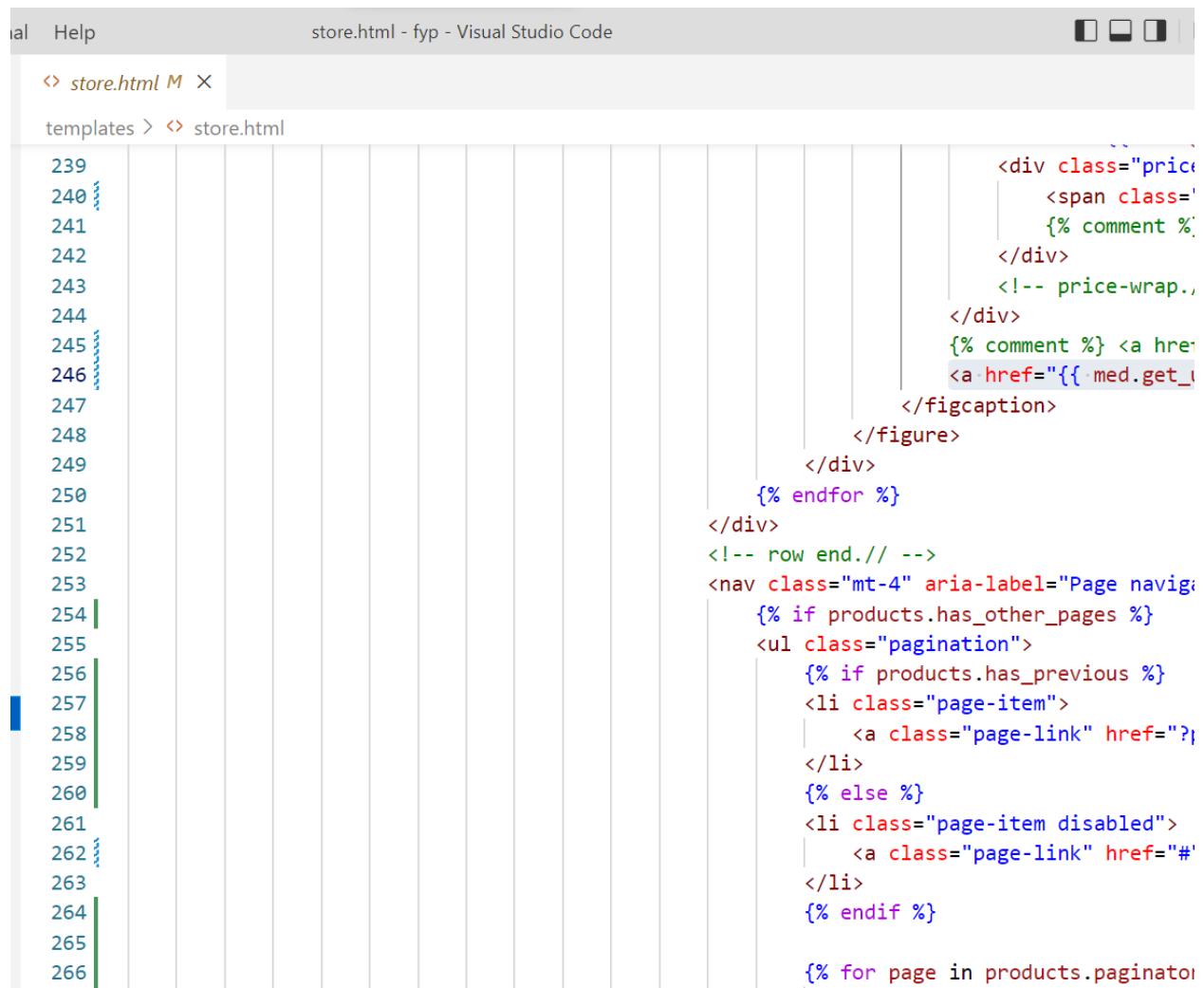


```
Terminal Help readblog.html - fyp - Visual Studio Code
...
... <div> readblog.html <x>
templates > <div> readblog.html
1   {% extends "base2.html" %}
2   {% load static %}
3   {% block start %}
4       <section>
5           <div class="container">
6               {% for blog in blog_obj %}
7                   <div class="row text-center">
8                       <div class="col-10 mx-auto">
9                           
10                      </div>
11                  </div>
12                  <div class="row">
13                      <div class="col-8 col-lg-10 mx-auto">
14                          <div class="text-content">{{ blog.content|safe }}</div>
15                      </div>
16                  </div>
17                  {% endfor %}
18              </div>
19          </section>
20      {% endblock %}
21
22
```

The screenshot shows a Visual Studio Code interface with the following details:

- File Explorer:** Shows a tree structure with "templates > rishi.html".
- Editor:** Displays the content of the file "rishi.html".
- Top Bar:** Shows "Help" and the file name "rishi.html - fyp - Visual Studio Code".

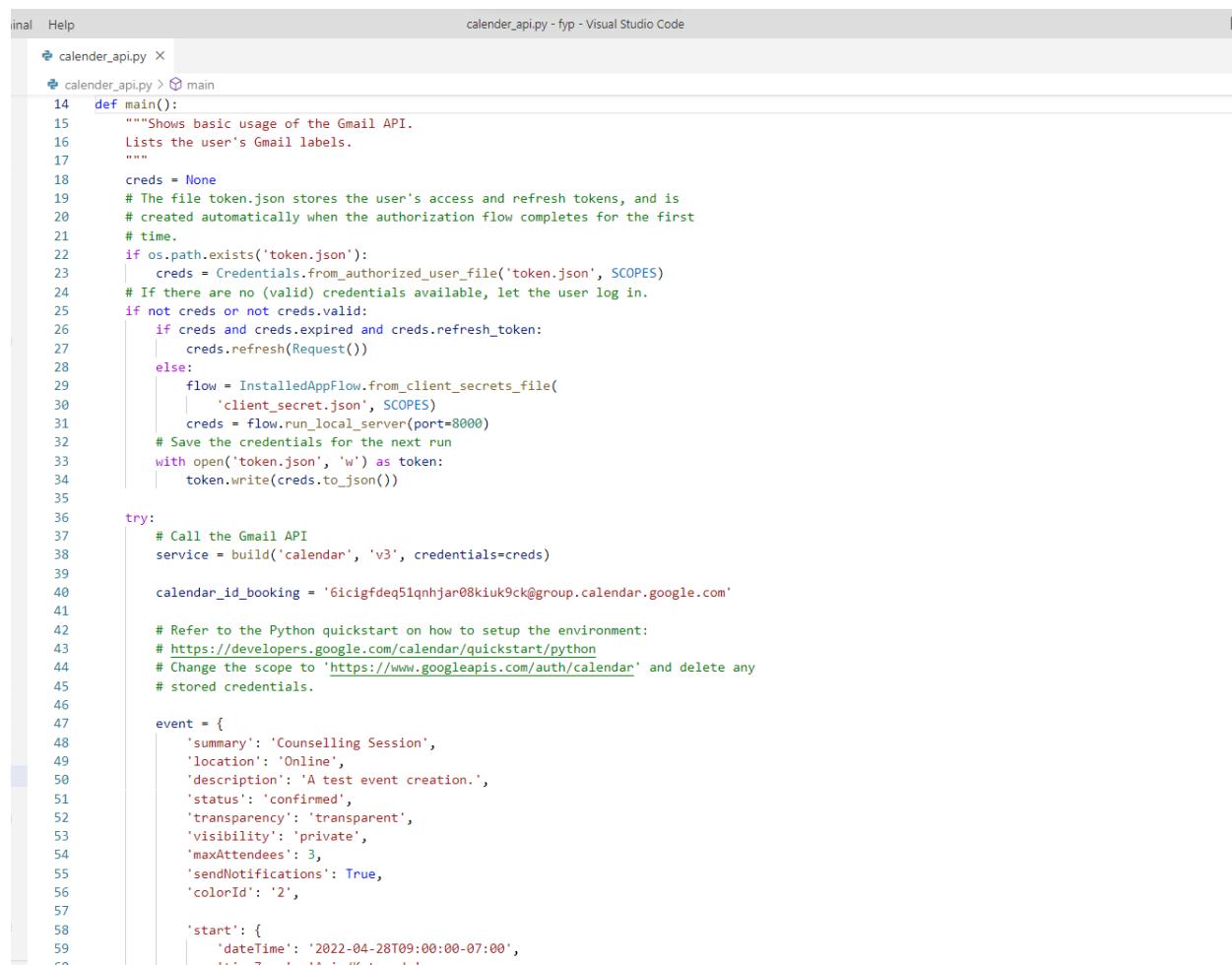
```
1  {% extends "base2.html" %}  
2  {% load static %}  
3  {% block start %}  
4  <style>  
5  
6  
7      .ftco-section {  
8          padding: 6em 0;  
9          position: relative;  
10     }  
11  
12     .container {  
13         width: 100%;  
14         padding-right: 15px;  
15         padding-left: 15px;  
16         margin-right: auto;  
17         margin-left: auto;  
18     }  
19     .container {  
20         max-width: 1250px;  
21     }  
22     @media (min-width: 1200px)  
23     {  
24         .container{  
25             max-width: 1140px;  
26         }  
27     }  
28 }
```



```
<div class="price">
    <span class='
        {% comment %}
    </div>
    <!-- price-wrap. -->
</div>
    {% comment %} <a href="#">
        <a href="{{ med.get_absolute_url }}>
            <img alt="Thumbnail image" />
            <div class="caption">
                <h3>{{ med.title }}</h3>
                <p>{{ med.description }}</p>
                <div class="rating">
                    <span style="color: #{{ med.rating }};">{{ med.rating }}
                </div>
                <div class="price">
                    <span class='
                        {% comment %}
                    </div>
                    <!-- price-wrap. -->
                </div>
                <div class="button">
                    <a href="#">
                        <span>Buy Now</span>
                    </a>
                </div>
            </div>
        </a>
    {% endcomment %}
</div>
</div>
<% endfor %>
</div>
<!-- row end.-->
<nav class="mt-4" aria-label="Page navigation">
    {% if products.has_other_pages %}
        <ul class="pagination">
            {% if products.has_previous %}
                <li class="page-item">
                    <a class="page-link" href="#"><span>Previous</span></a>
                </li>
            {% else %}
                <li class="page-item disabled">
                    <a class="page-link" href="#">
                        <span>Previous</span>
                    </a>
                </li>
            {% endif %}
            <li class="page-item" style="margin-left: 20px;">
                <a class="page-link" href="#">
                    <span>Next</span>
                </a>
            </li>
        </ul>
    {% endif %}
</nav>
```

```
<? load static ?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="shortcut icon" href="{% static 'images/favicon.ico' %}" type="image/x-icon">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Guardian</title>
    <link rel="stylesheet" type="text/css" href="https://cdn.jsdelivr.net/npm/slick-carousel@1.8.1/slick/slick.css" />
    <link rel="stylesheet" href="{% static 'css/base.css' %}" />
    <!-- <link rel="stylesheet" href="../css/test.css" > -->
    <!-- jQuery -->
    <script src="{% static 'js/jquery-2.0.0.min.js' %}" type="text/javascript"></script>
    <!-- Bootstrap4 files-->
    <script src="{% static 'js/bootstrap.bundle.min.js' %}" type="text/javascript"></script>
    <link href="{% static 'css/bootstrap.css' %}" rel="stylesheet" type="text/css"/>
    <!-- custom style -->
    <link href="{% static 'css/ui.css' %}" rel="stylesheet" type="text/css"/>
    <!-- <link href="css/responsive.css" rel="stylesheet" media="only screen and (max-width: 1200px)" -->
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/boxicons@latest/css/boxicons.min.css" />
    <link rel="stylesheet" type="text/css" href="https://cdn.jsdelivr.net/npm/slick-carousel@1.8.1/slick/slick.css" />
    <!-- custom javascript -->
    <script src="{% static 'js/script.js' %}" type="text/javascript"></script>
```

### 7.3.2. Code for the Automation Script

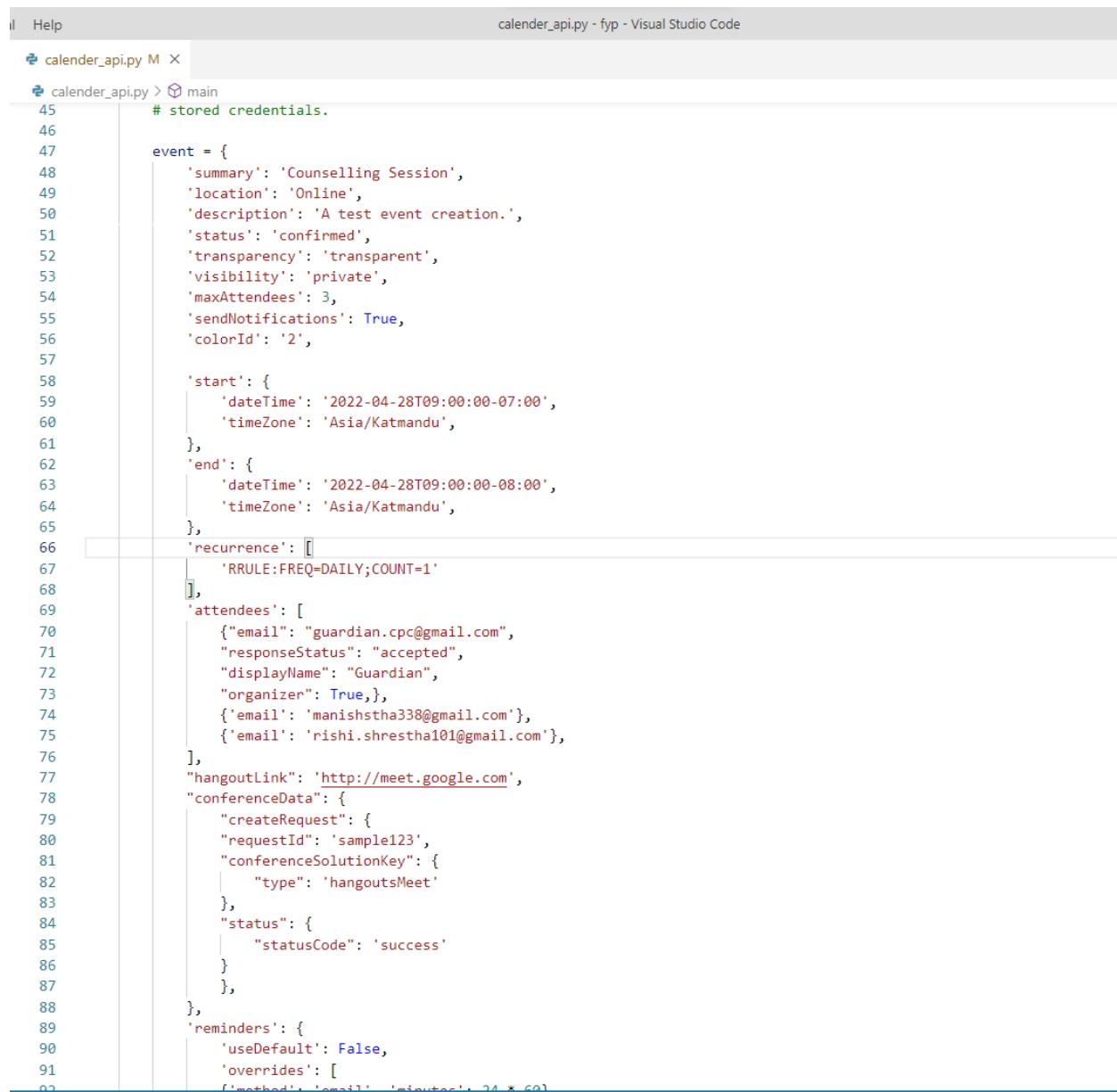


A screenshot of Visual Studio Code showing the code for a Python script named `calender_api.py`. The code uses the Gmail API to create a calendar event. It handles credential loading from a JSON file and manages token refreshes. The code then creates an event with a summary, location, and specific start and end times.

```
final Help
calender_api.py - fyp - Visual Studio Code
calender_api.py > main

14 def main():
15     """Shows basic usage of the Gmail API.
16     Lists the user's Gmail labels.
17     """
18
19     creds = None
20     # The file token.json stores the user's access and refresh tokens, and is
21     # created automatically when the authorization flow completes for the first
22     # time.
23     if os.path.exists('token.json'):
24         creds = Credentials.from_authorized_user_file('token.json', SCOPES)
25     # If there are no (valid) credentials available, let the user log in.
26     if not creds or not creds.valid:
27         if creds and creds.expired and creds.refresh_token:
28             creds.refresh(Request())
29         else:
30             flow = InstalledAppFlow.from_client_secrets_file(
31                 'client_secret.json', SCOPES)
32             creds = flow.run_local_server(port=8000)
33             # Save the credentials for the next run
34             with open('token.json', 'w') as token:
35                 token.write(creds.to_json())
36
37     try:
38         # Call the Gmail API
39         service = build('calendar', 'v3', credentials=creds)
40
41         calendar_id_booking = '6icigfdeq51qnjhj08kiuk9ck@google.com'
42
43         # Refer to the Python quickstart on how to setup the environment:
44         # https://developers.google.com/calendar/quickstart/python
45         # Change the scope to 'https://www.googleapis.com/auth/calendar' and delete any
46         # stored credentials.
47
48         event = {
49             'summary': 'Counselling Session',
50             'location': 'Online',
51             'description': 'A test event creation.',
52             'status': 'confirmed',
53             'transparency': 'transparent',
54             'visibility': 'private',
55             'maxAttendees': 3,
56             'sendNotifications': True,
57             'colorId': '2',
58
59             'start': {
60                 'dateTime': '2022-04-28T09:00:00-07:00',
61             }
62         }
63
64         event = service.events().insert(calendarId=calendar_id_booking, body=event).execute()
65         print(f'Event created: {event.get("htmlLink")}')
66     except HttpError as error:
67         print(f'An error occurred: {error}')


```



The screenshot shows a Visual Studio Code window with the file 'calender\_api.py' open. The code is a Python script for creating a Google Calendar event. It defines an event object with various properties like summary, location, and start/end times. It also includes a recurrence rule and attendee information. The code uses the Google Calendar API's event creation parameters.

```
45     # stored credentials.
46
47     event = {
48         'summary': 'Counselling Session',
49         'location': 'Online',
50         'description': 'A test event creation.',
51         'status': 'confirmed',
52         'transparency': 'transparent',
53         'visibility': 'private',
54         'maxAttendees': 3,
55         'sendNotifications': True,
56         'colorId': '2',
57
58         'start': {
59             'dateTime': '2022-04-28T09:00:00-07:00',
60             'timeZone': 'Asia/Katmandu',
61         },
62         'end': {
63             'dateTime': '2022-04-28T09:00:00-08:00',
64             'timeZone': 'Asia/Katmandu',
65         },
66         'recurrence': [
67             'RRULE:FREQ=DAILY;COUNT=1'
68         ],
69         'attendees': [
70             {"email": "guardian.cpc@gmail.com",
71              "responseStatus": "accepted",
72              "displayName": "Guardian",
73              "organizer": True,},
74             {'email': 'manishstha338@gmail.com'},
75             {'email': 'rishi.shrestha101@gmail.com'},
76         ],
77         "hangoutLink": 'http://meet.google.com',
78         "conferenceData": {
79             "createRequest": {
80                 "requestId": 'sample123',
81                 "conferenceSolutionKey": {
82                     "type": 'hangoutsMeet'
83                 },
84                 "status": {
85                     "statusCode": 'success'
86                 }
87             },
88             "reminders": {
89                 'useDefault': False,
90                 'overrides': [
91                     {'method': 'email', 'minutes': 24 * 60}
92                 ]
93             }
94         }
95     }
```

```

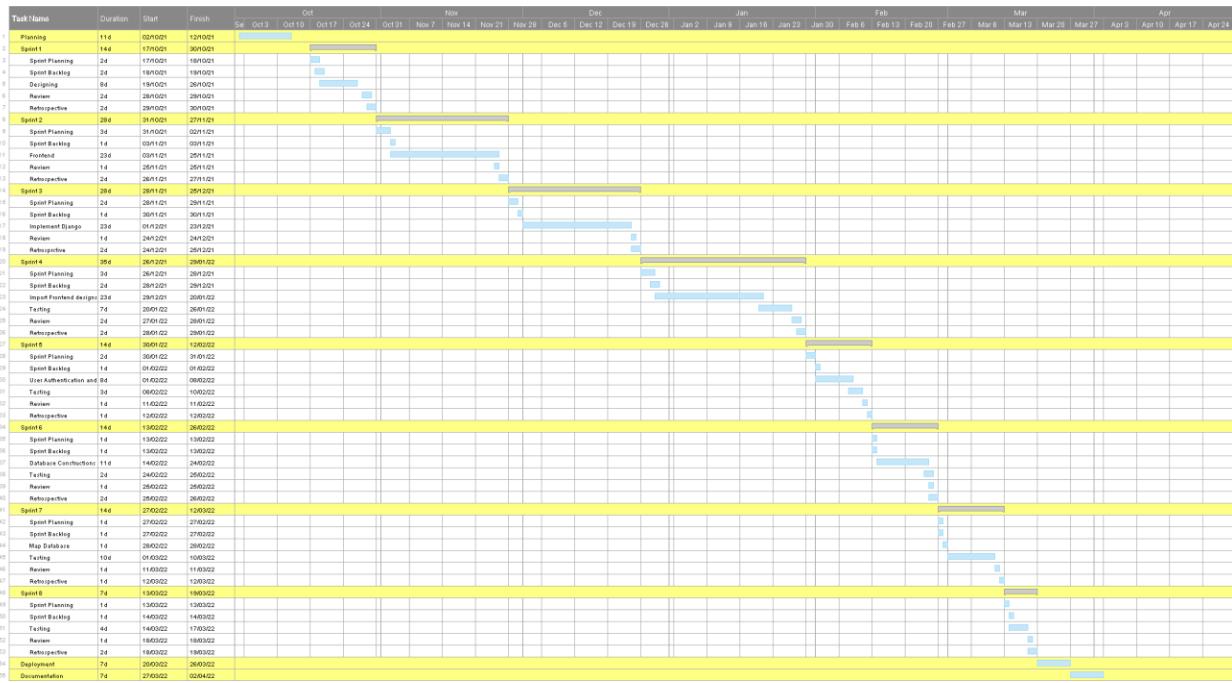
    user.save()

    # USER activation
    current_site = get_current_site(request)
    mail_subject = "Please activate your Guardian account"
    message = render_to_string('account/account_verification_email.html', {
        'user': user,
        'domain': current_site,
        'uid': urlsafe_base64_encode(force_bytes(user.pk)),
        'token': default_token_generator.make_token(user),
    })
    to_email = email
    userst = None
    if user_type == 'customer':
        userst = Type_user(user=user, is_customer=True)
        userst.save()
    elif user_type == 'counsellor':
        userst = Type_user(user=user, is_counsellor=True)
        userst.save()
    elif user_type == 'pharmacist':
        userst = Type_user(user=user, is_pharmacist=True)
        userst.save()
    send_email = EmailMessage(mail_subject, message, to=[to_email])
    send_email.send()
    # messages.success(request, 'Account created successfully. Please verify your email and login to continue.')
    return redirect('/account/login/?command=verification&email=' + email)

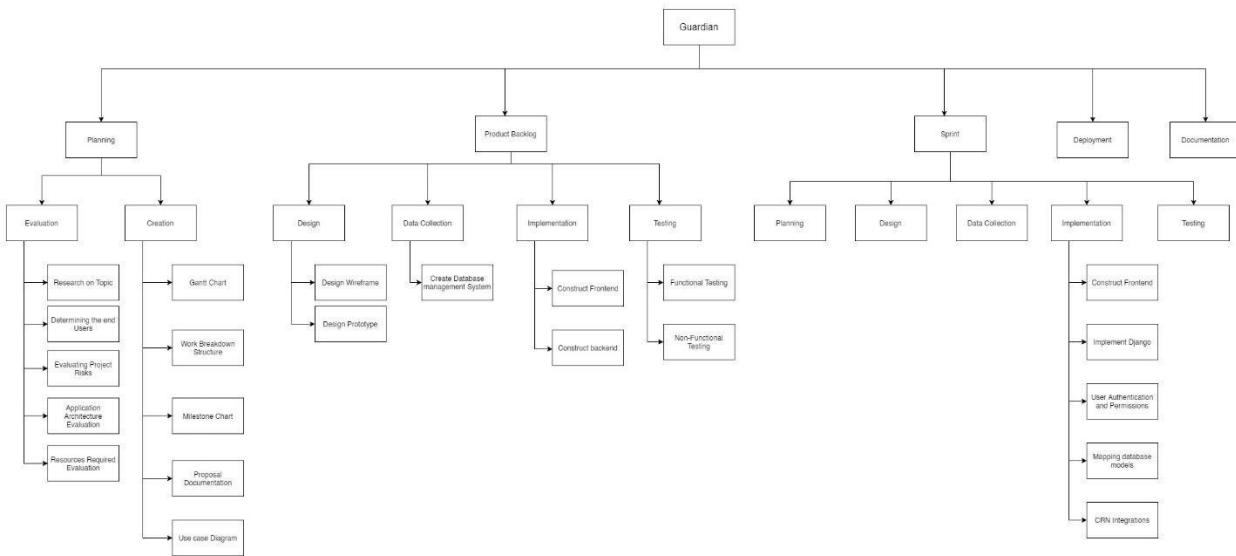
```

## 7.4. Design

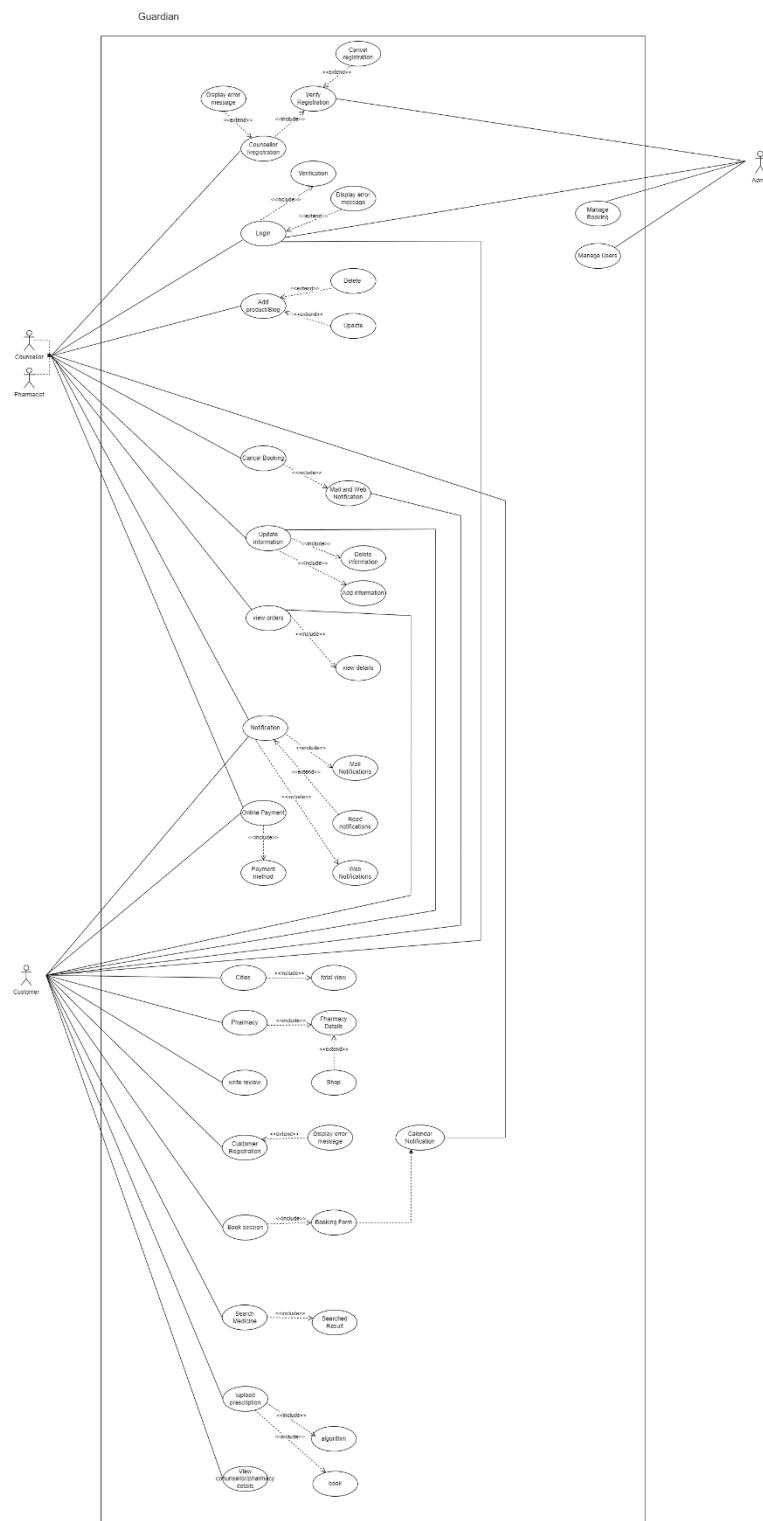
### 7.4.1. Gantt Chart



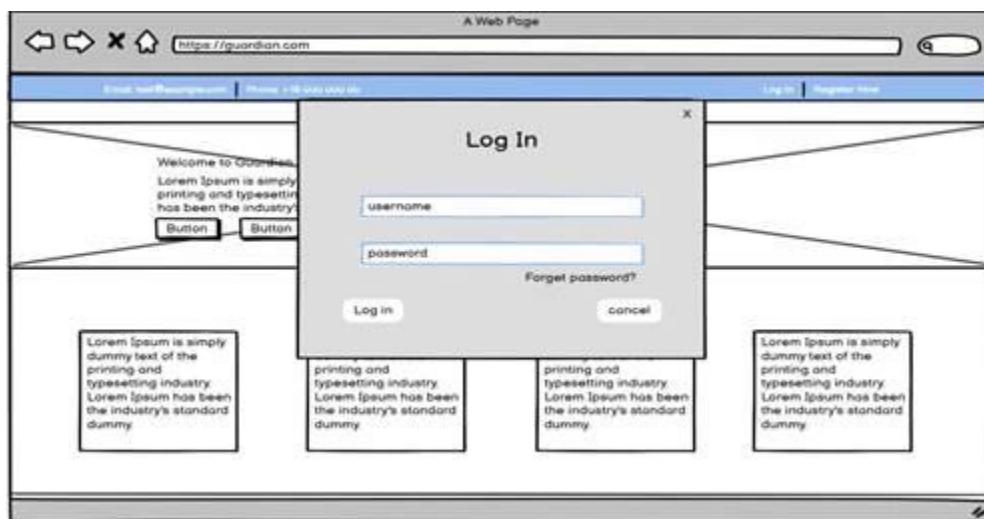
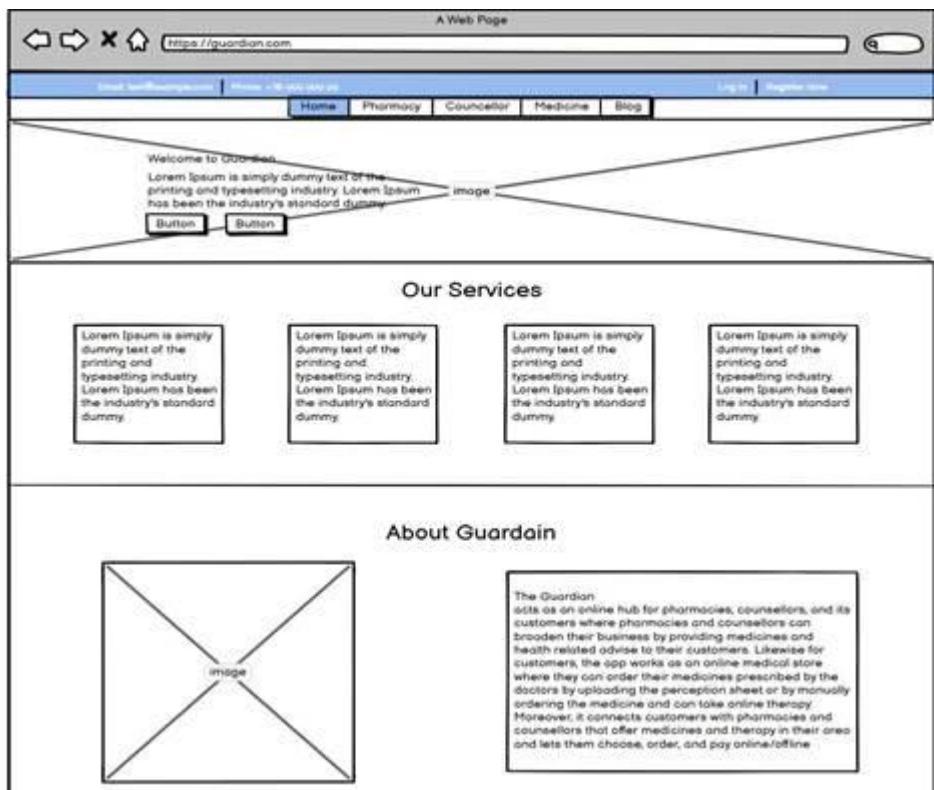
### 7.4.2. Work Breakdown Structure

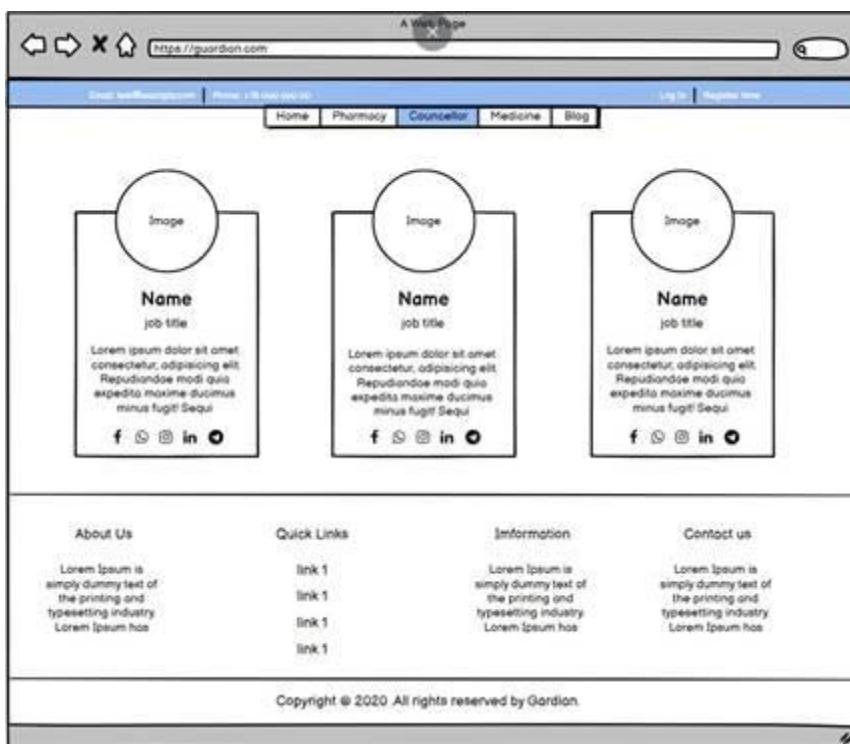
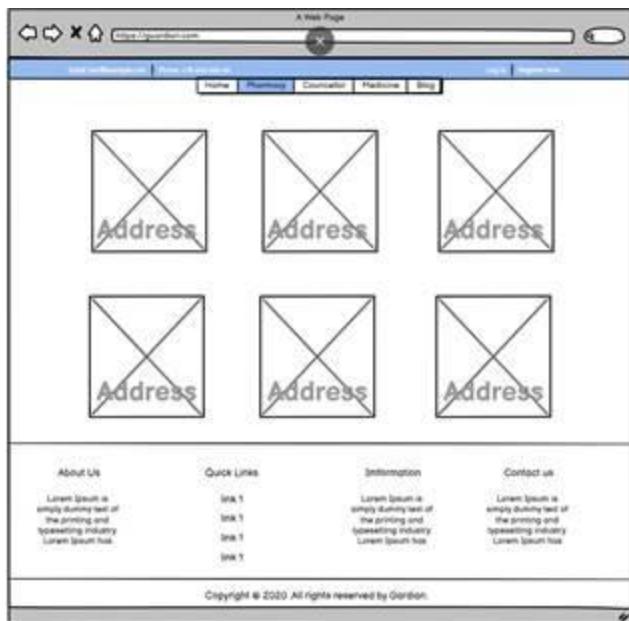


### 7.4.3. Use Case



#### 7.4.4. Wireframe





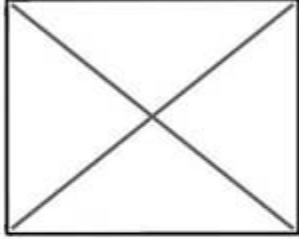
A Web Page

<https://guardian.com>

Email: test@example.com | Phone: +98-000 000 00 | Log In | Register Now

Home | Pharmacy | Councillor | Medicine | Blog

Name



Name  
position

+977 - 98-1234-234

test@gardian.com

[Book Appointment](#)

[f](#) [g](#) [i](#) [in](#) [o](#)

**Specialty**

**Degree**

**Training**

**Working**

**About Us**

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.

**Quick Links**

link 1

link 1

link 1

link 1

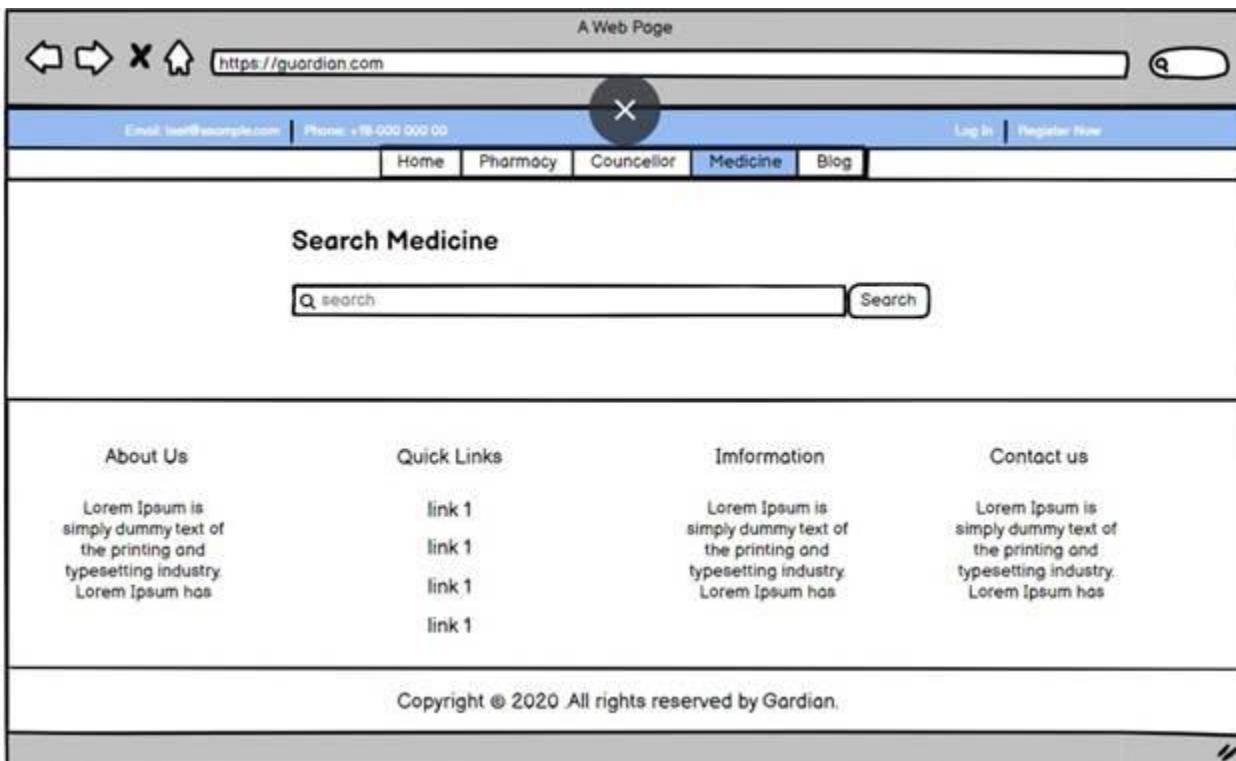
**Information**

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.

**Contact us**

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.

Copyright © 2020 All rights reserved by Gardian.



A Web Page

<https://guardian.com>

Email: test@example.com | Phone: +18-000 000 00

Log In | Register Now

Home Pharmacy Councillor Medicine Blog

## Medicine

### Name of Medicine

Common name

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy. Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy.

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy. Lorem Ipsum is simply dummy text of the printing and typesetting industry.

**About Us**

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy.

**Quick Links**

- link 1
- link 1
- link 1
- link 1

**Information**

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy.

**Contact us**

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy.

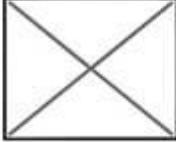
Copyright © 2020 All rights reserved by Gordian.

A Web Page  
<https://guardian.com>

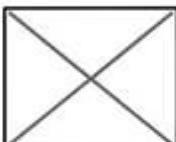
Email: test@example.com | Phone: +91 900 000 0000 | Log In | Register Now

[Home](#) [Pharmacy](#) [Councillor](#) [Medicine](#) [Blog](#)

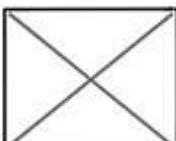
**Title**  
 Author Date  
 Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy. Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy.



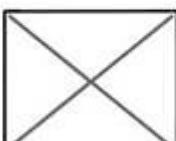
**Title**  
 Author Date  
 Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy. Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy.



**Title**  
 Author Date  
 Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy. Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy.



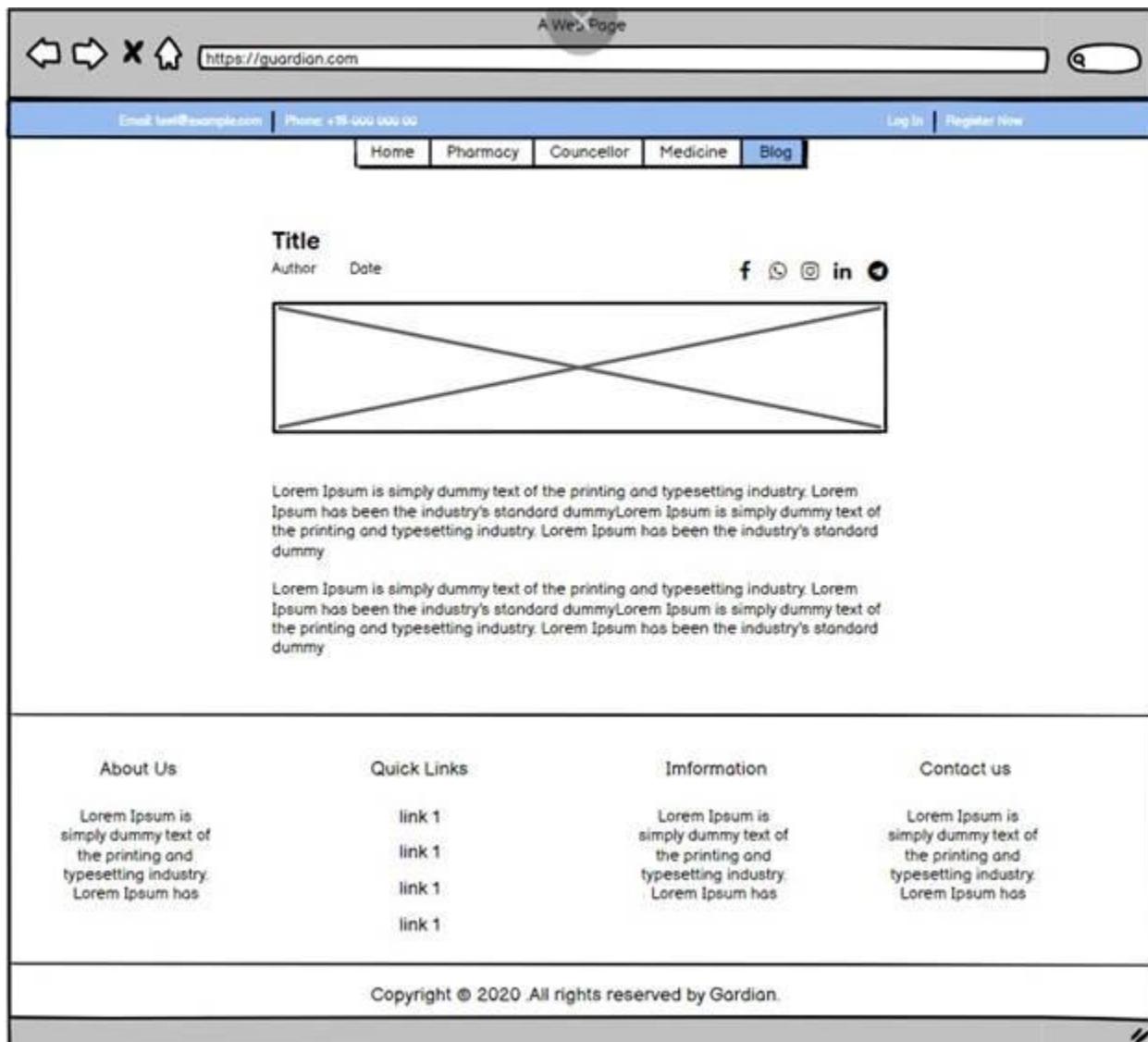
**Title**  
 Author Date  
 Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy. Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy.




---

About Us	Quick Links	Information	Contact us
	link 1 link 1 link 1 link 1		

Copyright © 2020 .All rights reserved by Gordian.



## 7.5. Screenshots of the System

Email: test@example.com | Phone: +18-000 000 00

Log In | Register Now

Home Pharmacy Counsellor Medicine Blog

Welcome to Guardian

**Share us your Problems**

Lorem Ipsum is simply dummy text of the printing and typesetting industry.  
Lorem Ipsum has been the industry's standard dummy.

Read More Shop Now

**Our services**

Place your medicine Get Readable Contact Near by Reserve therapy

Email: test@example.com | Phone: +18-000 000 00

Log In | Register

Home Pharmacy Counsellor Medicine Blog

**Cities**

Dhara	Itahari	Biratnagar	Dulari
Belbari	Birgunj	Taraharaha	Budi Khola

Email: test@example.com | Phone: +18-000 000 00

[Log In](#) | [Register Now](#)

Home    Pharmacy    Councillor    Medicine    Blog



**Manish Shrestha**  
Counsellor

this is a description



**Teen Counselling Center**  
Counsellor

The Teen Counselling center provides services like teen counselling, group counselling, and organizes various kinds of programmes on mental health. Here the Counsellors conduct lectures, workshops, and programmes throughout the academic year.

Email: test@example.com | Phone: +18-000 000 00

[Rishi Shrestha](#) | [Logout](#)

Home    Pharmacy    Councillor    Medicine    Blog

## Teen Counselling Center



**License Number:** 3224/2002/122  
**Working Hour :** 9:00AM to 4:00PM  
**Working Days :** Sun, Mon, Tue, Wen, Thu, Fri  
**Cell Phone number :** +977-9801010101  
**Email :** teen@counselling.com

**Description**  
The Teen Counselling center provides services like teen counselling, group counselling, and organizes various kinds of programmes on mental health. Here the Counsellors conduct lectures, workshops, and programmes throughout the academic year.

[Book an Appointment](#)

**Specialised On**

**Book Meeting**

Meeting Title  Age

Marital Status

Date  Time

Please check counsellor office time before scheduling.

Description

Describe the purpose of booking a meeting session.

Email: test@example.com | Phone: +18-000 000 00

Log in | Register Now

Home Pharmacy Councillor Medicine Blog

## Search

Lore Ipsum as their default model text, and a search for 'lore ipsum' will uncover many web sites still in their infancy.

Medicine name

---

<b>About us</b>	<b>Quik Links</b>	<b>Information</b>	<b>CONTACT US</b>
It was popularised in the 1960 with the release of Latest sheets containing Lore Ipsum passage.  <a href="#">Read More</a>	Home About us Products Blog Services Gallery Contact us	Privacy Policy Terms and Conditions Products Brands Conditions of Use Internet based Ads	Aamar Path, Dharan 56700 Dharan- 13, Sunsari company@gmail.com +977-9876543210 Kathmandu, Nepal

Copyright © 2020. All rights reserved by [Guardian](#).

Email: test@example.com | Phone: +18-000 000 00

Log in | Register Now

Home Pharmacy Councillor Medicine Blog

## All Blogs



Manish Shrestha · Apr 26, 2022, 5 am · [Read](#)

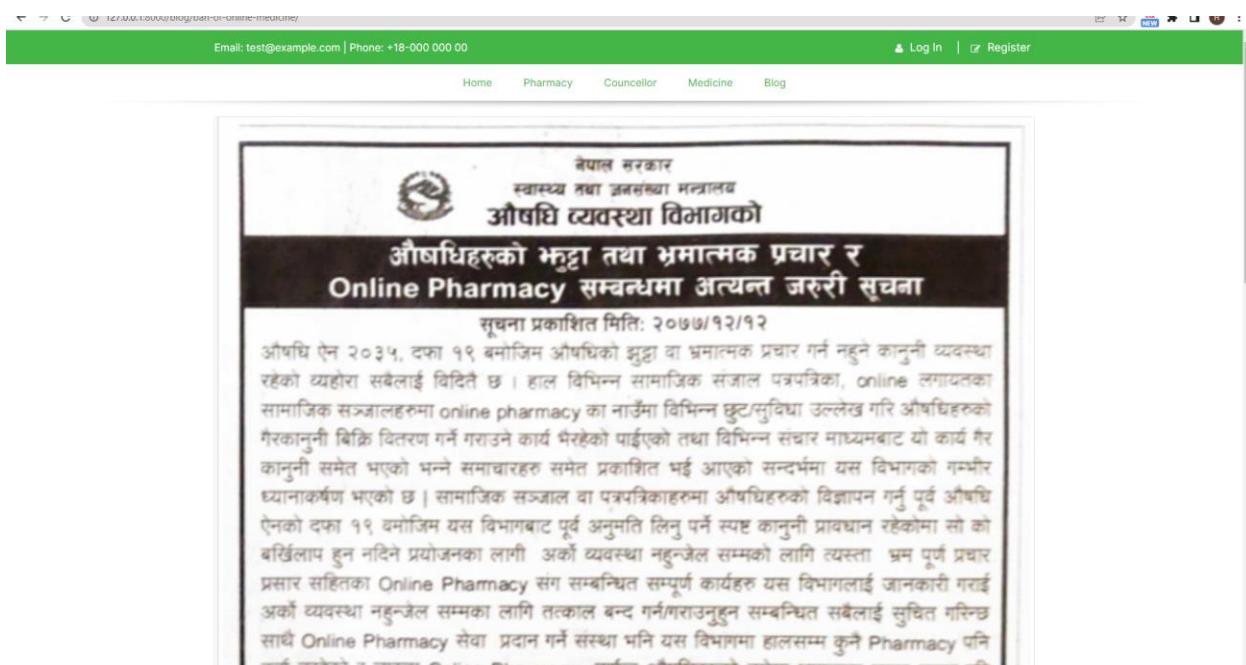


Manish Shrestha · Apr 26, 2022, 6:04 a.m. · [Read](#)

[1](#) [2](#) [3](#) [4](#) [→](#)

---

<b>About us</b>	<b>Quik Links</b>	<b>Information</b>	<b>CONTACT US</b>
It was popularised in the 1960 with the release of Latest sheets containing Lore Ipsum passage.	Home About us Products Blog Services Gallery Contact us	Privacy Policy Terms and Conditions Products Brands Conditions of Use Internet based Ads	Aamar Path, Dharan 56700 Dharan- 13, Sunsari company@gmail.com +977-9876543210 Kathmandu, Nepal



Add Products

Add Product

Name	Alternative Name	
Cost	Stock	
Image		
<input type="button" value="Choose File"/> No file chosen		
Category*	Usual Dose for child	Usual Dose for Adult
Contraindication	Indication	
Special Precautions		Adverse Effect

Email: test@example.com | Phone: +18-000 000 00      Rishi Shrestha | Logout

Home    Pharmacy    Councillor    Medicine    Blog

## Pharmacies

**Test Pharmacy nepal**  
this is a description of test pharmacy nepal

[Visit](#)

**Hamro Swastiya**  
Hamro swastiya pharmacy is the clinical health science that links medical science with chemistry and it is charged with the discovery, production, disposal, safe and effective use, and control of medications and drugs.

[Visit](#)

Terms and conditions.

Copyright © 2020. All rights reserved by Gardian.

Email: test@example.com | Phone: +18-000 000 00      Rishi Shrestha | Logout

Home    Pharmacy    Councillor    Medicine    Blog

## Test Pharmacy nepal

OGRM: **099312312**  
INN: **099312312**  
Working Hour: **9 to 6**  
Working Days: **Sun, Mon, Tue, Wen, Thu, Fri**  
Cell Phone number: **9810503530**  
Email: **Submittyc@gmail.com**

**Description**  
this is a description of test pharmacy nepal

[Store](#)

## Our services

Email: test@example.com | Phone: +98-000 000 00       Rishi Shrestha | 

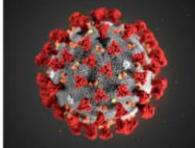
Home    Pharmacy    Councillor    Medicine    Blog

**Store**    All category  

Categories

- Liquid
- Inhalers
- Injections
- Suppositories
- Tablets
- Implants

2 Items found



Paracetamol  
Rs: 123/-

[View Details](#)



Cough Syrup  
Rs: 150/-

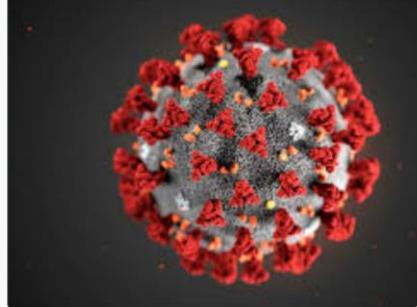
[View Details](#)

[Terms and conditions](#)

Copyright © 2020. All rights reserved by Gardian.

Email: test@example.com | Phone: +98-000 000 00       Rishi Shrestha | 

Home    Pharmacy    Councillor    Medicine    Blog



### Paracetamol

Rs: 123/-

**Dose**  
45ml

**Drugs Type**  
 Low     High

**Side Effects**

- side effects 1
- side effects 1
- side effects 1

[Add to cart](#)

[Terms and conditions](#)

Copyright © 2020. All rights reserved by Gardian.

Email: test@example.com | Phone: +98-000 000 00      Rishi Shrestha | Logout

Home   Pharmacy   Councillor   Medicine   Blog

PRODUCT	QUANTITY	PRICE
Paracetamol paracetamol	1	Rs: 123 Rs: 123 each

**Remove**

Total price: Rs: 123 /-  
Shipping Cost: Rs: 2.46 /-

Total: **Rs: 125.46** /-

**Cash on Delivery**

**Pay with PayPal**

**Debit or Credit Card**

Powered by **PayPal**

[Terms and conditions](#)

Copyright © 2020 .All rights reserved by Gardian.

Guardian   127.0.0.1:8000/cart/   Incognito

Email: test@example.com | Phone: +98-000 000 00      Rishi Shrestha | Logout

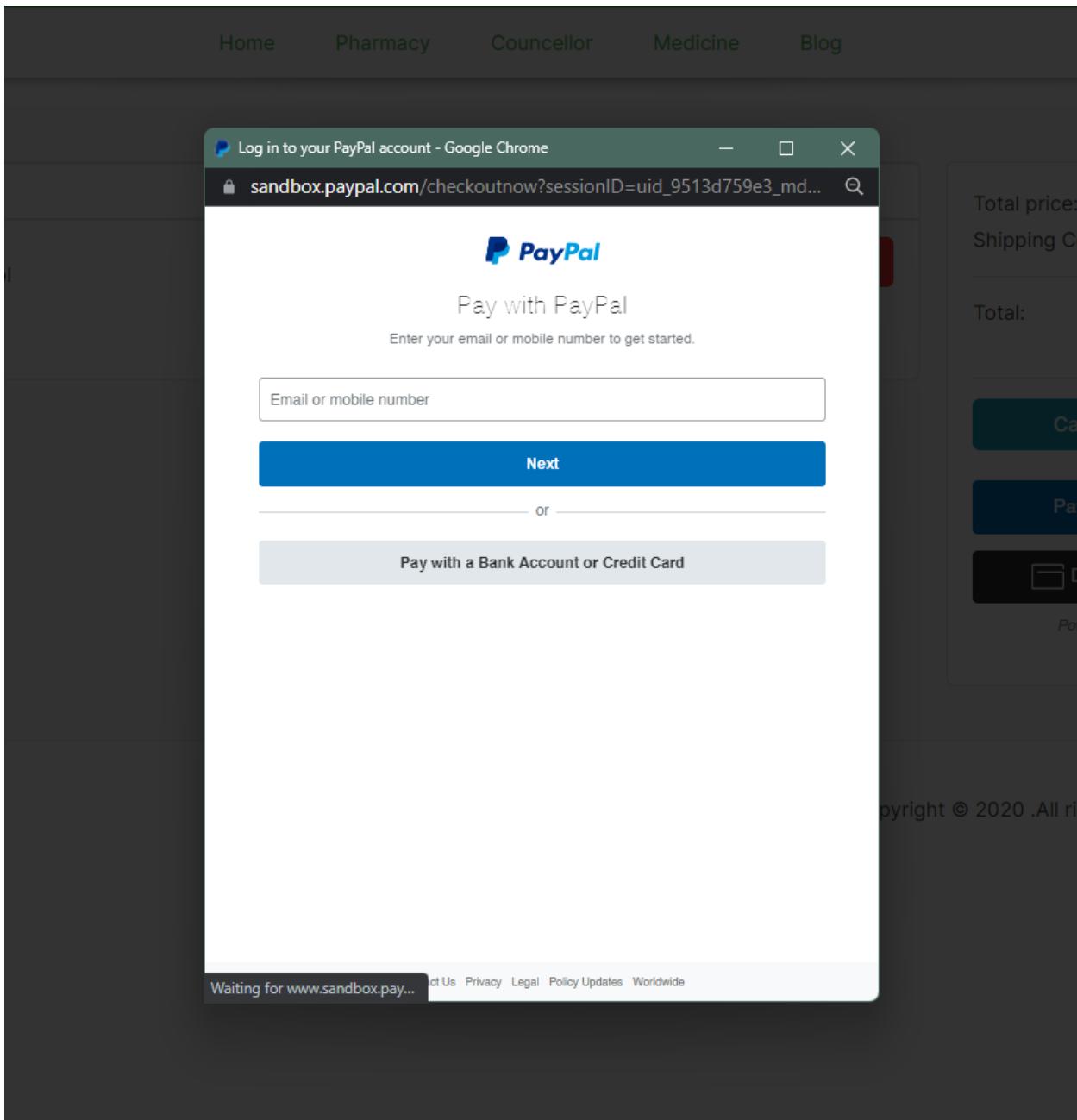
Home   Pharmacy   Councillor   Medicine   Blog

**Your cart is empty**  
You have no items in your cart.

**Continue shopping**

[Terms and conditions](#)

Copyright © 2020 .All rights reserved by Gardian.



Email: test@example.com | Phone: +18-000 000 00      Anish KC | Logout

Home    Pharmacy    Councillor    Medicine    Blog

[Add Blog](#)
[View added Blogs](#)

### Add Blog

Title:

Content:

Type something

Powered by Froala

Characters : 0

Image:  No file chosen

[Add Blog](#)
[Clear](#)

[Terms and conditions](#)

Copyright © 2020 .All rights reserved by Gardian.

127.0.0.1:8000/councillor/added\_blog/      Anish KC | Logout

Home    Pharmacy    Councillor    Medicine    Blog

TITLE	DATE	UPDATE	REMOVE	Total Blogs:
 hj Alternative Name: name1	April 27, 2022, 8:09 a.m.	<a href="#">Update</a>	<a href="#">Remove</a>	4

[Go to Dashboard](#)
[Add Blog](#)

[Terms and conditions](#)

Copyright © 2020 .All rights reserved by Gardian.

**Sign in**

Email Address

Password

[Forgot password?](#)

**Login**

Don't have account? [Sign up](#)

**Sign up**

First name Last name

First name Last name

Email Who are you?

example@domain.com Customer

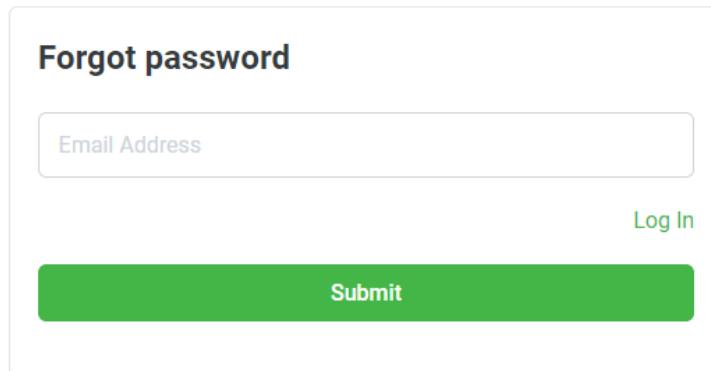
Create password Conform password

Password Confirm Password

I agree the [term and condition](#).

**Register**

Have an account? [Log In](#)



The image shows a user interface for a 'Forgot password' feature. It consists of a light gray rectangular box with rounded corners. At the top left, the text 'Forgot password' is displayed in bold black font. Below this is a horizontal input field with a thin gray border, containing the placeholder 'Email Address'. To the right of the input field, the word 'Log In' is written in a small green font. At the bottom of the box is a large green rectangular button with the word 'Submit' centered in white.

## 7.6. Future Work

The functions and features of the application are completed as per the proposal. However, this application can be improved further, and new features can be added to the application. Some of the features that can be added to enhance this application are as follows:

- Implementing reply feature for comment and review
- Order tracking
- Prescription reading
- Implementing Chat system
- Enhance Multi-store platform
- Host on trustworthy hosting platform.