



Islington college
(इस्लिङ्टन कलेज)

Module Code & Module Title

CS6P05NI Final Year Project

Assessment Weightage & Type

40% Final Project Report

Year and Semester

2021-22 Autumn

Student Name: Abhishek Rokaya

London Met ID: 19030695

College ID: NP01CP4A190164

Internal Supervisor: Mohit Sharma

External Supervisor: Ankit Tandukar

Assignment Due Date: 27th April 2022

Assignment Submission Date: 27th April 2022

Word Count: 8410

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 marks of zero will be awarded.

Abstract

The following report is based on the web application called 'Dental Appointment System'. The development was done using Python Language in Django Framework. A dental treatment and consultation is required for everyone. Due to the rise on technology, every hospital and clinic is shifting towards having online method of management and appointment which is much easier to record. But in the context of Nepal, there is still a lack of proper system for it as people still have to wait hours in line to get a consultation.

This project was created for the client 'Phulchowki Dental Clinic' which also supports viewing report of the appointment of the users. This report has a detailed explanation on how the system was created, what methodologies was selected along with the design and the testing. A detailed research was done to develop this report.

Acknowledgement

Firstly, I would like to express my special thanks of gratitude to Mr. Mohit Sharma and Mr. Ankit Tandukar, who were the supervisors of this project. Without their guidance, this project would be next to impossible. I am very grateful for their guidance, suggestions and valuable feedbacks regarding the project along with their dedication and supervision to complete this project.

I would also like to thank Islington College for providing me with this opportunity of doing this project. I would also like to thank my client, Dr. Prashamsa Khadka, who is the founder of the Phulchowki Dental Clinic for her valuable feedback and requirement which helped me aid to create this project. I would also like to thank my friends who helped me solve many problems that had arisen during the development of this project and also for their valuable response during the survey.

Without all of it, this project would not have been completed. I would like to thank everyone mentioned above again because of whom I was able to complete this report and this project. Thank you everyone.

Table of Contents

1. Introduction	1
1.1 Project Description.....	1
1.2 Problem Scenario	1
1.3 Project as a Solution.....	2
1.4 Aims and Objectives	4
1.5 Structure of the report.....	4
1.5.1 Background.....	4
1.5.2 Development.....	4
1.5.3 Testing and Analysis	5
1.5.4 Conclusion	5
2. Background	6
2.1 About the end users.....	6
2.1.1 Discussion with client	6
2.1.2 End Users	7
2.2 Understanding the solution	8
2.3 Similar Projects.....	10
2.4 Comparisons.....	13
2.5 Critical evaluation of the comparisons	14
3. Development	15
3.1 Considered Methodologies	15
3.2 Selected Methodology	17
3.3 Phases of Methodology	18
3.4 Survey	20

3.4.1 Pre-Survey Results	20
3.4.2 Post-Survey Results.....	21
3.5 Requirement Analysis	22
3.5.1 Feature Requirements	22
3.5.2 Hardware and Software Requirements	22
3.6 Design	23
3.6.1 Overall Use Case Diagram.....	24
3.6.2 Functional UML Diagrams.....	25
3.7 Implementation	57
3.7.1 System Architecture	57
3.7.2 Frontend Implementation	59
3.7.3 Database Implementation	65
3.7.4 Backend Implementation.....	67
4. Testing and analysis.....	69
4.1 Test Plan	69
4.1.1 Black Box Testing, Test Plan	69
4.1.2 White Box Testing, Test Plan	70
4.1.3 Integration Testing, Test Plan	71
4.1.4 System Testing, Test Plan	71
4.1.5 Client Evaluation Testing, Test Plan	72
4.2 Black Box Testing	73
4.3 White Box Testing.....	89
4.4 Integration Testing	100
4.5 System Testing	134
4.6 Client Evaluation Testing	135

4.7 Critical Analysis	140
5. Conclusion	141
5.1 Legal, Social and Ethical Issues	142
5.1.1 Legal Issues	142
5.1.2 Social Issues	142
5.1.3 Ethical Issues	142
5.2 Advantages.....	143
5.3 Limitations	144
5.4 Future Work.....	144
6. Bibliography and References	145
7. Appendix	148
7.1 Appendix A: Background	148
7.1.2 Discussion with client	148
7.2 Appendix B: Development	150
7.2.1 Considered Methodologies.....	150
7.2.2 Selected Methodology.....	153
7.3 Appendix C: Pre-Survey	155
7.3.1 Pre-Survey Form.....	155
7.3.2 Sample of filled pre-survey forms.....	159
7.3.3 Pre-Survey Result	168
7.4 Appendix D: Post-Survey.....	173
7.4.1 Post-Survey Form	173
7.4.2 Sample of Filled Post-Survey Forms.....	176
7.4.3 Post-Survey Result	182
7.5 Appendix E: Sample Codes	186

7.5.1 Sample code of the UI.....	186
7.5.2 Sample code for the Automation Script.....	227
7.6 Appendix F: Designs.....	242
7.6.1 Functional UML Diagrams.....	244
7.6.2 Gantt Chart.....	313
7.6.3 Work Breakdown Structure	314
7.6.4 Algorithms and Flowcharts	315
7.6.5 ER Diagram Creation Process	319
7.6.6 Creation of Use Case Diagram	326
7.6.7 Class Diagram.....	329
7.6.8 Software Requirement Specification (SRS).....	330
7.6.9 Wireframes.....	352
7.7 Appendix G: Testing	375
7.7.1 Black Box Testing	375
7.7.2 White Box Testing	391
7.8 Appendix H: Screenshots of the System	414
7.9 Appendix I: User Feedback	448
7.9.1 User Feedback Form	448
7.9.2 Sample of Filed User Feedback Forms.....	451
7.9.3 User Feedback Form Result	454
7.10 Appendix J - User Manual	460
7.11 Appendix K: Future Work.....	479
7.11.1 Readings for future work	479

List of Figures

Figure 1: Survey of problem	2
Figure 2: Survey of solution 1.....	3
Figure 3: Survey of solution 2.....	3
Figure 4: Client Role Permission.....	6
Figure 5: Client's Requirements	7
Figure 6: Dent-inn.....	10
Figure 7: Dentalife.....	11
Figure 8: Kumari Dental Care.....	12
Figure 9: Overall Use Case Diagram.....	24
Figure 10: Use Case Diagram – User Registration	25
Figure 11: Collaboration Diagram – User Registration.....	26
Figure 12: Activity Diagram – User Registration.....	27
Figure 13: Sequence Diagram – User Registration	28
Figure 14: Use Case Diagram – User Login.....	29
Figure 15: Collaboration Diagram – User Login	29
Figure 16: Activity Diagram – User Login	30
Figure 17: Sequence Diagram – User Login	31
Figure 18: Use Case Diagram – User View Profile.....	32
Figure 19: Collaboration Diagram – User View Profile	32
Figure 20: Activity Diagram – User View Profile	33
Figure 21: Sequence Diagram – User View Profile	34
Figure 22: Use Case Diagram – User Edit Profile	35
Figure 23: Collaboration Diagram – User Edit Profile.....	35
Figure 24: Activity Diagram – User Edit Profile	36
Figure 25: Sequence Diagram – User Edit Profile.....	37
Figure 26: Use Case Diagram – User Book Appointment	38
Figure 27: Collaboration Diagram – User Book Appointment.....	39
Figure 28: Activity Diagram – User Book Appointment	40

Figure 29: Sequence Diagram – User Book Appointment.....	41
Figure 30: Use Case Diagram – User View Appointment.....	42
Figure 31: Collaboration Diagram – User View Appointment	43
Figure 32: Activity Diagram – User View Appointment.....	43
Figure 33: Sequence Diagram – User View Appointment.....	44
Figure 34: Use Case Diagram – User Cancel Appointment	45
Figure 35: Collaboration Diagram – User Cancel Appointment.....	45
Figure 36: Activity Diagram – User Cancel Appointment.....	46
Figure 37: Sequence Diagram – User Cancel Appointment.....	47
Figure 38: Use Case Diagram – User View Doctor	48
Figure 39: Collaboration Diagram – User View Doctor.....	48
Figure 40: Activity Diagram – User View Doctor	49
Figure 41: Sequence Diagram – User View Doctor.....	50
Figure 42: Use Case Diagram – User View Report.....	51
Figure 43: Collaboration Diagram – User View Report.....	51
Figure 44: Activity Diagram – User View Report	52
Figure 45: Sequence Diagram – User View Report	53
Figure 46: Use Case Diagram – User View Services.....	54
Figure 47: Collaboration Diagram – User View Services.....	54
Figure 48: Activity Diagram – User View Services	55
Figure 49: Sequence Diagram – User View Services.....	56
Figure 50: System Architecture (AskPython, 2022).....	58
Figure 51: Frontend Implementation Code – 1	59
Figure 52: Frontend Implementation Code – 2	60
Figure 53: Frontend Implementation Code – 3	60
Figure 54: Frontend Implementation Code – 4	61
Figure 55: Frontend Implementation Code – 5	62
Figure 56: Frontend Implementation Code – 6	62
Figure 57: Frontend Implementation Code – 7	63
Figure 58: Frontend View	64
Figure 59: Database Implementation Code – 1	65

Figure 60: Database Implementation Code - 2.....	65
Figure 61: Database Implementation Code – 3.....	66
Figure 62: Database Implementation Code – 4.....	66
Figure 63: Backend Implementation Code - 1	67
Figure 64: Backend Implementation Code - 2	68
Figure 65: Black Box Test 1 (Unsuccessful).....	73
Figure 66: Black Box Test 1 - Error Code.....	74
Figure 67: Black Box Test 1 - Fixed Code.....	74
Figure 68: Black Box Test 1 (Successful).....	75
Figure 69: Black Box Test 2 (Unsuccessful) - 1	76
Figure 70: Black Box Test 2 (Unsuccessful) - 2	77
Figure 71: Black Box Test 2 (Unsuccessful) – 3.....	77
Figure 72: Black Box Test 2 - Error Code.....	78
Figure 73: Black Box Test 2 - Fixed Code.....	78
Figure 74: Black Box Test 2 (Successful) - 1	79
Figure 75: Black Box Test 2 (Successful) - 2	80
Figure 76: Black Box Test 3 (Unsuccessful).....	81
Figure 77: Black Box Test 3 - Error Code.....	82
Figure 78: Black Box Test 3 - Fixed Code.....	82
Figure 79: Black Box Test 3 (Successful).....	83
Figure 80: Black Box Test 4 (Unsuccessful) - 1	84
Figure 81: Black Box Test 4 (Unsuccessful) – 2.....	85
Figure 82: Black Box Test 4 - Error Code.....	85
Figure 83: Black Box Test 4 - Fixed Code.....	86
Figure 84: Black Box Test 4 (Successful) - 1	87
Figure 85: Black Box Test 4 (Successful) - 2	88
Figure 86: White Box Test 1 - 1.....	89
Figure 87: White Box Test 1 - 2.....	90
Figure 88: White Box Test 1 - 3.....	90
Figure 89: White Box Test 2 - 1.....	91
Figure 90: White Box Test 2 - 2.....	92

Figure 91: White Box Test 2 - 3.....	92
Figure 92: White Box Test 3 - 1.....	93
Figure 93: White Box Test 3 - 2.....	94
Figure 94: White Box Test 4 - 1.....	95
Figure 95: White Box Test 4 - 2.....	96
Figure 96: White Box Test 4 - 3.....	96
Figure 97: White Box Test 5 - 1.....	97
Figure 98: White Box Test 5 – 2.....	98
Figure 99: White Box Test 5 – 3.....	98
Figure 100: Integration Test 1 - 1	100
Figure 101: Integration Test 1 - 2	101
Figure 102: Integration Test 1 - 3.....	101
Figure 103: Integration Test 1 – 4	102
Figure 104: Integration Test 1 - 5	102
Figure 105: Integration Test 1 - Database Before	103
Figure 106: Integration Test 1 - Database After	103
Figure 107: Integration Test 2 - 1	104
Figure 108: Integration Test 2 – 2	105
Figure 109: Integration Test 2 - 3	105
Figure 110: Integration Test 2 - Database Before	106
Figure 111: Integration Test 2 - Database After	106
Figure 112: Integration Test 3 - 1	107
Figure 113: Integration Test 3 – 2	108
Figure 114: Integration Test 3 – 3	108
Figure 115: Integration Test 3 – 4	109
Figure 116: Integration Test 3 – 5	109
Figure 117: Integration Test 3 - Database Before	110
Figure 118: Integration Test 3 - Database After	110
Figure 119: Integration Test 4 – 1	111
Figure 120: Integration Test 4 – 2	112
Figure 121: Integration Test 4 - 3.....	112

Figure 122: Integration Test 4 - Database Before	113
Figure 123: Integration Test 4 - Database After	113
Figure 124: Integration Test 5 - 1	114
Figure 125: Integration Test 5 - 2	115
Figure 126: Integration Test 5 - 3	115
Figure 127: Integration Test 5 - Database Before	116
Figure 128: Integration Test 5 - Database After	116
Figure 129: Integration Test 6 - 1	117
Figure 130: Integration Test 6 – 2	118
Figure 131: Integration Test 6 - 3	118
Figure 132: Integration Test 6 - 4	119
Figure 133: Integration Test 6 - 5	119
Figure 134: Integration Test 6 - Database Before	120
Figure 135: Integration Test 6 - Database After	120
Figure 136: Integration Test 7 - 1	121
Figure 137: Integration Test 7 – 2	122
Figure 138: Integration Test 7 – 3	122
Figure 139: Integration Test 7 - Database Before	123
Figure 140: Integration Test 7 - Database After	123
Figure 141: Integration Test 8 – 1	124
Figure 142: Integration Test 8 - 2	125
Figure 143: Integration Test 8 - 3	125
Figure 144: Integration Test 8 - Database before	126
Figure 145: Integration Test 8 - Database After	126
Figure 146: Integration Test 9 - 1	127
Figure 147: Integration Test 9 – 2	128
Figure 148: Integration Test 9 - 3	128
Figure 149: Integration Test 9 - 4	129
Figure 150: Integration Test 9 – 5	129
Figure 151: Integration Test 9 - Database Before	130
Figure 152: Integration Test 9 - Database After	130

Figure 153: Integration Test 10 - 1	131
Figure 154: Integration Test 10 – 2	132
Figure 155: Integration Test 10 - 3	132
Figure 156: Integration Test 10 - Database Before	133
Figure 157: Integration Test 10 - Database After	133
Figure 158: System Testing 1	134
Figure 159: Client Evaluation Testing – 1.....	135
Figure 160: Client Evaluation Testing – 2.....	135
Figure 161: Client Evaluation Testing – 3.....	135
Figure 162: Client Evaluation Testing – 4.....	136
Figure 163: Client Evaluation Testing – 5.....	136
Figure 164: Client Evaluation Testing – 6.....	137
Figure 165: Client Evaluation Testing – 7.....	137
Figure 166: Client Evaluation Testing – 8.....	138
Figure 167: Client Evaluation Testing – 9.....	138
Figure 168: Client Evaluation Testing – 10.....	139
Figure 169: RUP Methodology	150
Figure 170: Prototype Methodology	151
Figure 171: Incremental Methodology	152
Figure 172: Pre-Survey Form - 1.....	155
Figure 173: Pre-Survey Form - 2.....	156
Figure 174: Pre-Survey Form - 3.....	156
Figure 175: Pre-Survey Form - 4.....	157
Figure 176: Pre-Survey Form - 5.....	157
Figure 177: Pre-Survey Form - 6.....	158
Figure 178: Sample 1 of pre-survey form - 1	159
Figure 179: Sample 1 of pre-survey form - 2.....	159
Figure 180: Sample 1 of pre-survey form - 3.....	160
Figure 181: Sample 1 of pre-survey form - 4.....	160
Figure 182: Sample 1 of pre-survey form - 5.....	161
Figure 183: Sample 2 of pre-survey form - 1	162

Figure 184: Sample 2 of pre-survey form - 2	162
Figure 185: Sample 2 of pre-survey form - 3	163
Figure 186: Sample 2 of pre-survey form - 4	163
Figure 187: Sample 2 of pre-survey form - 5	164
Figure 188: Sample 3 of pre-survey form - 1	165
Figure 189: Sample 3 of pre-survey form - 2	165
Figure 190: Sample 3 of pre-survey form - 3	166
Figure 191: Sample 3 of pre-survey form - 4	166
Figure 192: Sample 3 of pre-survey form - 5	167
Figure 193: Pre-Survey Result – 1	168
Figure 194: Pre-Survey Result - 2	168
Figure 195: Pre-Survey Result – 3	169
Figure 196: Pre-Survey Result - 4	169
Figure 197: Pre-Survey Result - 5	170
Figure 198: Pre-Survey Result - 6	170
Figure 199: Pre-Survey Result - 7	171
Figure 200: Pre-Survey Result - 8	172
Figure 201: Post-Survey Form - 1	173
Figure 202: Post-Survey Form - 2	174
Figure 203: Post-Survey Form - 3	174
Figure 204: Post-Survey Form - 4	175
Figure 205: Post-Survey Form - 5	175
Figure 206: Sample 1 of post-survey form - 1	176
Figure 207: Sample 1 of post-survey form - 2	176
Figure 208: Sample 1 of post-survey form - 3	177
Figure 209: Sample 1 of post-survey form - 4	177
Figure 210: Sample 2 of post-survey form - 1	178
Figure 211: Sample 2 of post-survey form - 2	178
Figure 212: Sample 2 of post-survey form - 3	179
Figure 213: Sample 2 of post-survey form - 4	179
Figure 214: Sample 3 of post-survey form - 1	180

Figure 215: Sample 3 of post-survey form - 2	180
Figure 216: Sample 3 of post-survey form - 3	181
Figure 217: Sample 3 of post-survey form - 4	181
Figure 218: Post-Survey Result – 1	182
Figure 219: Post-Survey Result - 2	182
Figure 220: Post-Survey Result - 3	183
Figure 221: Post-Survey Result - 4	183
Figure 222: Post-Survey Result - 5	184
Figure 223: Post-Survey Result - 6	184
Figure 224: Post-Survey Result - 7	185
Figure 225: Sample Code of UI - 1.....	186
Figure 226: Sample Code of UI - 2.....	187
Figure 227: Sample Code of UI - 3.....	187
Figure 228: Sample Code of UI - 4.....	188
Figure 229: Sample Code of UI - 5.....	188
Figure 230: Sample Code of UI - 6.....	189
Figure 231: Sample Code of UI - 7.....	189
Figure 232: Sample Code of UI - 8.....	190
Figure 233: Sample Code of UI – 9.....	191
Figure 234: Sample Code of UI - 10.....	192
Figure 235: Sample Code of UI - 11.....	192
Figure 236: Sample Code of UI - 12.....	193
Figure 237: Sample Code of UI - 13.....	194
Figure 238: Sample Code of UI - 14.....	194
Figure 239: Sample Code of UI - 15.....	195
Figure 240: Sample Code of UI - 16.....	196
Figure 241: Sample Code of UI - 17.....	196
Figure 242: Sample Code of UI - 18.....	197
Figure 243: Sample Code of UI - 19.....	198
Figure 244: Sample Code of UI - 20.....	199
Figure 245: Sample Code of UI - 21.....	200

Figure 246: Sample Code of UI - 22.....	201
Figure 247: Sample Code of UI - 23.....	202
Figure 248: Sample Code of UI - 24.....	203
Figure 249: Sample Code of UI – 25.....	204
Figure 250: Sample Code of UI - 26.....	205
Figure 251: Sample Code of UI - 27	206
Figure 252: Sample Code of UI - 28.....	207
Figure 253: Sample Code of UI - 29.....	208
Figure 254: Sample Code of UI - 30.....	209
Figure 255: Sample Code of UI - 31	210
Figure 256: Sample Code of UI - 32.....	211
Figure 257: Sample Code of UI - 33.....	211
Figure 258: Sample Code of UI – 34.....	212
Figure 259: Sample Code of UI - 35.....	213
Figure 260: Sample Code of UI - 36.....	214
Figure 261: Sample Code of UI - 37.....	215
Figure 262: Sample Code of UI - 38.....	216
Figure 263: Sample Code of UI - 39.....	217
Figure 264: Sample Code of UI – 40.....	218
Figure 265: Sample Code of UI - 41	219
Figure 266: Sample Code of UI - 42.....	220
Figure 267: Sample Code of UI - 43.....	221
Figure 268: Sample Code of UI - 44.....	222
Figure 269: Sample Code of UI – 45.....	223
Figure 270: Sample Code of UI - 46.....	224
Figure 271: Sample Code of UI - 47	225
Figure 272: Sample Code of UI - 48.....	226
Figure 273: Sample Code of Automation Script - 1	227
Figure 274: Sample Code of Automation Script - 2	228
Figure 275: Sample Code of Automation Script - 2	229
Figure 276: Sample Code of Automation Script - 3	230

Figure 277: Sample Code of Automation Script - 4	231
Figure 278: Sample Code of Automation Script - 5	232
Figure 279: Sample Code of Automation Script - 6	232
Figure 280: Sample Code of Automation Script - 7	233
Figure 281: Sample Code of Automation Script - 8	234
Figure 282: Sample Code of Automation Script - 9	234
Figure 283: Sample Code of Automation Script - 10	235
Figure 284: Sample Code of Automation Script - 11	235
Figure 285: Sample Code of Automation Script - 12	236
Figure 286: Sample Code of Automation Script - 13	236
Figure 287: Sample Code of Automation Script - 14	237
Figure 288: Sample Code of Automation Script - 15	237
Figure 289: Sample Code of Automation Script - 16	238
Figure 290: Sample Code of Automation Script - 17	238
Figure 291: Sample Code of Automation Script - 18	239
Figure 292: Sample Code of Automation Script - 19	240
Figure 293: Sample Code of Automation Script - 20	240
Figure 294: Sample Code of Automation Script - 21	241
Figure 295: Use Case Diagram – Admin Login	244
Figure 296: Collaboration Diagram – Admin Login.....	245
Figure 297: Activity Diagram – Admin Login	246
Figure 298: Sequence Diagram – Admin Login.....	247
Figure 299: Use Case Diagram – Admin View Users.....	248
Figure 300: Collaboration Diagram – Admin View Users	248
Figure 301: Activity Diagram – Admin View Users	249
Figure 302: Sequence Diagram – Admin View Users	250
Figure 303: Use Case Diagram – Admin Delete Users	251
Figure 304: Collaboration Diagram – Admin Delete Users.....	251
Figure 305: Activity Diagram – Admin Delete Users	252
Figure 306: Sequence Diagram – Admin Delete Users.....	253
Figure 307: Use Case Diagram – Admin View Doctors.....	254

Figure 308: Collaboration Diagram – Admin View Doctors	254
Figure 309: Activity Diagram – Admin View Doctors	255
Figure 310: Sequence Diagram – Admin View Doctors	256
Figure 311: Use Case Diagram – Admin Add Doctors	257
Figure 312: Collaboration Diagram – Admin Add Doctors.....	257
Figure 313: Activity Diagram – Admin Add Doctors	258
Figure 314: Sequence Diagram – Admin Add Doctors.....	259
Figure 315: Use Case Diagram – Admin Delete Doctors	260
Figure 316: Collaboration Diagram – Admin Delete Doctors.....	261
Figure 317: Activity Diagram – Admin Delete Doctors	262
Figure 318: Sequence Diagram – Admin Delete Doctors.....	263
Figure 319: Use Case Diagram – Admin Edit Doctors	264
Figure 320: Collaboration Diagram – Admin Edit Doctors	264
Figure 321: Activity Diagram – Admin Edit Doctors.....	265
Figure 322: Sequence Diagram – Admin Edit Doctors.....	266
Figure 323: Use Case Diagram – Admin Add Report.....	267
Figure 324: Collaboration Diagram – Admin Add Report	268
Figure 325: Activity Diagram – Admin Add Report	269
Figure 326: Sequence Diagram – Admin Add Report	270
Figure 327: Use Case Diagram – Admin Delete Report.....	271
Figure 328: Collaboration Diagram – Admin Delete Report	272
Figure 329: Activity Diagram – Admin Delete Report	273
Figure 330: Sequence Diagram – Admin Delete Report	274
Figure 331: Use Case Diagram – Admin Edit Report	275
Figure 332: Collaboration Diagram – Admin Edit Report.....	275
Figure 333: Activity Diagram – Admin Edit Report	276
Figure 334: Sequence Diagram – Admin Edit Report.....	277
Figure 335: Use Case Diagram – Admin View Reports.....	278
Figure 336: Collaboration Diagram – Admin View Reports	278
Figure 337: Activity Diagram – Admin View Reports	279
Figure 338: Sequence Diagram – Admin View Reports	280

Figure 339: Use Case Diagram – Admin Book Appointment.....	281
Figure 340: Collaboration Diagram – Admin Book Appointment	281
Figure 341: Activity Diagram – Admin Book Appointment.....	282
Figure 342: Sequence Diagram – Admin Book Appointment	283
Figure 343: Use Case Diagram – Admin View Appointments	284
Figure 344: Collaboration Diagram – Admin View Appointments.....	284
Figure 345: Activity Diagram – Admin View Appointments.....	285
Figure 346: Sequence Diagram – Admin View Appointments	286
Figure 347: Use Case Diagram – Admin Cancel Appointment.....	287
Figure 348: Collaboration Diagram – Admin Cancel Appointment	287
Figure 349: Activity Diagram – Admin Cancel Appointment.....	288
Figure 350: Sequence Diagram – Admin Cancel Appointment	289
Figure 351: Use Case Diagram – Admin Approve Appointment	290
Figure 352: Collaboration Diagram – Admin Approve Appointment	290
Figure 353: Activity Diagram – Admin Approve Appointment.....	291
Figure 354: Sequence Diagram – Admin Approve Appointment	292
Figure 355: Use Case Diagram – Admin Reject Appointment.....	293
Figure 356: Collaboration Diagram – Admin Reject Appointment	293
Figure 357: Activity Diagram – Admin Reject Appointment.....	294
Figure 358: Sequence Diagram – Admin Reject Appointment	295
Figure 359: Use Case Diagram – Admin Finish Appointment	296
Figure 360: Collaboration Diagram – Admin Finish Appointment.....	296
Figure 361: Activity Diagram – Admin Finish Appointment.....	297
Figure 362: Sequence Diagram – Admin Finish Appointment	298
Figure 363: Use Case Diagram – Admin View Services	299
Figure 364: Collaboration Diagram – Admin View Services	299
Figure 365: Activity Diagram – Admin View Services.....	300
Figure 366: Sequence Diagram – Admin View Services	301
Figure 367: Use Case Diagram – Admin Add Services.....	302
Figure 368: Collaboration Diagram – Admin Add Services	302
Figure 369: Activity Diagram – Admin Add Services	303

Figure 370: Sequence Diagram – Admin Add Services	304
Figure 371: Use Case Diagram – Admin Delete Services.....	305
Figure 372: Collaboration Diagram – Admin Delete Services	306
Figure 373: Activity Diagram – Admin Delete Services	307
Figure 374: Sequence Diagram – Admin Delete Services	308
Figure 375: Use Case Diagram – Admin Edit Service.....	309
Figure 376: Collaboration Diagram – Admin Edit Service	309
Figure 377: Activity Diagram – Admin Edit Service	310
Figure 378: Sequence Diagram – Admin Edit Service	311
Figure 379: Gantt chart	313
Figure 380: Work Breakdown Structure	314
Figure 381: Bubble Sort Algorithm	315
Figure 382: Flowchart of the overall system.....	318
Figure 383: User-patientUser Relation.....	319
Figure 384: patientUser-Appointment Relation	320
Figure 385: Report-Appointment Relation	320
Figure 386: Doctor-Appointment Relation	320
Figure 387: Service-Appointment Relation.....	321
Figure 388: Final ERD:.....	324
Figure 389: Rectangle which represents System	326
Figure 390: Human stick figure which represents Actor	326
Figure 391: Oval which represents Use Case	327
Figure 392: Solid line which represents the relation between actor and use case	327
Figure 393: Include Relation	327
Figure 394: Extend Relation.....	327
Figure 395: Generalisation Relation	328
Figure 396: Class Diagram.....	329
Figure 397: System Perspective Diagram	332
Figure 398: Overall Use Case of the System	333
Figure 399: UI of Home Page 1.....	344
Figure 400: UI of Home Page 2.....	345

Figure 401: UI of Home Page 3.....	345
Figure 402: UI of Home Page 4.....	346
Figure 403: UI of Home Page 5.....	346
Figure 404: UI of Home Page 6.....	347
Figure 405: UI of Home Page 7.....	347
Figure 406: UI of Home Page 8.....	348
Figure 407: UI of Home Page 9.....	348
Figure 408: Wireframe - Home Page.....	352
Figure 409: Wireframe - User Registration.....	353
Figure 410: Wireframe - User Login	354
Figure 411: Wireframe - Admin Login.....	355
Figure 412: Wireframe - User Dashboard	356
Figure 413: Wireframe - User View Appointment.....	357
Figure 414: Wireframe - User Book Appointment.....	358
Figure 415: Wireframe - User View Doctor.....	359
Figure 416: Wireframe - User View Report	360
Figure 417: Wireframe - User View Service	361
Figure 418: Wireframe - User View Profile.....	362
Figure 419: Wireframe - User Edit Profile.....	363
Figure 420: Wireframe - Admin Dashboard.....	364
Figure 421: Wireframe - Admin View Appointment	365
Figure 422: Wireframe - Admin Approve Appointment.....	366
Figure 423: Wireframe - Admin Book Appointment	367
Figure 424: Wireframe - Admin View Doctor	368
Figure 425: Wireframe - Admin Add Doctor	369
Figure 426: Wireframe - Admin View Report	370
Figure 427: Wireframe - Admin Add Report	371
Figure 428: Wireframe - Admin View Service.....	372
Figure 429: Wireframe - Admin Add Service	373
Figure 430: Wireframe - Admin View Users	374
Figure 431: Black Box Test 5 (Unsuccessful).....	375

Figure 432: Black Box Test 5 - Error Code.....	376
Figure 433: Black Box Test 5 - Fixed Code.....	376
Figure 434: Black Box Test 5 (Successful).....	377
Figure 435: Black Box Test 6 (Unsuccessful) - 1	378
Figure 436: Black Box Test 6 (Unsuccessful) - 2	379
Figure 437: Black Box Test 6 (Unsuccessful) – 3.....	379
Figure 438: Black Box Test 6 - Error Code.....	380
Figure 439: Black Box Test 6 - Fixed Code.....	380
Figure 440: Black Box Test 6 (Successful) - 1	381
Figure 441: Black Box Test 6 (Successful) – 2.....	382
Figure 442: Black Box Test 6 (Successful) - 3	382
Figure 443: Black Box Test 7 (Unsuccessful).....	383
Figure 444: Black Box Test 7 - Error Code.....	384
Figure 445: Black Box Test 7 - Fixed Code.....	384
Figure 446: Black Box Test 7 (Successful).....	385
Figure 447: Black Box Test 8 (Unsuccessful) - 1	386
Figure 448: Black Box Test 8 (Unsuccessful) – 2.....	387
Figure 449: Black Box Test 8 - Error Code.....	387
Figure 450: Black Box Test 8 - Fixed Code.....	387
Figure 451: Black Box Test 8 (Successful) - 1	388
Figure 452: Black Box Test 8 (Successful) - 2	389
Figure 453: Black Box Test 8 - Database before.....	389
Figure 454: Black Box Test 8 - Database after.....	389
Figure 455: White Box Test 6 - 1.....	391
Figure 456: White Box Test 6 – 2.....	392
Figure 457: White Box Test 6 - 3.....	392
Figure 458: White Box Test 7	393
Figure 459: White Box Test 8 - 1.....	394
Figure 460: White Box Test 8 – 2.....	395
Figure 461: White Box Test 8 - 3.....	395
Figure 462: White Box Test 9 - 1.....	396

Figure 463: White Box Test 9 - 2.....	397
Figure 464: White Box Test 10 - 1.....	398
Figure 465: White Box Test 10 - 2.....	399
Figure 466: White Box Test 11	400
Figure 467: White Box Test 12 - 1.....	401
Figure 468: White Box Test 12 – 2.....	402
Figure 469: White Box Test 12 - 3.....	402
Figure 470: White Box Test 13 - 1.....	403
Figure 471: White Box Test 13 – 2.....	404
Figure 472: White Box Test 13 - 3.....	404
Figure 473: White Box Test 14 - 1.....	405
Figure 474: White Box Test 14 – 2.....	406
Figure 475: White Box Test 14 – 3.....	406
Figure 476: White Box Test 15 - 1.....	407
Figure 477: White Box Test 15 – 2.....	408
Figure 478: White Box Test 15 – 3.....	408
Figure 479: White Box Test 16 - 1.....	409
Figure 480: White Box Test 16 - 2.....	410
Figure 481: White Box Test 16 - 3.....	410
Figure 482: White Box Test 17 - 1.....	411
Figure 483: White Box Test 17 - 2.....	412
Figure 484: White Box Test 17 - 3.....	412
Figure 485: Screenshot of the system - 1.....	414
Figure 486: Screenshot of the system - 2.....	415
Figure 487: Screenshot of the system - 3.....	416
Figure 488: Screenshot of the system - 4.....	417
Figure 489: Screenshot of the system - 5.....	418
Figure 490: Screenshot of the system - 6.....	419
Figure 491: Screenshot of the system - 7.....	420
Figure 492: Screenshot of the system - 8.....	421
Figure 493: Screenshot of the system - 9.....	422

Figure 494: Screenshot of the system - 10.....	423
Figure 495: Screenshot of the system - 11.....	424
Figure 496: Screenshot of the system - 12.....	425
Figure 497: Screenshot of the system - 13.....	426
Figure 498: Screenshot of the system - 14.....	427
Figure 499: Screenshot of the system - 15.....	428
Figure 500: Screenshot of the system - 16.....	429
Figure 501: Screenshot of the system - 17.....	430
Figure 502: Screenshot of the system - 18.....	431
Figure 503: Screenshot of the system - 19.....	432
Figure 504: Screenshot of the system – 20.....	433
Figure 505: Screenshot of the system - 21.....	434
Figure 506: Screenshot of the system - 22.....	435
Figure 507: Screenshot of the system - 23.....	436
Figure 508: Screenshot of the system - 24.....	437
Figure 509: Screenshot of the system - 25.....	438
Figure 510: Screenshot of the system - 26.....	439
Figure 511: Screenshot of the system - 27.....	440
Figure 512: Screenshot of the system - 28.....	441
Figure 513: Screenshot of the system - 29.....	442
Figure 514: Screenshot of the system - 30.....	443
Figure 515: Screenshot of the system - 31.....	444
Figure 516: Screenshot of the system - 32.....	445
Figure 517: Screenshot of the system - 33.....	446
Figure 518: Screenshot of the system - 34.....	447
Figure 519: User Feedback Form - 1	448
Figure 520: User Feedback Form - 2	449
Figure 521: User Feedback Form - 3	449
Figure 522: User Feedback Form - 4	450
Figure 523: User Feedback Form - 5	450
Figure 524: Sample of Filled User Feedback Form – 1	451

Figure 525: Sample of Filled User Feedback Form - 2.....	452
Figure 526: Sample of Filled User Feedback Form - 3.....	452
Figure 527: Sample of Filled User Feedback Form - 4.....	453
Figure 528: Result of User Feedback Form - 1	454
Figure 529: Result of User Feedback Form - 2	454
Figure 530: Result of User Feedback Form - 3	455
Figure 531: Result of User Feedback Form - 4	455
Figure 532: Result of User Feedback Form – 5	456
Figure 533: Result of User Feedback Form - 6	456
Figure 534: Result of User Feedback Form – 7	457
Figure 535: Result of User Feedback Form - 8	457
Figure 536: Result of User Feedback Form - 9	458
Figure 537: Result of User Feedback Form - 10	458
Figure 538: Result of User Feedback Form - 11	459
Figure 539: User Manual – 1	460
Figure 540: User Manual – 2.....	461
Figure 541: User Manual – 3.....	462
Figure 542: User Manual – 4.....	463
Figure 543: User Manual – 5.....	464
Figure 544: User Manual – 6.....	464
Figure 545: User Manual – 7	465
Figure 546: User Manual – 8	466
Figure 547: User Manual – 9	467
Figure 548: User Manual – 10	467
Figure 549: User Manual – 11	468
Figure 550: User Manual – 12.....	469
Figure 551: User Manual – 13.....	470
Figure 552: User Manual – 14.....	471
Figure 553: User Manual – 15.....	471
Figure 554: User Manual – 16.....	472
Figure 555: User Manual – 17	473

Figure 556: User Manual – 18.....	474
Figure 557: User Manual – 19.....	475
Figure 558: User Manual – 20.....	476
Figure 559: User Manual – 21	476
Figure 560: User Manual – 22.....	477
Figure 561: User Manual – 23.....	478
Figure 562: User Manual – 24.....	478
Figure 563: Readings for Future Work – 1	479
Figure 564: Readings for Future Work – 2	480
Figure 565: Readings for Future Work – 3	480
Figure 566: Readings for Future Work – 3	481
Figure 567: Readings for Future Work – 4	482
Figure 568: Readings for Future Work – 5	482
Figure 569: Readings for Future Work – 6	483
Figure 570: Readings for Future Work – 7	483
Figure 571: Readings for Future Work – 8	484
Figure 572: Readings for Future Work – 9	485
Figure 573: Readings for Future Work – 10	485
Figure 574: Readings for Future Work – 11	486

List of Tables

Table 1: Analysis of Similar Project	13
Table 2: High Level Use Case – User Registration	25
Table 3: High Level Use Case – User Login	28
Table 4: High Level Use Case – User View Profile	31
Table 5: High Level Use Case – User Edit Profile	34
Table 6: High Level Use Case – User Book Appointment	38
Table 7: High Level Use Case – User View Appointment	42
Table 8: High Level Use Case – User Cancel Appointment	44
Table 9: High Level Use Case – User View Doctors	47
Table 10: High Level Use Case – User View Report	50
Table 11: High Level Use Case – User View Services	53
Table 12: Black Box Testing Test Plan	69
Table 13: White Box Testing Test Plan	70
Table 14: Integration Testing Test Plan	71
Table 15: System Testing Test Plan	71
Table 16: Black Box Test 1 (Unsuccessful)	73
Table 17: Black Box Test 1 (Successful)	75
Table 18: Black Box Test 2 (Unsuccessful)	76
Table 19: Black Box Test 2 (Successful)	79
Table 20: Black Box Test 3 (Unsuccessful)	81
Table 21: Black Box Test 3 (Successful)	83
Table 22: Black Box Test 4 (Unsuccessful)	84
Table 23: Black Box Test 4 (Successful)	87
Table 24: White Box Test 1	89
Table 25: White Box Test 2	91
Table 26: White Box Test 3	93
Table 27: White Box Test 4	95
Table 28: White Box Test 5	97

Table 29: Integration Test 1	100
Table 30: Integration Test 2	104
Table 31: Integration Test 3	107
Table 32: Integration Test 4	111
Table 33: Integration Test 5	114
Table 34: Integration Test 6	117
Table 35: Integration Test 7	121
Table 36: Integration Test 8	124
Table 37: Integration Test 9	127
Table 38: Integration Test 10	131
Table 39: System Testing 1.....	134
Table 40: High Level Use Case – Admin Login.....	244
Table 41: High Level Use Case – Admin View Users.....	247
Table 42: High Level Use Case – Admin Delete Users	250
Table 43: High Level Use Case – Admin View Doctors.....	253
Table 44: High Level Use Case – Admin Add Doctors	256
Table 45: High Level Use Case – Admin Delete Doctors	260
Table 46: High Level Use Case – Admin Edit Doctors	263
Table 47: High Level Use Case – Admin Add Report.....	267
Table 48: High Level Use Case – Admin Delete Report.....	271
Table 49: High Level Use Case – Admin Edit Report.....	274
Table 50: High Level Use Case – Admin View Reports	278
Table 51: High Level Use Case – Admin Book Appointment	280
Table 52: High Level Use Case – Admin View Appointments	284
Table 53: High Level Use Case – Admin Cancel Appointment	286
Table 54: High Level Use Case – Admin Approve Appointment	289
Table 55: High Level Use Case – Admin Reject Appointment	292
Table 56: High Level Use Case – Admin Finish Appointment	295
Table 57: High Level Use Case – Admin View Services	298
Table 58: High Level Use Case – Admin Add Services.....	301
Table 59: High Level Use Case – Admin Delete Services.....	305

Table 60: High Level Use Case – Admin Edit Services.....	308
Table 61: User Attributes.....	321
Table 62: patientUser Attributes	322
Table 63: Doctor Attribute	322
Table 64: Service Attribute	323
Table 65: Appointment attribute	323
Table 66: Report attribute.....	324
Table 67: Functional Requirement 1	337
Table 68: Functional Requirement 2	338
Table 69: Functional Requirement 3	339
Table 70: Functional Requirement 4	340
Table 71: Functional Requirement 5	341
Table 72: Functional Requirement 6	342
Table 73: Functional Requirement 7	343
Table 74: Performance Requirement	350
Table 75: Safety Requirement.....	350
Table 76: Other Software Quality Attributes	351
Table 77: Black Box Test 5 (Unsuccessful).....	375
Table 78: Black Box Test 5 (Successful).....	377
Table 79: Black Box Test 6 (Unsuccessful).....	378
Table 80: Black Box Test 6 (Successful).....	381
Table 81: Black Box Test 7 (Unsuccessful).....	383
Table 82: Black Box Test 7 (Successful).....	385
Table 83: Black Box Test 8 (Unsuccessful).....	386
Table 84: Black Box Test 8 (Successful).....	388
Table 85: White Box Test 6	391
Table 86: White Box Test 7	393
Table 87: White Box Test 8	394
Table 88: White Box Test 9	396
Table 89: White Box Test 10	398
Table 90: White Box Test 11	400

Table 91: White Box Test 12	401
Table 92: White Box Test 13	403
Table 93: White Box Test 14	405
Table 94: White Box Test 15	407
Table 95: White Box Test 16	409
Table 96: White Box Test 17	411

Table of Abbreviation

IT	Information Technology
ERD	Entity Relation Diagram
UML	Unified Modelling Language
RUP	Rational Unified Process
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
MVT	Model View Template
CDN	Content Delivery Network
VS	Visual Studio
URL	Uniform Resource Locator
JS	JavaScript
CI/CD	Continuous Integration / Continuous Delivery
UI	User Interface
UX	User Experience
SRS	System Requirement Specification
HTTP	HyperText Transfer Protocol
TCP/IP	Transmission Control Protocol / Internet Protocol.

1. Introduction

1.1 Project Description

Dental hygiene is the practice of keeping the mouth clean, and it is often regarded as the most effective method of preventing cavities (dental caries), gingivitis, periodontitis, and other dental problems. It also aids in the prevention of foul breath (halitosis). Regular dental visit is necessary since it helps to keep our teeth and gums healthy. It is recommended that we visit a dental clinic at least once every six months.

A good dental care is necessary if we face any problems related to our teeth and gums. If not treated properly, we may face various difficulties like constant tooth pain, not being able to eat properly and many more. There are millions of people who face daily dental related problems in their lives all over the world. In Nepal, there is no proper facility to get an appointment for our dental check-up. There are thousands of people who visit dental clinics in Nepal every single day. But due to lack of proper resources, people have to wait for hours and hours even if they have a slight problem just to consult with a doctor.

Time is a precious thing and we should not waste it waiting in line for hours. So keeping that in mind, this project is a dental application/management system which lets users to book an appointment for their dental requirements. This will help save people a lot of time and is convenient as well.

1.2 Problem Scenario

People who have a tight schedule don't have enough time to wait in long line for their dental check-up. Sometimes there is no guarantee if their turn will come or not. It is very hectic for people to wait in long lines for a few minutes of consultation as well. Especially now, in times of Covid, it is very risky to stay in crowds as well. Here, the clinics still use old paper style report so there is very high chance that the report will get lost or misplaced. So, if in case, the patient goes to a new dentist, there is a

chance that he/she might have to check their whole teeth again to see what had happened.

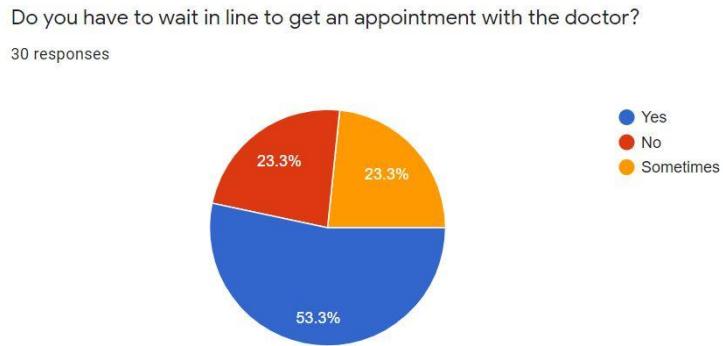


Figure 1: Survey of problem

According to a survey done, it was found that over 50% had to wait in long lines to get an appointment with the doctor whereas around 23% sometimes had to wait in lines.

1.3 Project as a Solution

This Dental Appointment System is important for the country as technology grows fast in the world. The goal of developing this project is to provide better solutions to the problems faced by patients. This website will allow customers to schedule an appointment for their dental check-up from home or from the office. In the present situation, there is no such appointment system available in Nepal. Patients find it difficult to book an appointment for their dental check-up. Either they must wait a very long line in crowd or must call some known doctors to schedule an appointment. Patients can also view their reports in their profile, so they will never lose their dental record and it will be much easier to access it in future if the patients face any problems.

Would you like to use a web application to schedule an appointment with the doctor?
30 responses

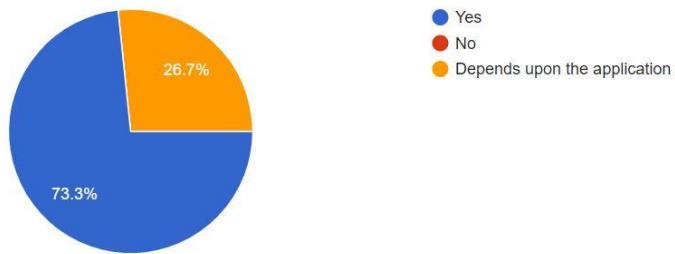


Figure 2: Survey of solution 1

According to the pie chart, it shows that over 70% of them would like to use a web based application to schedule an appointment with the doctor whereas the rest would like to use it depending upon the application.

Do you think this web application will be helpful?
30 responses

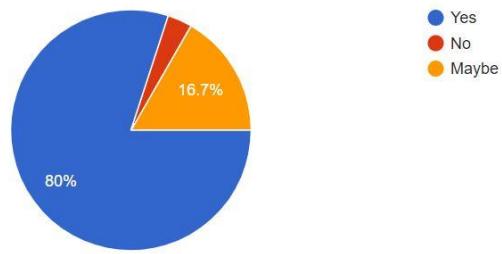


Figure 3: Survey of solution 2

According to the above pie chart of the survey, it was found that 80% of the people thought this web application will be useful whereas 16.7% thought it might be helpful.

At the end of this project, a prototype will be created that will provide a solution to the problem found. This system would be a web-based system where an overall online registration process and appointments will be carried out. Finally, the proposed system would save time for the patients.

1.4 Aims and Objectives

The primary aim of this project is to create a user-friendly web application that allows users to schedule a dental check-up appointment and save time rather than going to the clinic and waiting in line.

The following objectives are pursued in order to achieve this aim:

- To learn about the Python and its Django Framework along with its features.
- To use a relational database to comprehend and implement a database management system in the real world.
- To create a web application that allows people to book an appointment for their dental check-up.
- To work on the project using incremental methodology.
- Make the UI as user-friendly and interactive as possible.

1.5 Structure of the report

1.5.1 Background

- About end user
- Understanding the solution
- Similar Projects
- Comparisons

1.5.2 Development

- Considered Methodologies
- Selected Methodology
- Phases of methodology
- Survey Results
 - Pre-Survey Results
 - Post-Survey Results
- Requirement Analysis
- Design

- Implementation

1.5.3 Testing and Analysis

- Test Plan
 - Unit Testing, Test plan
 - System Testing, Test Plan
- Unit Testing
- System Testing
- Critical Analysis

1.5.4 Conclusion

- Legal, Social and Ethical Issues
 - Legal Issues
 - Social Issues
 - Ethical Issues
- Advantages
- Limitations
- Future Work

2. Background

2.1 About the end users

2.1.1 Discussion with client

[Click here to see the summary of the discussion with the client.](#)

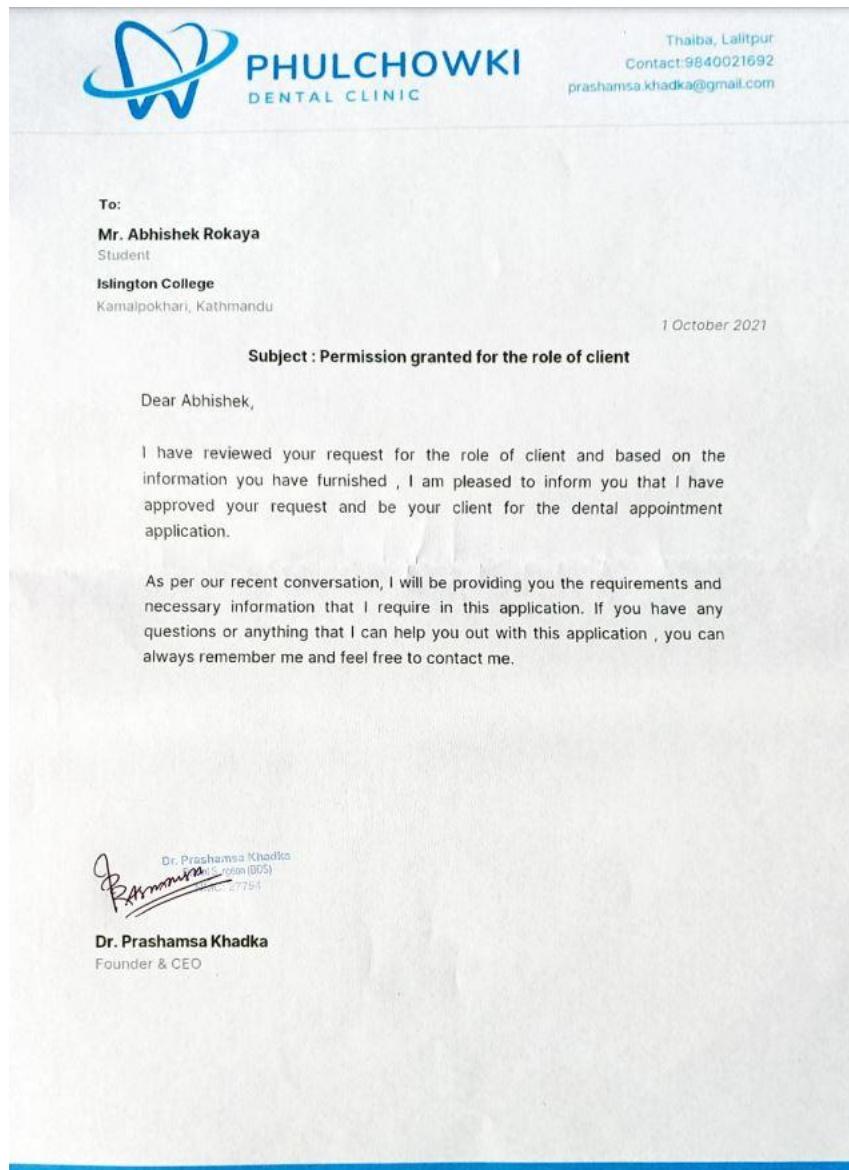


Figure 4: Client Role Permission

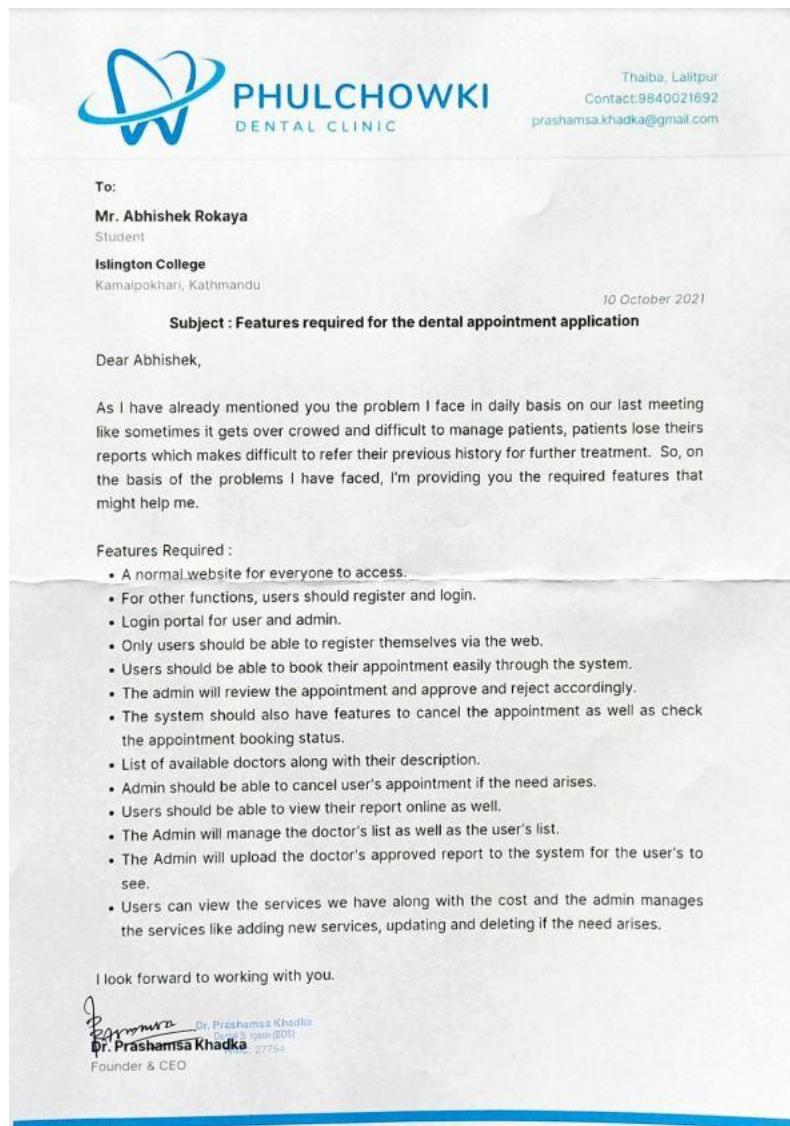


Figure 5: Client's Requirements

2.1.2 End Users

The end users for the application will be multiple people who wants to book a dental appointment. The targeted users are the general public looking for an easier way to schedule a dental appointment with the doctor.

2.2 Understanding the solution

2.2.1 Overview of the system

The system is web-based application named "Dental Appointment System" for "Phulchowki Dental Clinic" located in Thaiba, Lalitpur. The main objective of the application is to provide an online platform for users to book their dental appointment based on their time and convenience. It is a simple application in which user can also view their reports of their appointment visit and select their desired doctors and services.

2.2.2 Technology

Django Python Framework:

For my project, I have decided to use the Django Python Framework. The reason I chose this framework is because it has better CDN connectivity and content management. It also has a faster processing. Django uses the MTV architecture due to which the transmission over the internet is easier and faster.

HTML, CSS, JavaScript and Bootstrap

For the frontend of the web application, I have decided to use HTML, CSS, JavaScript and Bootstrap, which is the framework of CSS.

SQLite3

For the database, I have decided to use SQLite to store the database. Django uses SQLite database by default so it is easier to use this database than others which is why I chose this.

Visual Studio Code IDE:

The IDEs that I will be using to develop the application is Visual Studio Code. Microsoft's Visual Studio Code (often known as VS Code) is a free open source text editor. Windows, Linux, and macOS are all supported by VS Code. Although the editor is minimal, it contains several strong capabilities that have helped VS Code become one of the most popular development environment tools in recent years.

Hardware Requirements

For the project to run, it doesn't need any high-end hardware device. Just a normal working device with a proper internet connection will work fine.

2.3 Similar Projects

2.3.1 Dent-inn

Dent-inn is a dental clinic located at Durbar-marg of Kathmandu, Nepal. The application is a web based application. In this application, you can fill out an online appointment form. The form requires name, phone, email, date, time, preferred doctor and message. However, you cannot select a particular service from the appointment from itself. There is also a section where you can view the in house doctors. In this application, you do not need register as a user. Anyone can just visit the website and book an appointment. There is no section for uploading report meaning you can only book an appointment for your dental check-up and nothing more.

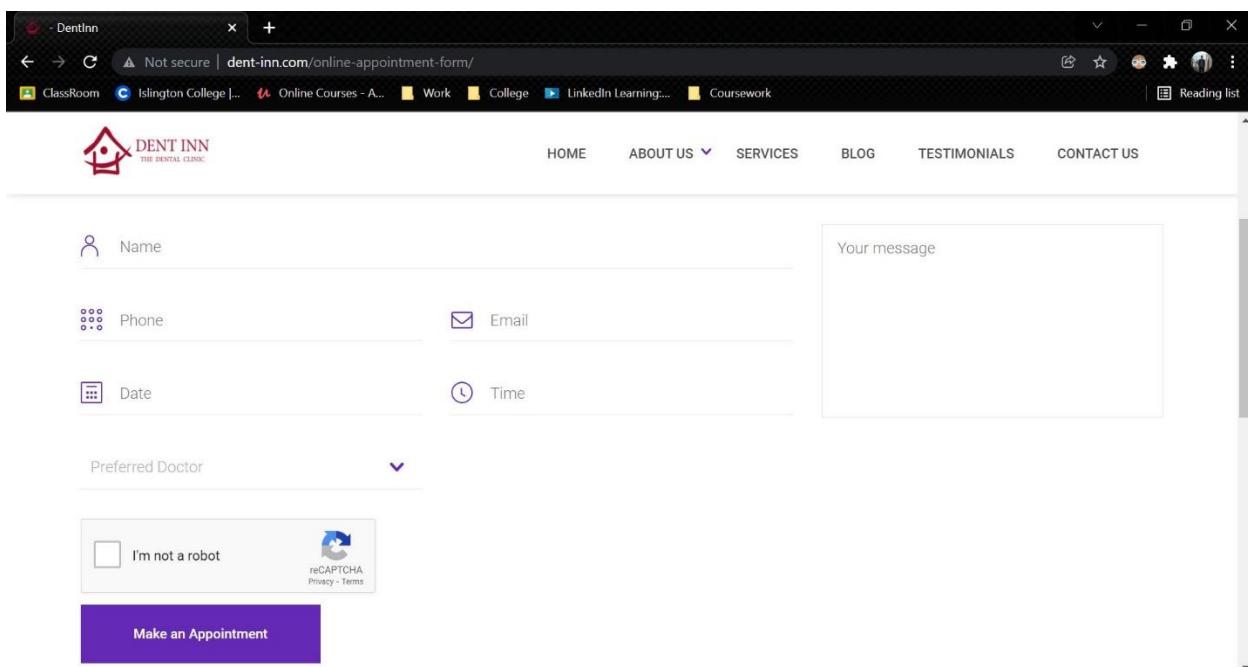

 A screenshot of a web browser displaying the Dent-inn online appointment form. The URL in the address bar is dent-inn.com/online-appointment-form/. The page has a dark header with the Dent-inn logo and navigation links for HOME, ABOUT US, SERVICES, BLOG, TESTIMONIALS, and CONTACT US. Below the header, there are input fields for Name, Phone, Date, and Time. A large text area labeled "Your message" is on the right. A dropdown menu for "Preferred Doctor" is visible. At the bottom, there is a reCAPTCHA verification box with the text "I'm not a robot" and a "Make an Appointment" button.

Figure 6: Dent-inn

Link to the website: <http://www.dent-inn.com/>

2.3.2 DentaLife

DentaLife is also a dental clinic located at Jamal, Kathmandu. Although the website is similar to Dent-inn, the appointment section of this web is quite different. It used google forms in order to collect the appointment details. This web uses no backend.

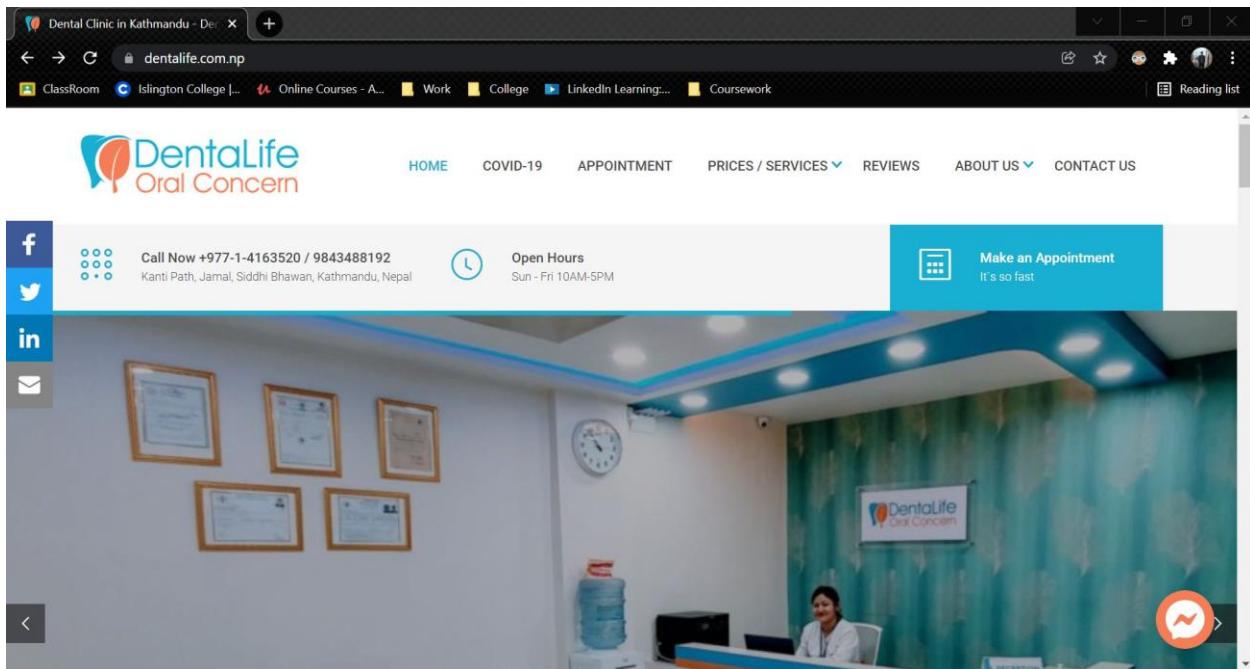


Figure 7: DentaLife

Link to the website: <https://dentalife.com.np/>

2.3.3 Kumari Dental Care

Kumari Dental Care is a dental clinic located at Jhamsikhel, Lalitpur. It is also a web based application where any user can book an appointment. In this web application, the first name, last name, email, needs and message. In this clinic, only one doctor is present. There is no login and register feature here as well. There is no report option here as well, meaning user will only use this website once for their appointment.

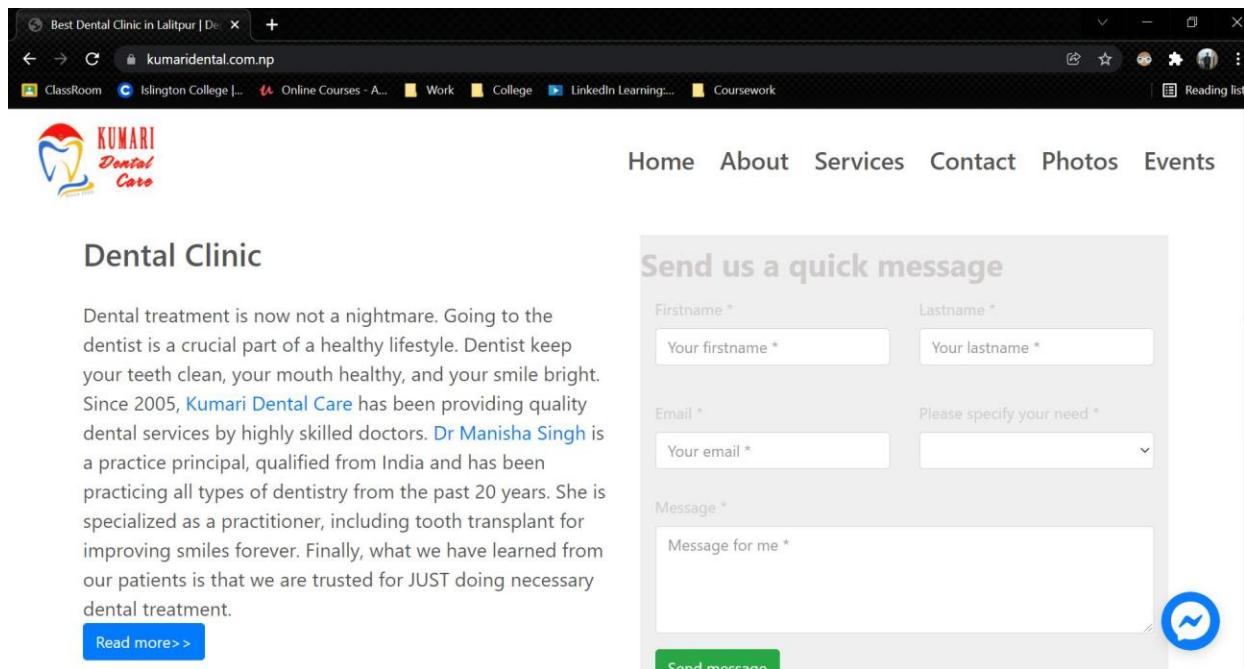


Figure 8: Kumari Dental Care

Link to the website: <https://kumaridental.com.np/index.html>

2.4 Comparisons

The table below shows the comparison of different web based applications for the online appointment booking system.

S. No	Features	Dent-inn	DentaLife	Kumari Dental	My System
1.	Online Appointment Booking Module	✓	✓	✓	✓
2.	Doctor Selection	✓	✓	✗	✓
3.	Service Selection	✗	✗	✓	✓
4.	View Report	✗	✗	✗	✓
5.	View Doctors	✓	✗	✓	✓
6.	Login	✗	✗	✗	✓
7.	Register	✗	✗	✗	✓
8.	Day and Time Selection	✓	✓	✓	✓
9.	Description	✓	✓	✓	✓

Table 1: Analysis of Similar Project

2.5 Critical evaluation of the comparisons

Comparing the systems, it is found that none of the other similar system have a login or register feature. 2 out 3 website uses Google form as their appointment booking system to collect the user's data. 2 out of 3 website also had the option to select which doctor they wanted to select to get an appointment with. Only one of the three website had the option to select what service the user wanted to get. Others used the description field to write what they wanted. All of the website does have the date and time selection with a place for extra description of their appointment. However, it appears none of the similar websites have the feature to view their report of the visit. All of the similar system focuses only on gaining patient for the appointment and does not seem to care afterwards. However, in my system, we keep track of every visits of the users along with their reports of every visit. In my system, we also show the details of the doctors with the qualification along with the price of the service. This system is very much transparent compared to others and it is for free meaning users do not have to pay any price to use this system.

3. Development

3.1 Considered Methodologies

3.1.1 RUP Methodology

Justification for considering the methodology:

- The development process can be done back and forth
- End users have the ability to provide continuous feedback.
- This model focuses on the quality of the software with accurate documentation.

Reason for not selecting this methodology

- It relies strongly on expert and skilled team members.
- The process of development is far too complicated and disorderly.
- The focus on integration throughout development may have an impact on testing and other quality assurance phases.

[Click here to see more about RUP Methodology.](#)

3.1.2 Prototype Methodology

Justification for considering the methodology:

- There is room for improvement, which implies that new requirements may be addressed quickly.
- It aids both developers and users in gaining a better understanding of the system.
- The integration needs are well defined, and the deployment channels are chosen early on.
- It has the ability to actively include users in the development process.

Reason for not selecting this methodology

- Because to the constantly changing client needs, it has weak documentation.
- The system's complexity may rise as a result.
- There may be an excessive amount of variety in requirements.

[Click here to view more about Prototype Methodology.](#)

3.2 Selected Methodology

Incremental Methodology

Reason for selecting this methodology:

- Produces usable software rapidly and early in the software development life cycle.
- This model is more adaptable, making changes to scope and needs less costly.
- A smaller iteration makes testing and debugging easier.
- Customers can respond to each built in this model.

[Click here to see the detailed reason to select this methodology.](#)

3.3 Phases of Methodology

There are 4 phases involves in the Incremental model and they are enumerated below:

3.3.1 Requirement Analysis:

The first stage in the incremental model is to analyse the product's problem situation and identify the specifications. The functional requirements of the system are understood by the requirement analysis team. This procedure is essential for designing apps utilizing the incremental model. All of the system's criteria are obtained during this procedure through study of publications, books, and web sites, as well as surveys and interviews with end users.

The requirements for my system were acquired through a survey and analysed based on how the end user would want to utilize the program and what features they desired.

3.3.2 Design and Development:

Following the collection of all specifications, the major functions and features to be employed in the software are determined. Risks are identified during this phase, as are the instruments that will be used. During this phase, wireframes, use case diagrams, collaboration diagrams, sequence diagrams, and class diagrams are also created. The functionality of the system and the development technique have been successfully designed.

After analysing the requirements, the general design of the project began with the creation of the system's wireframes. Following the creation of the wireframe, the class diagram and use cases for the overall system were established. Following the creation of the class diagram, the ERD for the system's database was created, and the system's code/development was initiated.

3.3.3 Testing:

After developing the system's functionality, it is tested in this phase to ensure that the developed system is fully functional and error-free. This phase assesses the consistency of all existing functionalities as well as the addition of new features. A variety of testing methods are utilized to test the activity of each task.

Following the end of the system's development, different forms of testing were performed, including unit testing, system testing, Blackbox testing, and Whitebox testing.

3.3.4 Implementation:

When the testing phase is finished, the project goes on to the implementation phase. It includes both the final coding developed during the development phase and the testing of the functionality during the testing phase. After this phase is completed, the number of working products is improved and updated all the way to the final system product. This phase reviews and discusses the entire system, as well as the features and modifications that will be required in future iterations.

After testing is completed, the system's final coding is accomplished, and the focus shifts to enhancing the system, fixing errors, and adding new features.

3.4 Survey

3.4.1 Pre-Survey Results

A survey was conducted during the requirement analysis phase where details about the scenario people face and what might make the application better was recorded. A total of 30 people filled out the survey and the details that was gathered are as follows:

1. 43.3% of them have their dental check-up in over a year, 26.7% of them in a year, 26.7% in every 6 months and 3.3% in every 3 months.
2. 53.3% have to wait in line to get an appointment, 23.3% sometime have to wait in line and 23.3% of them doesn't have to wait in line.
3. 86.7% have never used an online appointment booking system whereas 13.3% have.
4. 73.3% said they would like to use a web based application to schedule an appointment with the doctor whereas the remaining 26.7% said it would depend on the application.
5. 80% thinks this application will be helpful, 16.7% thinks the application might be helpful whereas 3.3% thought it will not be helpful.
6. 76.7% thinks having an online report will be useful whereas 23.3% thinks it might be useful.

Link to the survey: <https://forms.gle/x1DkhUFqgoDTmZ886>

[Click here to see the pre-survey results and responses](#)

3.4.2 Post-Survey Results

After the completion of the development phase, another survey was conducted which recorded the response of people to how the system was. A total of 55 people filled out the survey and the results that were gathered are as follows:

1. 74.5% of the total people rated 5 star for the UI of the application. 23.6% rated it as 4 star and 1.8% rated it as 3 star.
2. 87.3% of the total people rated 5 star for the functionality of the system whereas the rest 12.7% rated it as 4 star.
3. 49.1% of them liked viewing the report function the most among others. 40% liked the Booking of Appointment most whereas the rest 10.9% liked the browsing of doctor the most.
4. 100% of the people saw themselves as using this application.
5. 100% of the people would recommend this application to others.
6. Most of the people would like to see payment and chat system in the future.

Link to the survey: <https://forms.gle/Y7jvWTMVGvhn8wCr7>

[Click here to see the post-survey results and responses](#)

3.5 Requirement Analysis

3.5.1 Feature Requirements

3.5.1.1 Functional

- User can register and login in the system.
- User can schedule an appointment for their booking.
- User can cancel their booking.
- Admin can approve, reject or cancel the user's appointment.
- Admin can add report of the finished appointment.
- Admin can add, update and delete the doctors and services.

3.5.1.2 Non-Functional

- User can view their report of their finished appointment.
- User can view the list of doctors along with their details.
- User can view the services available along with the cost and description.

3.5.1.3 Usable

- There is a navigation bar available in the side for users to switch pages.
- Unregistered users can send their queries through contact form in home page.
- User can search and sort their appointment and report list.
- User can download a pdf of their report.

3.5.2 Hardware and Software Requirements

Since this project is a web application, there are no required hardware requirements for loading or accessing it. The following are some of the requirements for operating this web application:

- Good and stable Internet Connection.
- Any working laptop or computer.
- Web Browsers like Google Chrome, Microsoft Edge, etc.

[Click here to see the SRS Document of this project.](#)

3.6 Design

For the design part of the project, UML Diagrams, ER Diagram and Class Diagrams are created. Use Case Diagram, Collaboration Diagram, Activity Diagram and Sequence Diagrams are created for each of the function of the system. However, an overall use case diagram is also created for the overall system. ER Diagram and Class Diagram is also created for the overall system. The designs are given below:

[Click here to see the description of the designs.](#)

[Click here to see the Gantt chart.](#)

[Click here to see the Work Breakdown Structure.](#)

[Click here to see the algorithm and the flowchart.](#)

[Click here to see the Final ERD and its creation.](#)

[Click here to see the Class Diagram.](#)

[Click here to see the wireframes.](#)

[Click here to see the functional UML Diagrams of the admin side.](#)

3.6.1 Overall Use Case Diagram

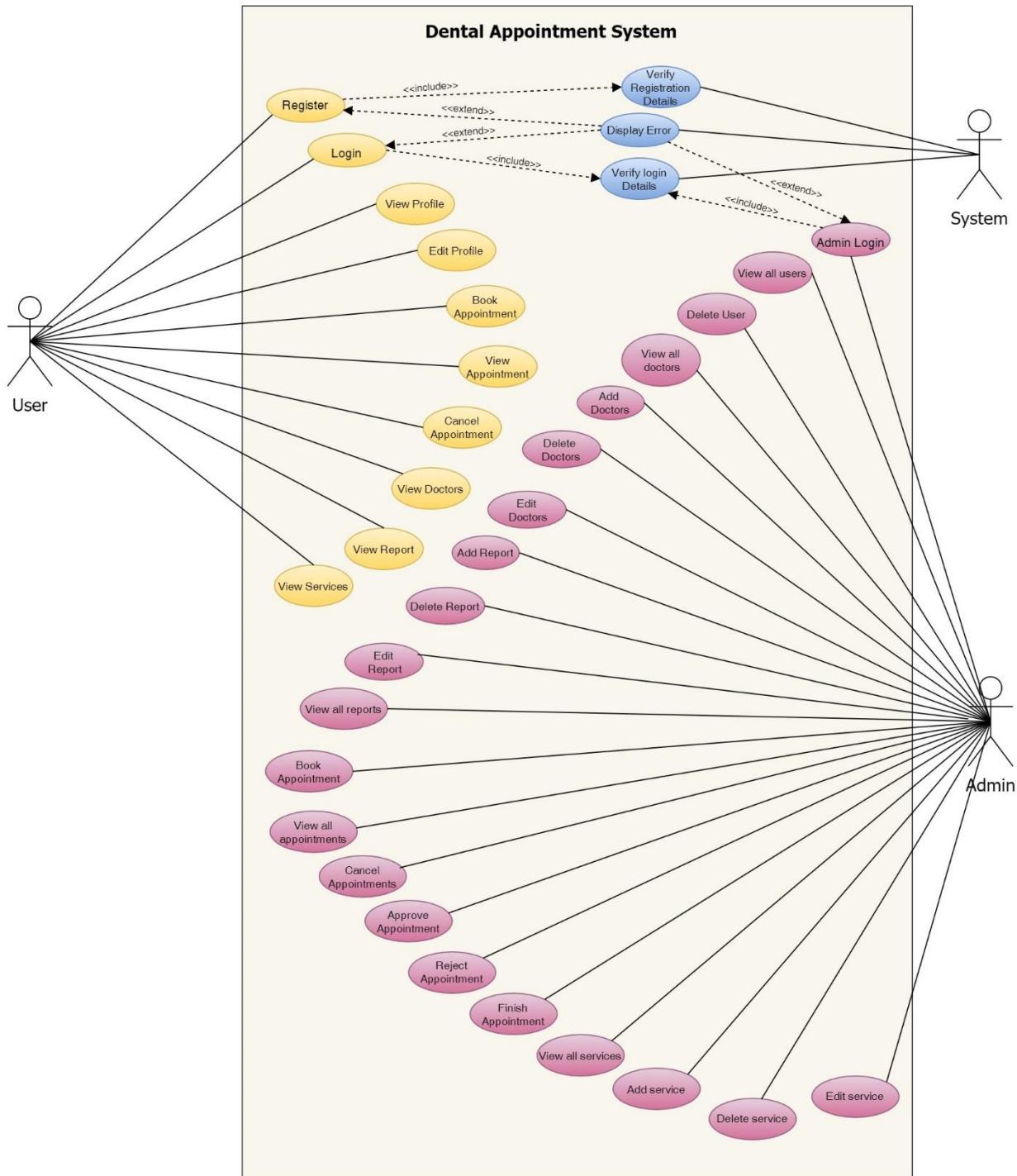


Figure 9: Overall Use Case Diagram

[Click here to view the creation of Use Case Diagram.](#)

3.6.2 Functional UML Diagrams

User

1. Register to the system

High Level Use Case Description

Use Case:	Register
Actor:	User
Description:	New users that are not registered in the system register into system. The registration is then verified by the system.

Table 2: High Level Use Case – User Registration

Use Case Diagram

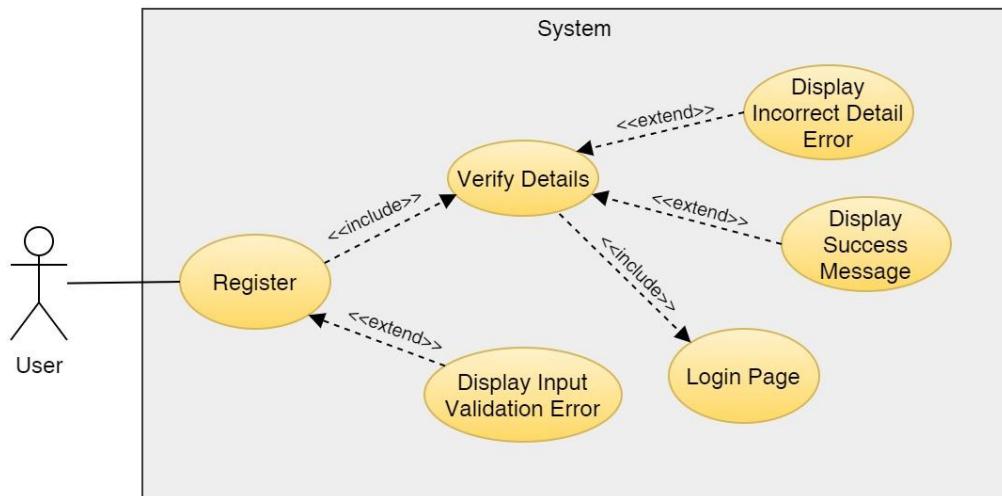


Figure 10: Use Case Diagram – User Registration

Collaboration Diagram

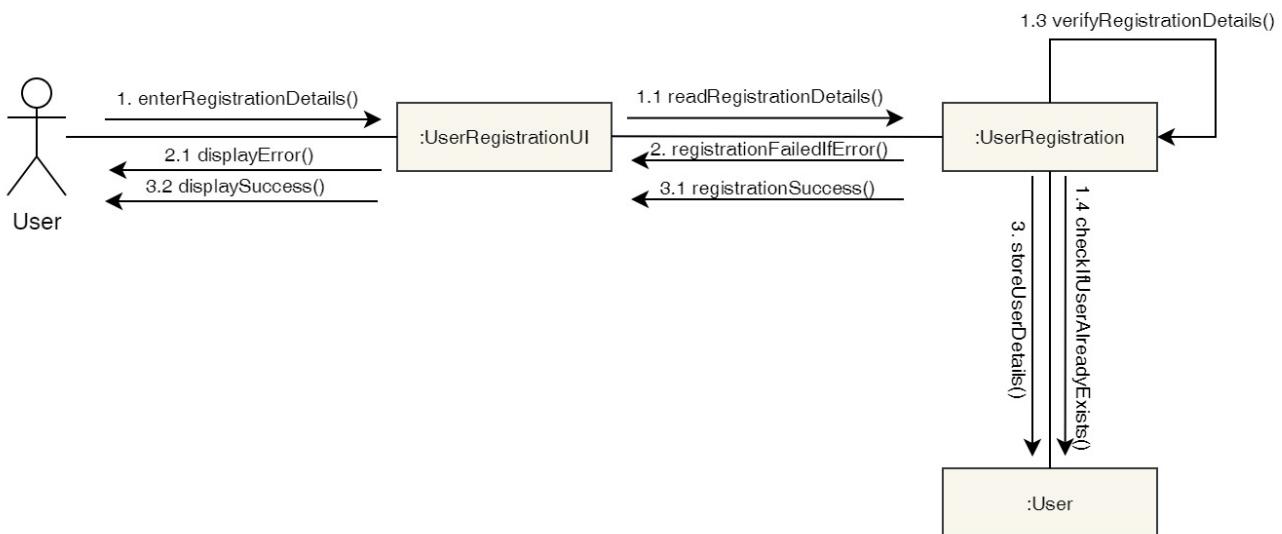


Figure 11: Collaboration Diagram – User Registration

Activity Diagram

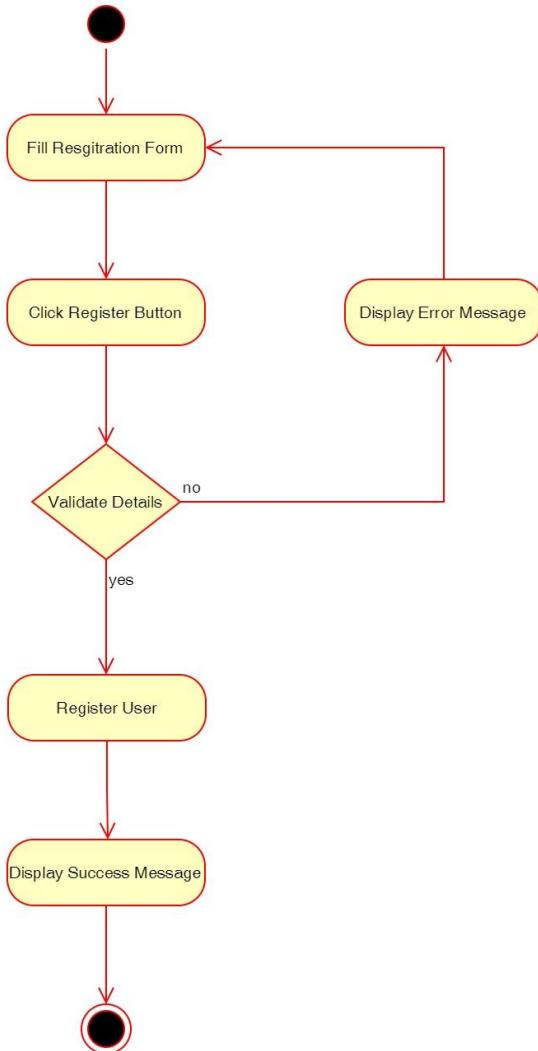


Figure 12: Activity Diagram – User Registration

Sequence Diagram

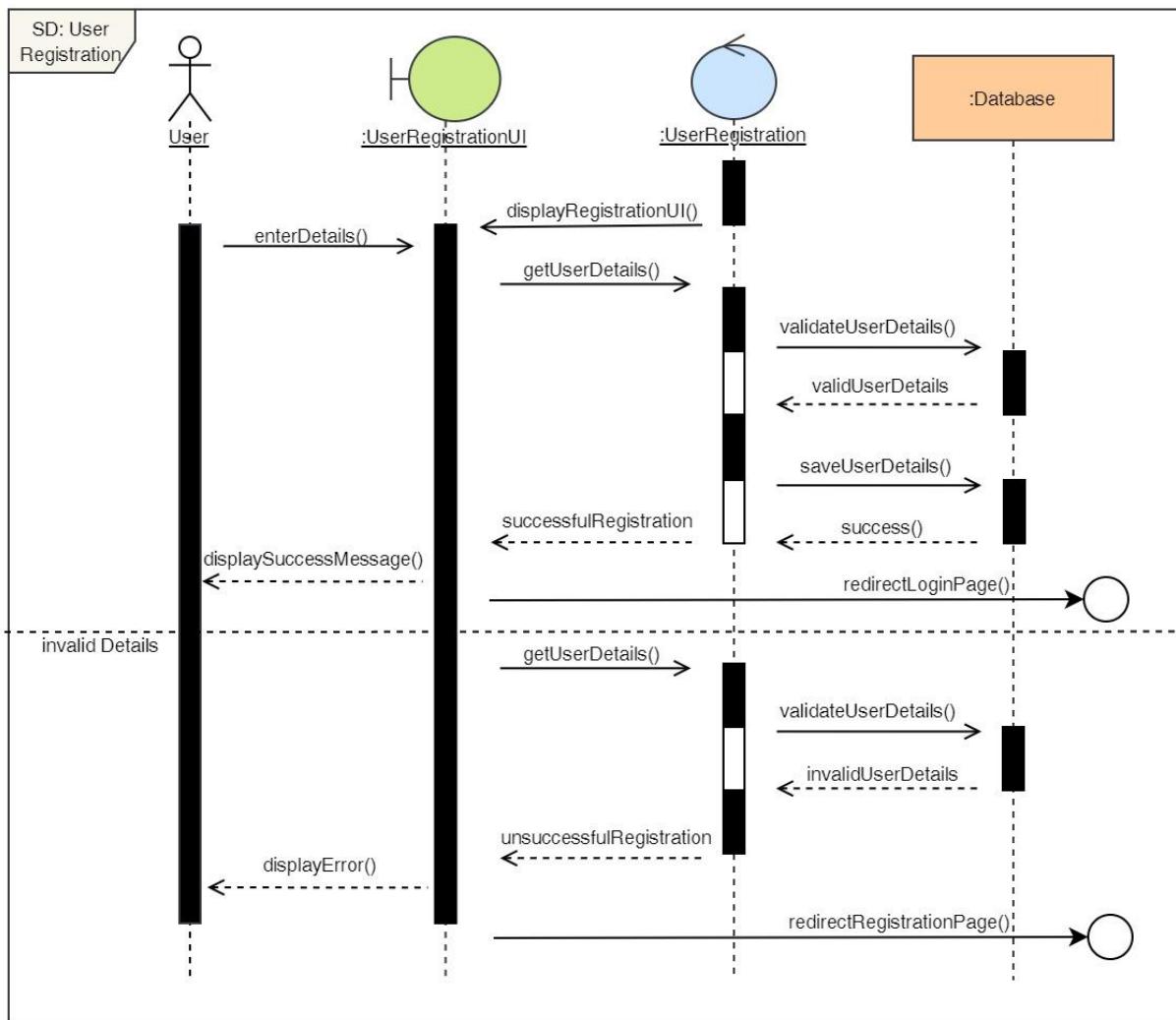


Figure 13: Sequence Diagram – User Registration

2. Login to the system

High Level Use Case Description

Use Case:	Login
Actor:	User
Description:	In order to use the system and use its features, the user must log in to the system. The login details is then verified by the system.

Table 3: High Level Use Case – User Login

Use Case Diagram

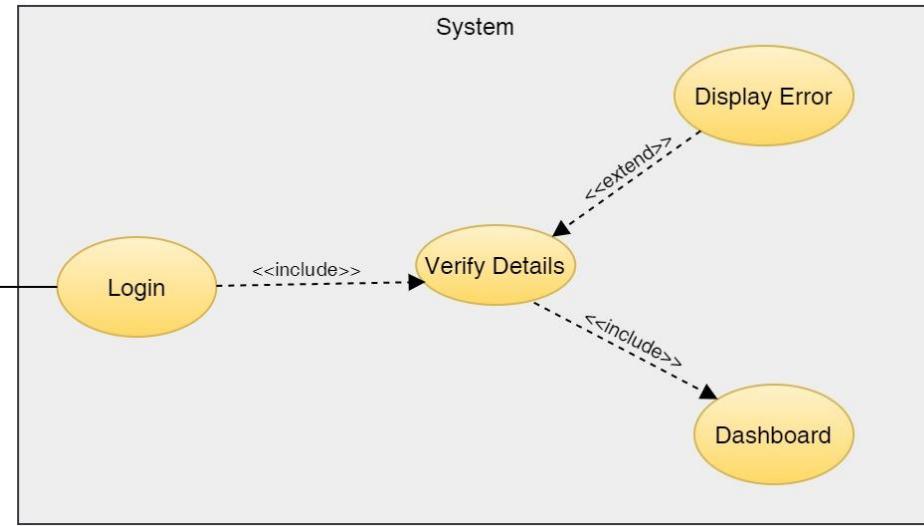


Figure 14: Use Case Diagram – User Login

Collaboration Diagram

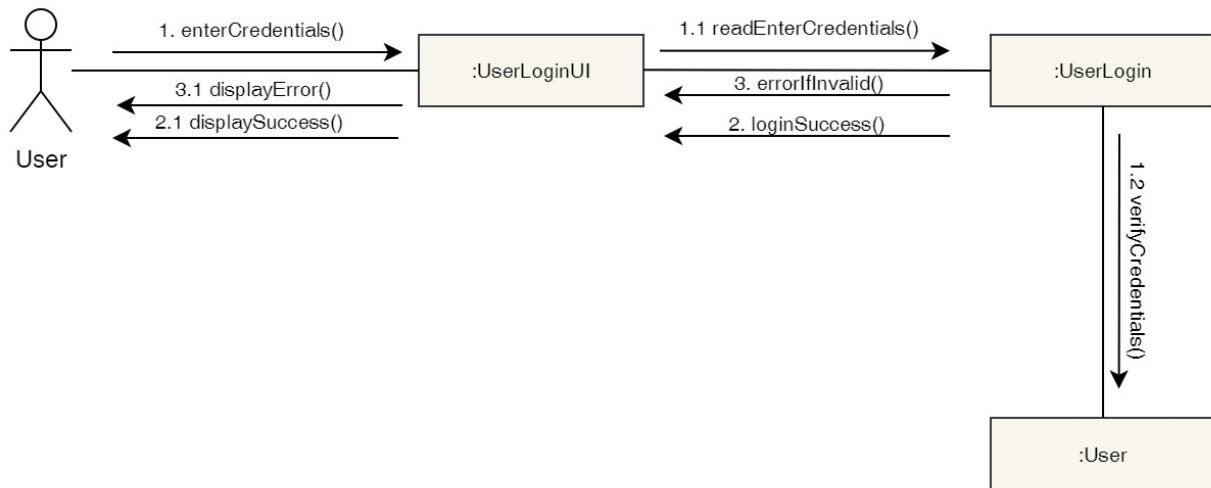
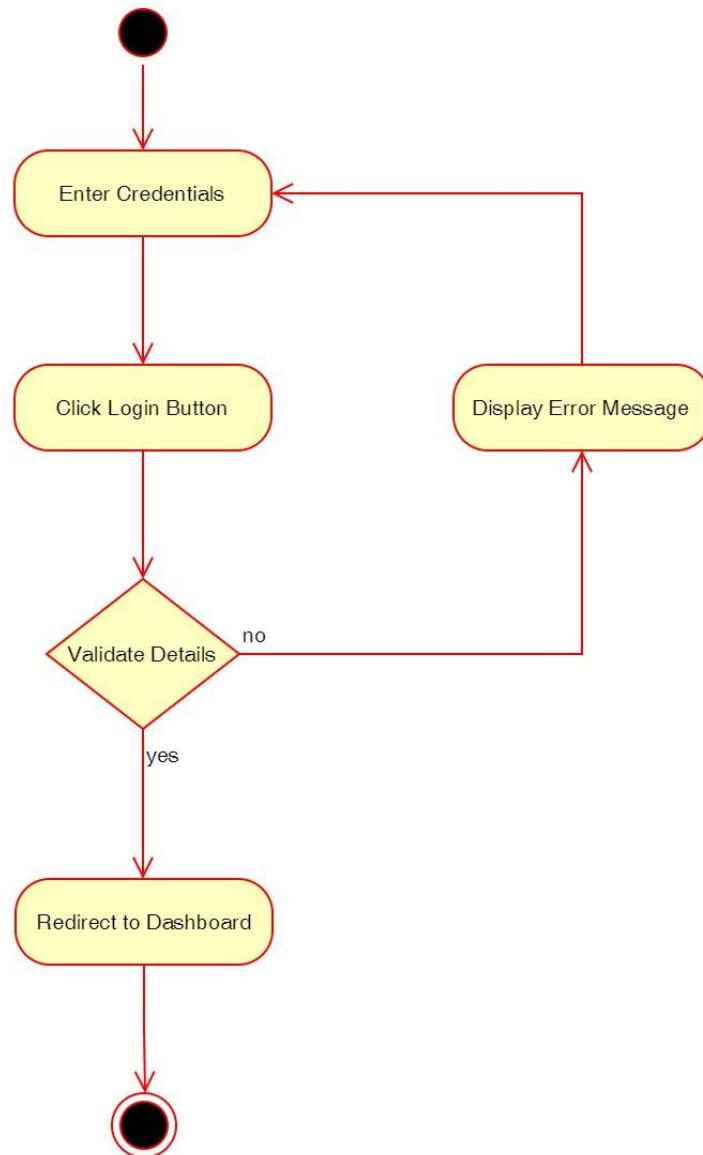


Figure 15: Collaboration Diagram – User Login

Activity Diagram*Figure 16: Activity Diagram – User Login*

Sequence Diagram

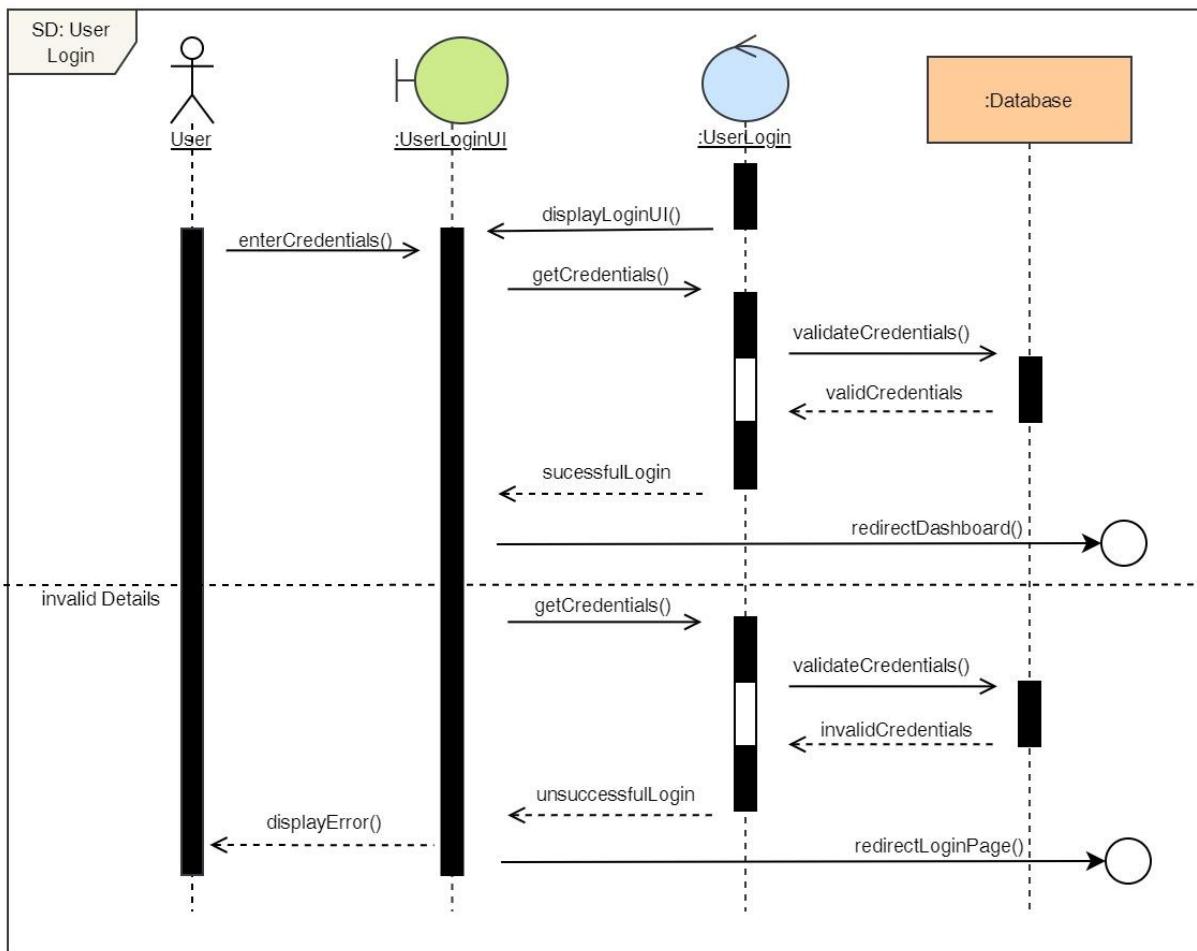


Figure 17: Sequence Diagram – User Login

3. View Profile

High Level Use Case Description

Use Case:	View Profile
Actor:	User
Description:	The user can view their profile where they can see their details which they input.

Table 4: High Level Use Case – User View Profile

Use Case Diagram

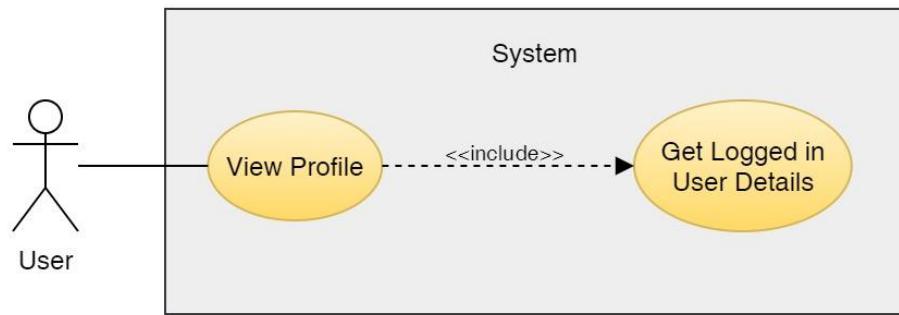


Figure 18: Use Case Diagram – User View Profile

Collaboration Diagram

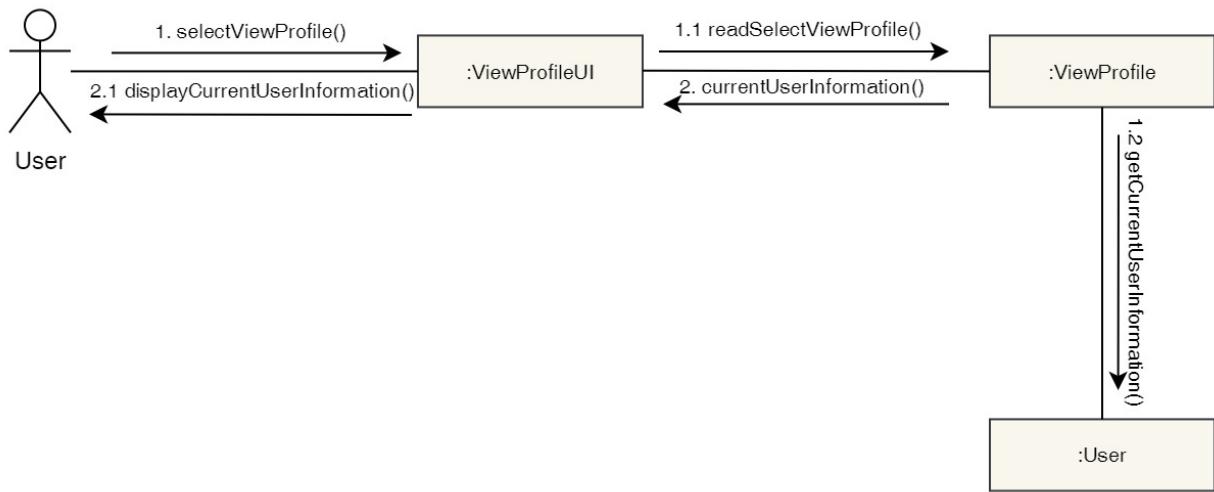


Figure 19: Collaboration Diagram – User View Profile

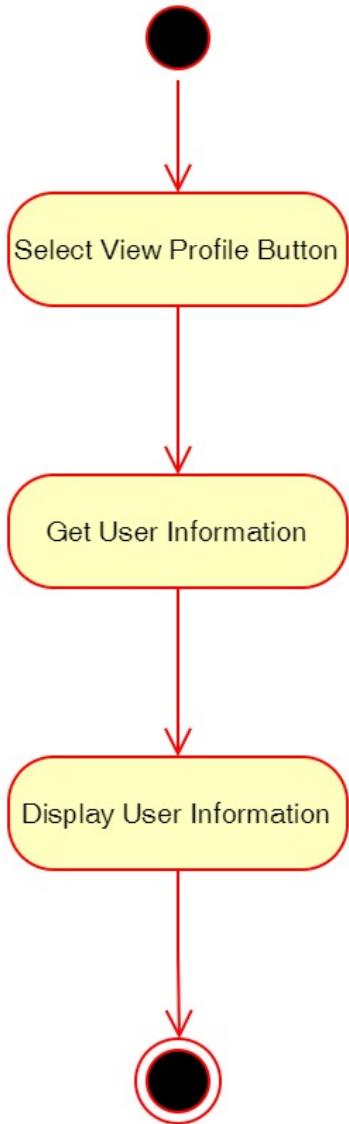
Activity Diagram

Figure 20: Activity Diagram – User View Profile

Sequence Diagram

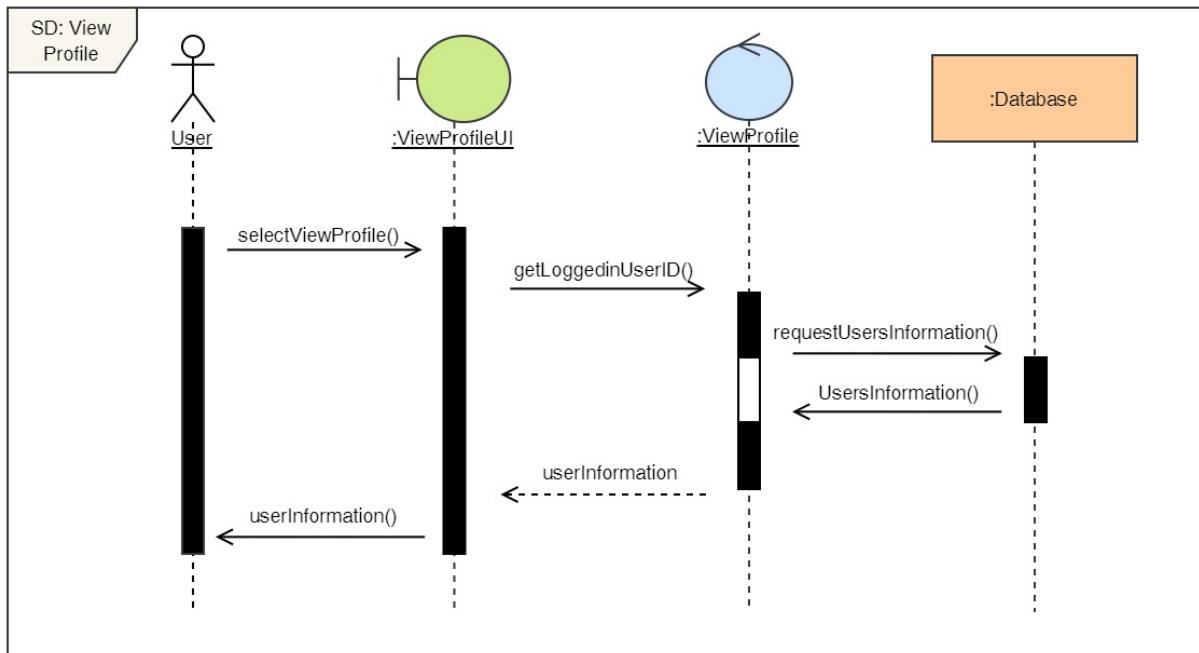


Figure 21: Sequence Diagram – User View Profile

4. Edit Profile

High Level Use Case Description

Use Case:	Edit Profile
Actor:	User
Description:	The user can update their profile details if they have changed anything from what they previously entered.

Table 5: High Level Use Case – User Edit Profile

Use Case Diagram

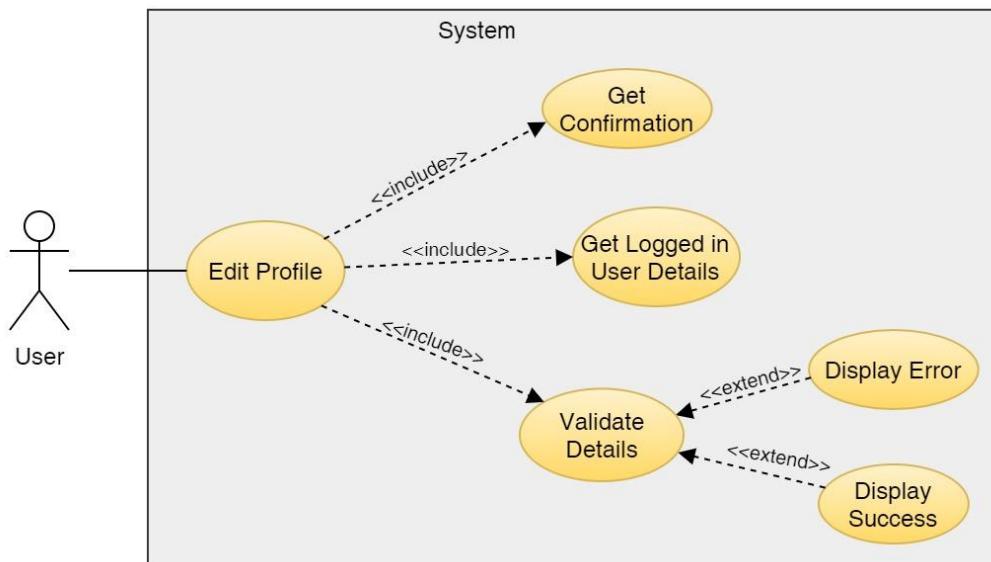


Figure 22: Use Case Diagram – User Edit Profile

Collaboration Diagram

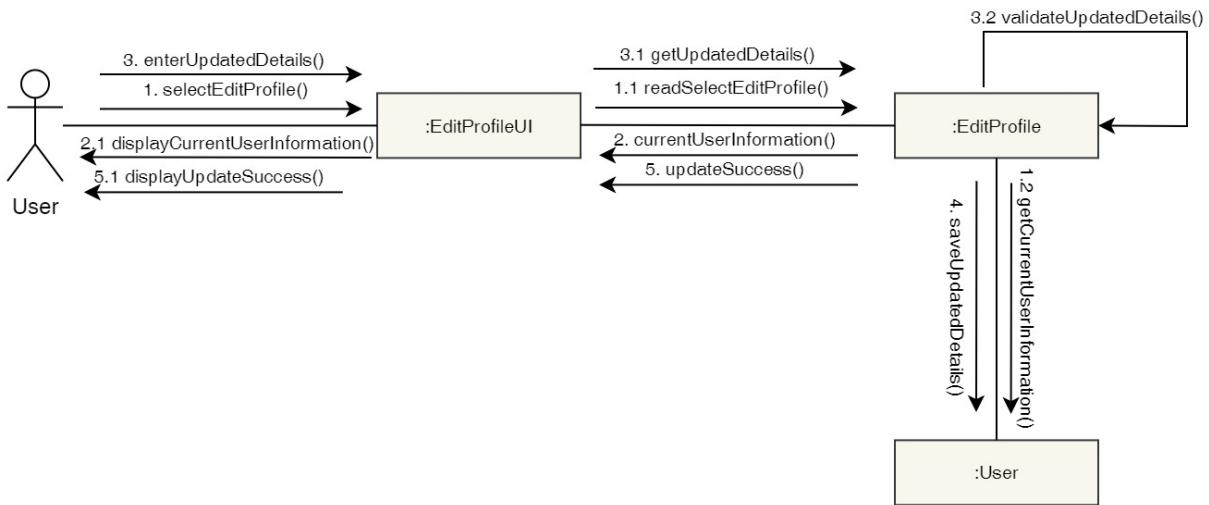


Figure 23: Collaboration Diagram – User Edit Profile

Activity Diagram

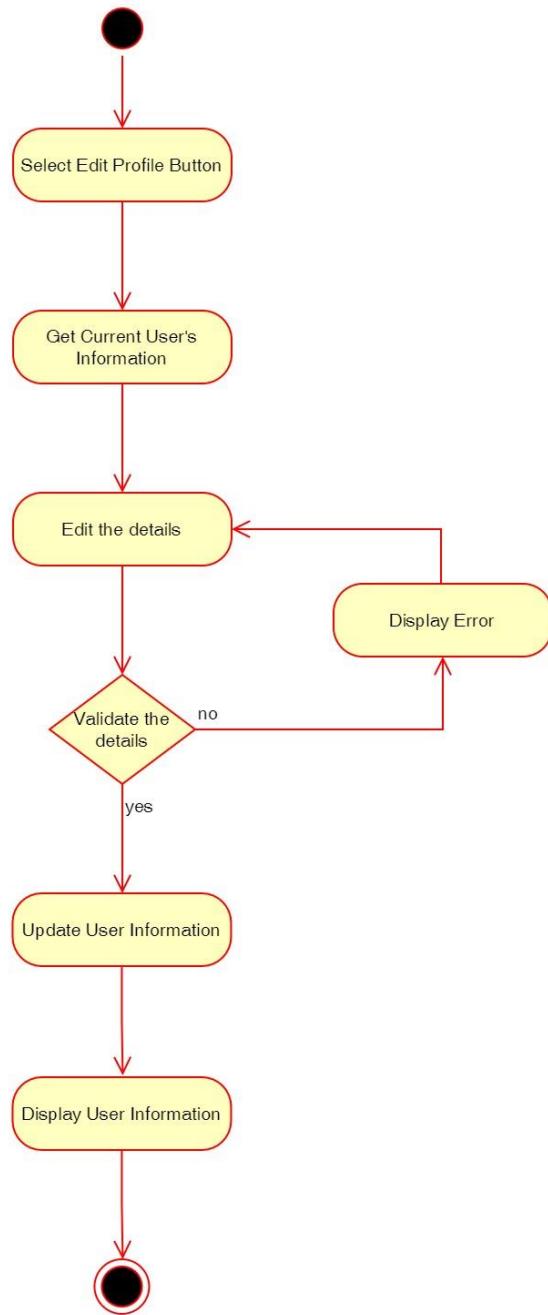


Figure 24: Activity Diagram – User Edit Profile

Sequence Diagram

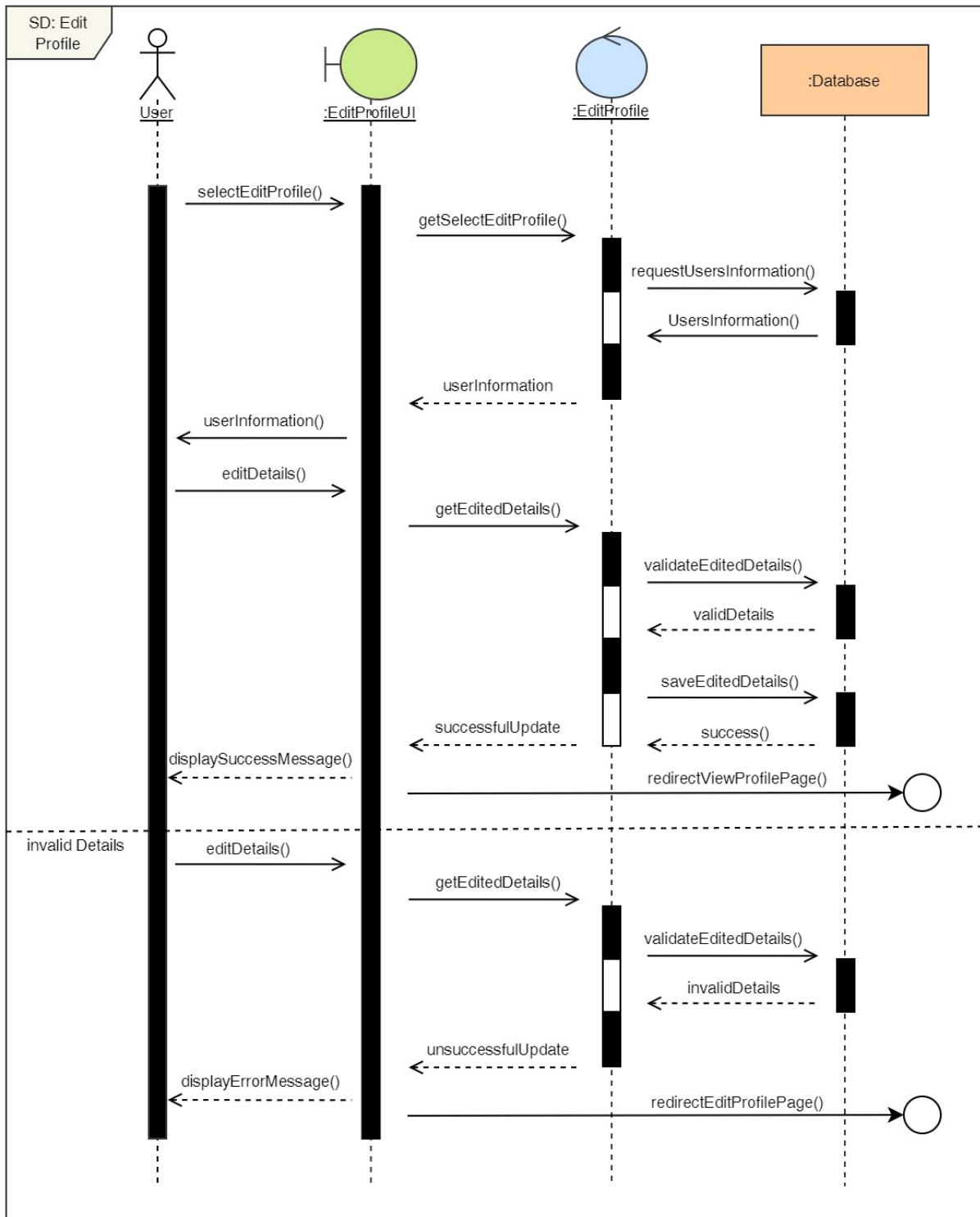


Figure 25: Sequence Diagram – User Edit Profile

5. Book a Dental Appointment

High Level Use Case Description

Use Case:	Book Appointment
Actor:	User
Description:	User can book appointment for any dental services by selecting the desired date, time, doctor and service.

Table 6: High Level Use Case – User Book Appointment

Use Case Diagram

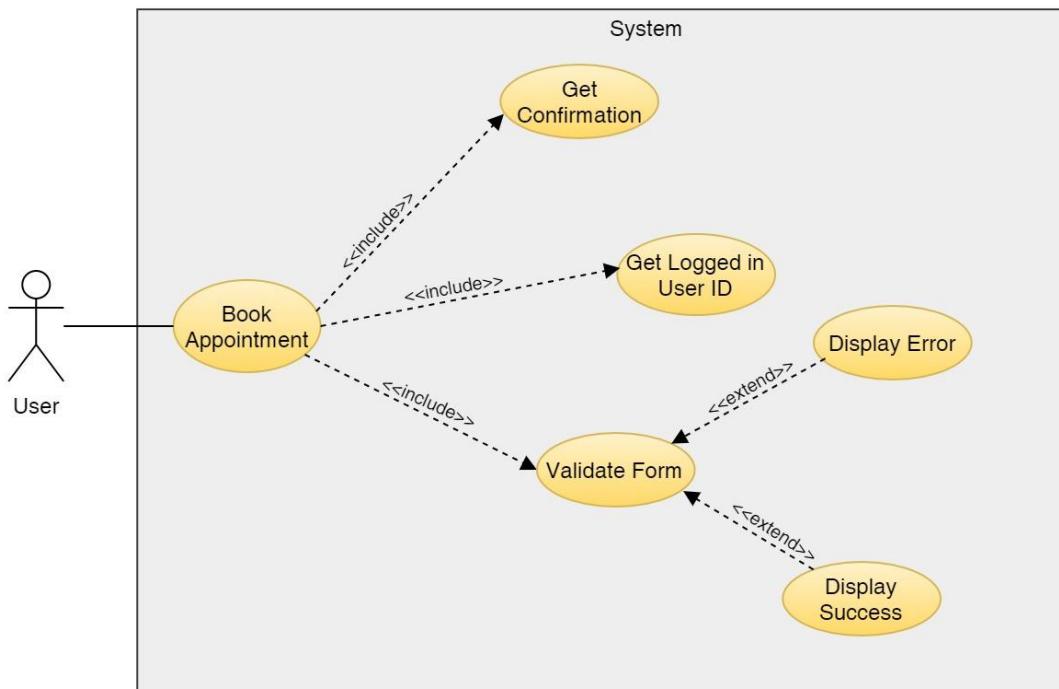


Figure 26: Use Case Diagram – User Book Appointment

Collaboration Diagram

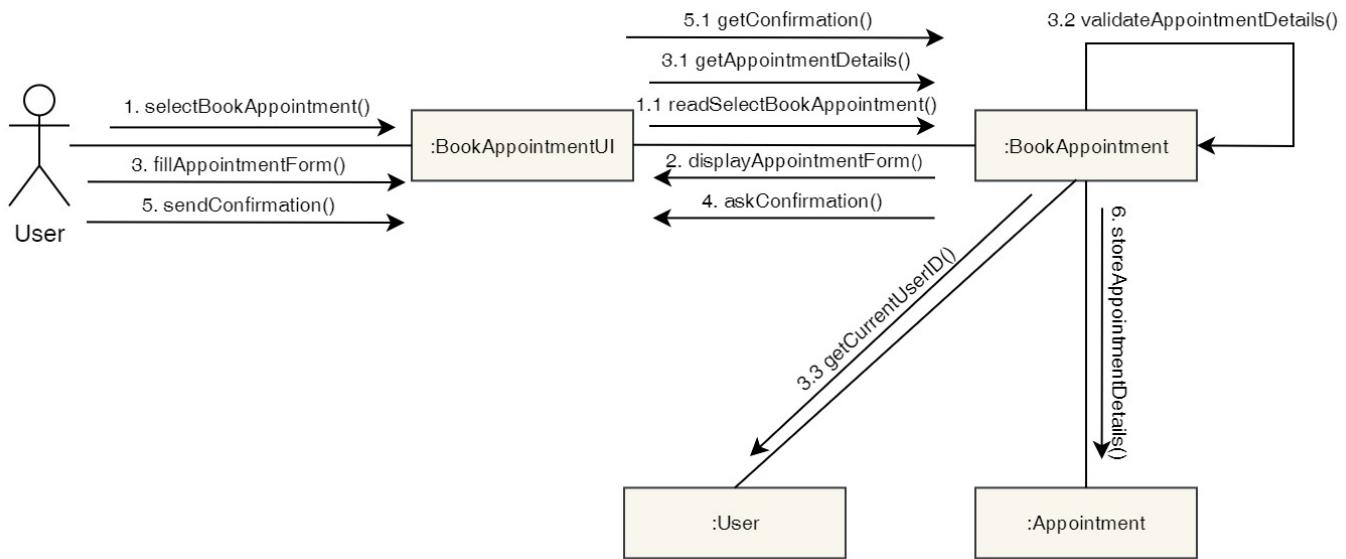


Figure 27: Collaboration Diagram – User Book Appointment

Activity Diagram

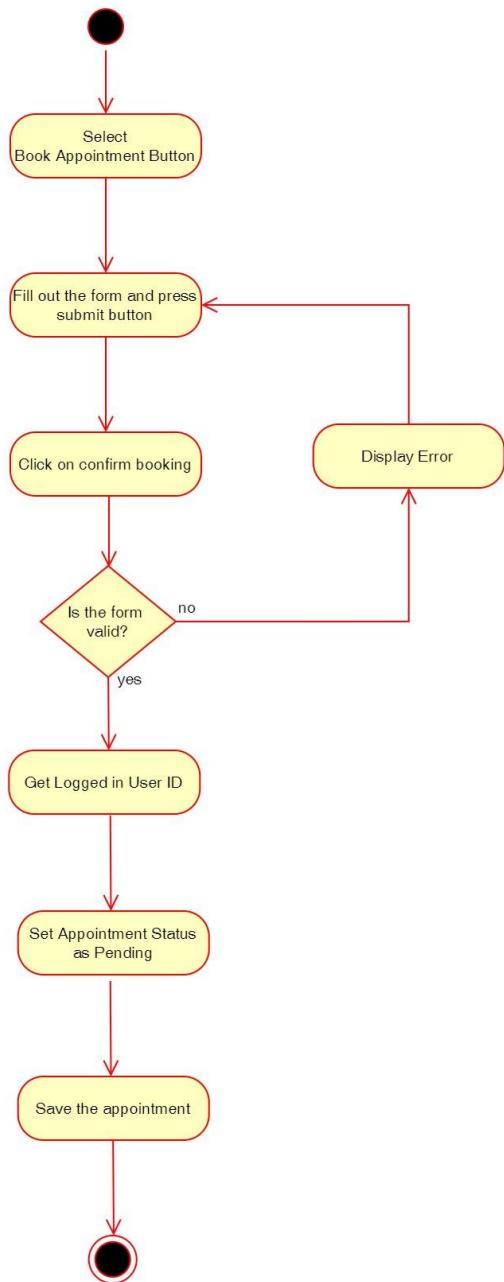


Figure 28: Activity Diagram – User Book Appointment

Sequence Diagram

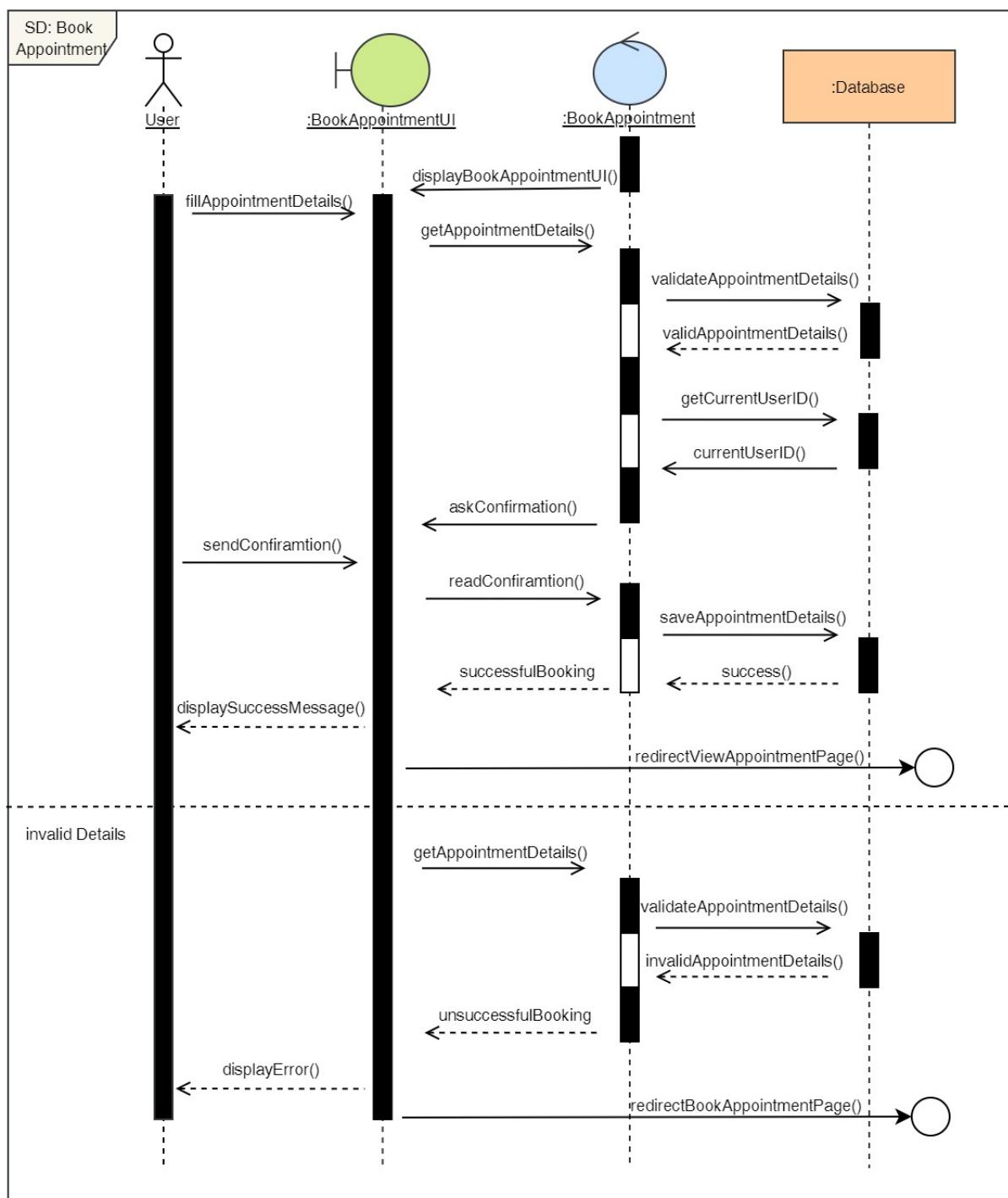


Figure 29: Sequence Diagram – User Book Appointment

6. View Appointment

High Level Use Case Description

Use Case:	View Appointment
Actor:	User
Description:	Users can view their current appointments as well as their appointments history.

Table 7: High Level Use Case – User View Appointment

Use Case Diagram

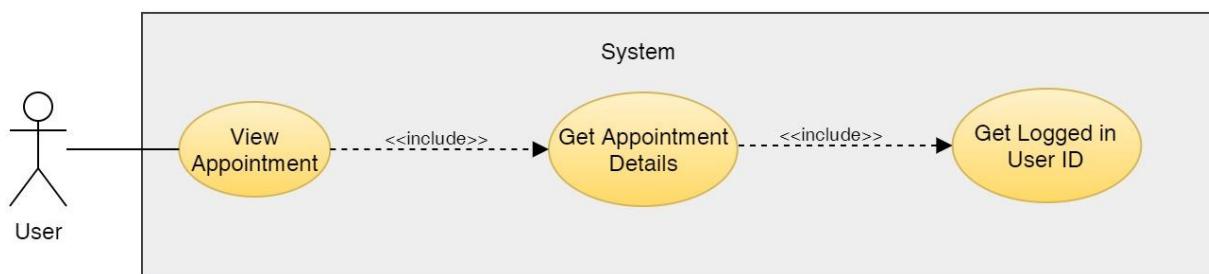


Figure 30: Use Case Diagram – User View Appointment

Collaboration Diagram

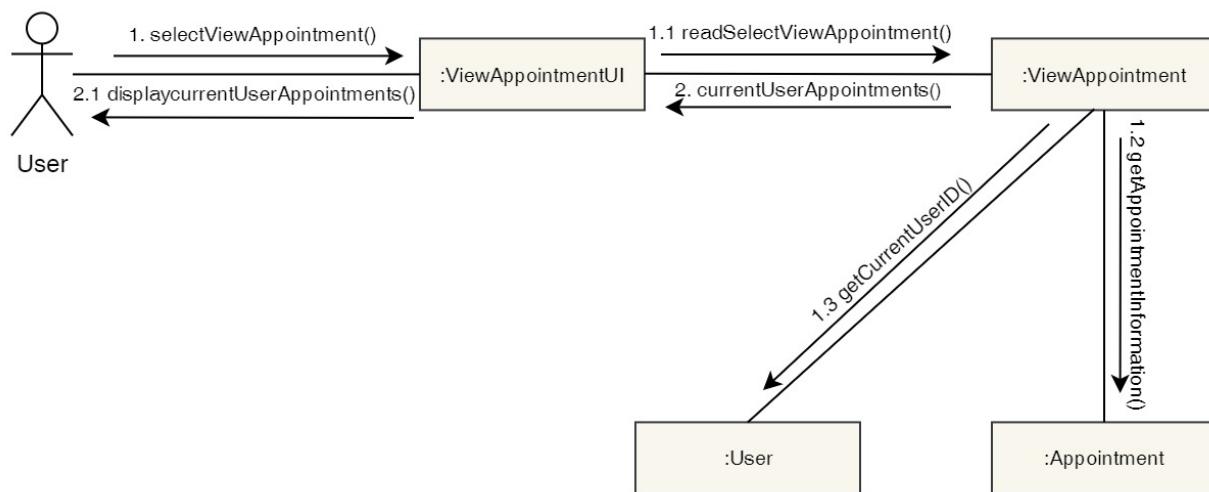


Figure 31: Collaboration Diagram – User View Appointment

Activity Diagram

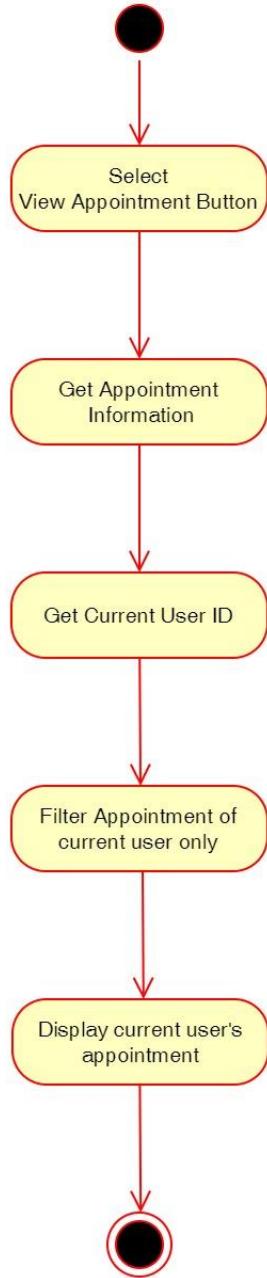


Figure 32: Activity Diagram – User View Appointment

Sequence Diagram

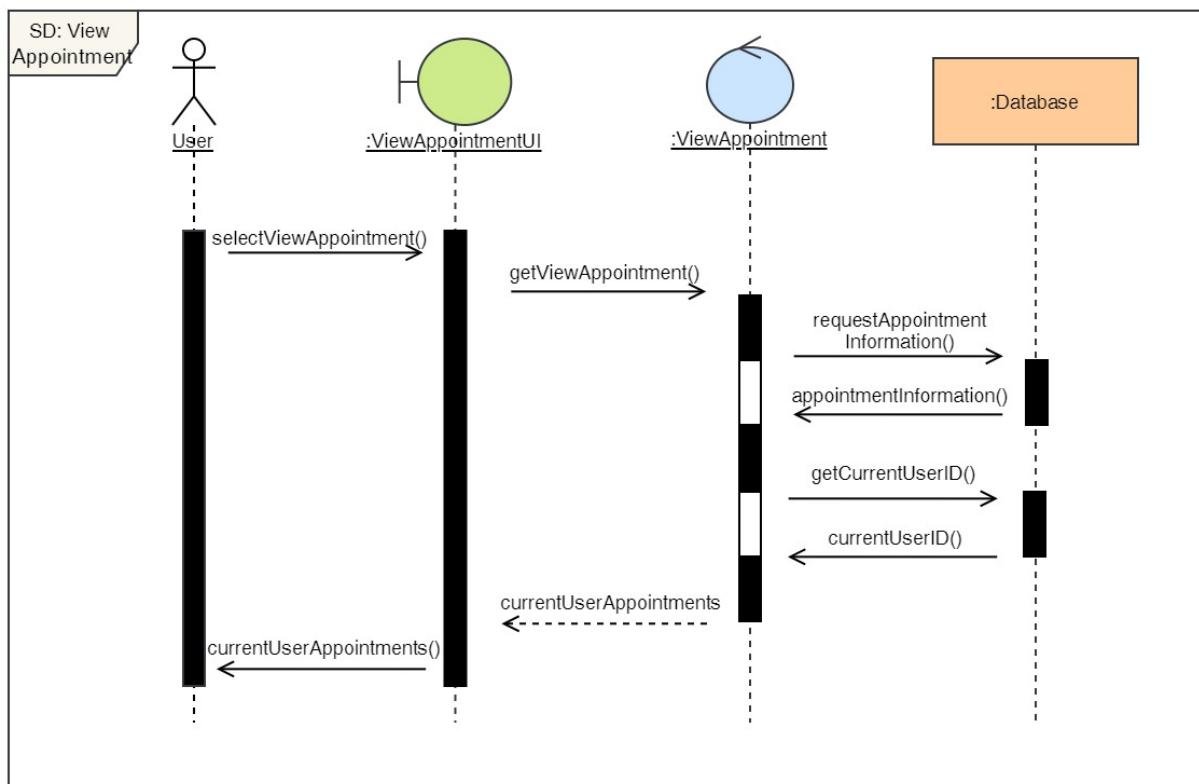


Figure 33: Sequence Diagram – User View Appointment

7. Cancel Appointment

High Level Use Case Description

Use Case:	Cancel Appointment
Actor:	User
Description:	The user can cancel their pending appointment or the appointment that has been confirmed.

Table 8: High Level Use Case – User Cancel Appointment

Use Case Diagram

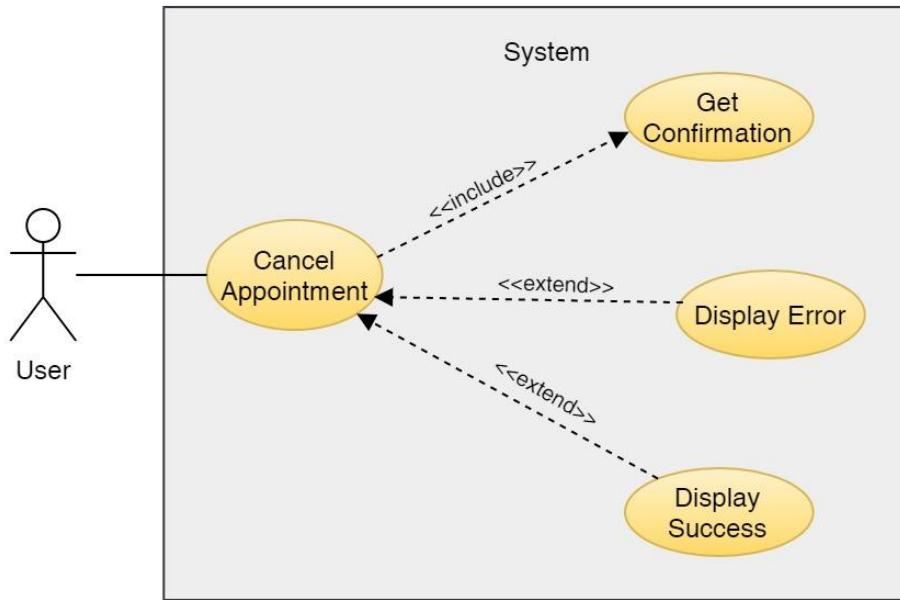


Figure 34: Use Case Diagram – User Cancel Appointment

Collaboration Diagram

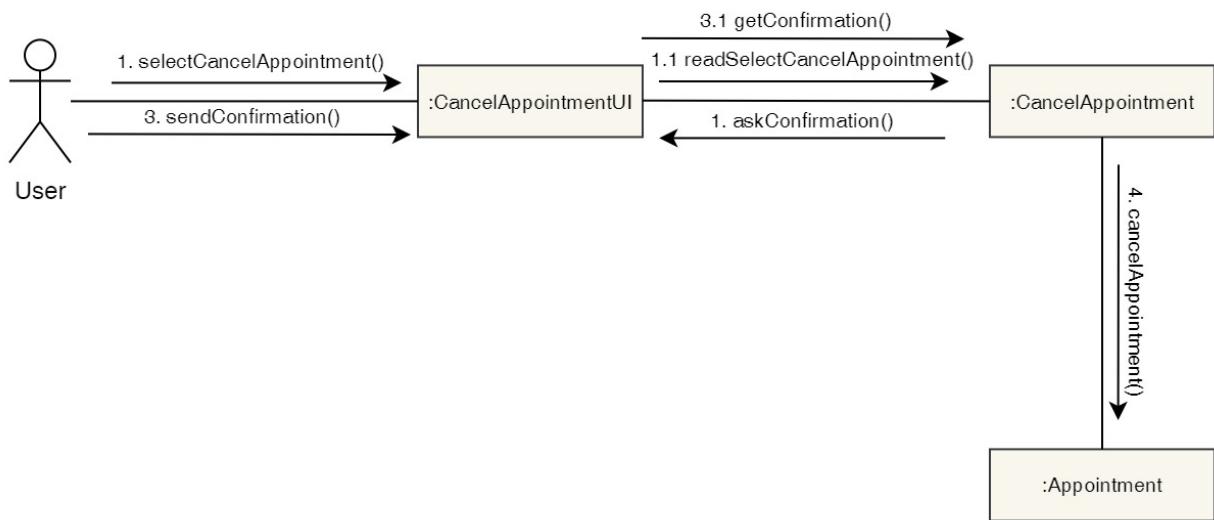


Figure 35: Collaboration Diagram – User Cancel Appointment

Activity Diagram

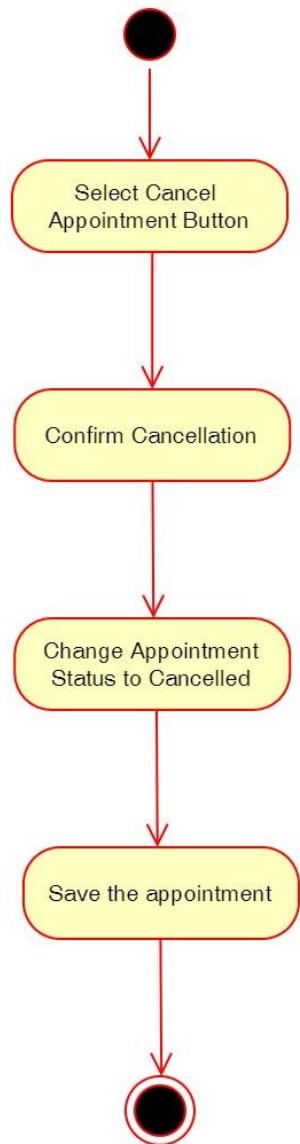


Figure 36: Activity Diagram – User Cancel Appointment

Sequence Diagram

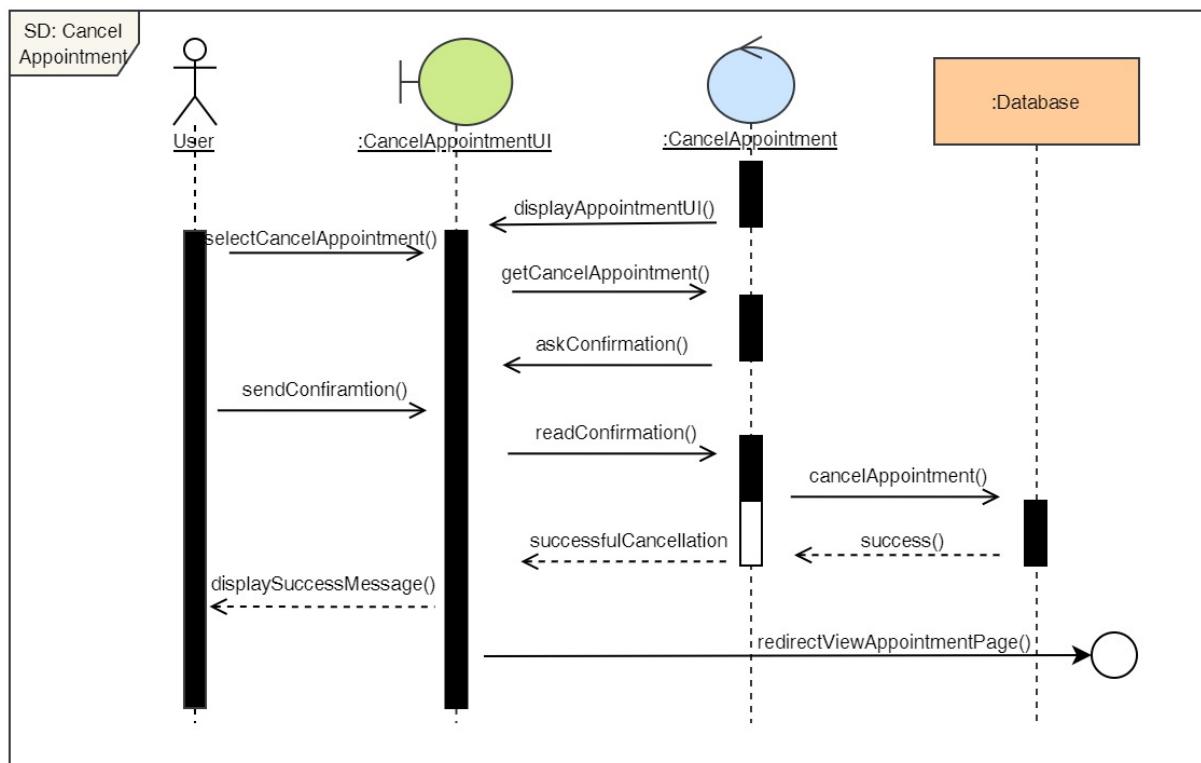


Figure 37: Sequence Diagram – User Cancel Appointment

8. View available doctors

High Level Use Case Description

Use Case:	View Doctors
Actor:	User
Description:	Users can view doctors that are available at the clinic and read their description and see their qualification as well.

Table 9: High Level Use Case – User View Doctors

Use Case Diagram

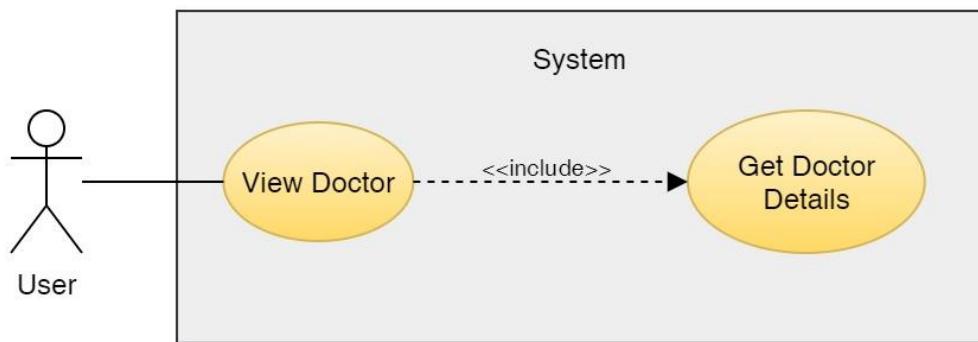


Figure 38: Use Case Diagram – User View Doctor

Collaboration Diagram

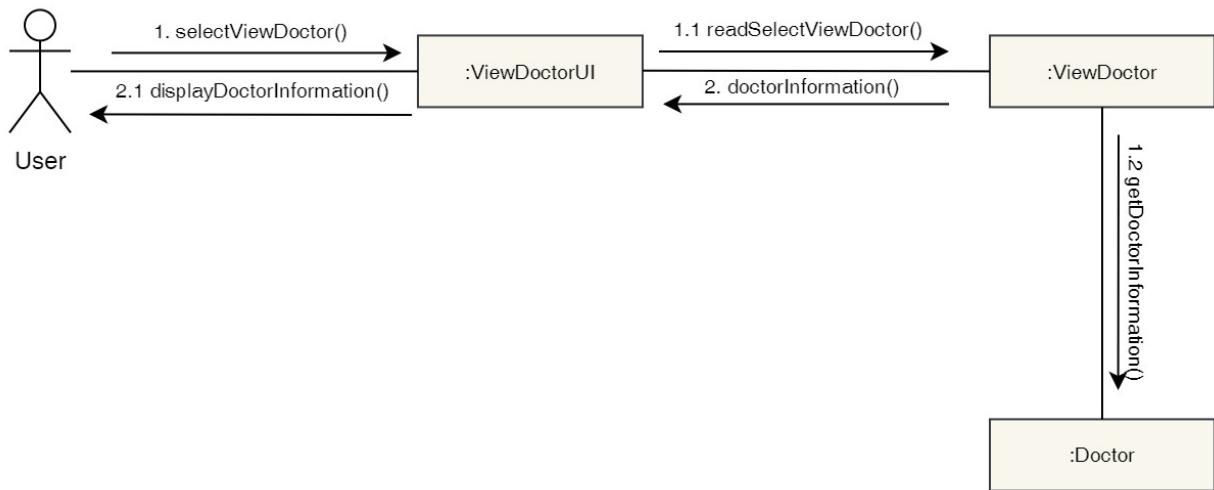


Figure 39: Collaboration Diagram – User View Doctor

Activity Diagram

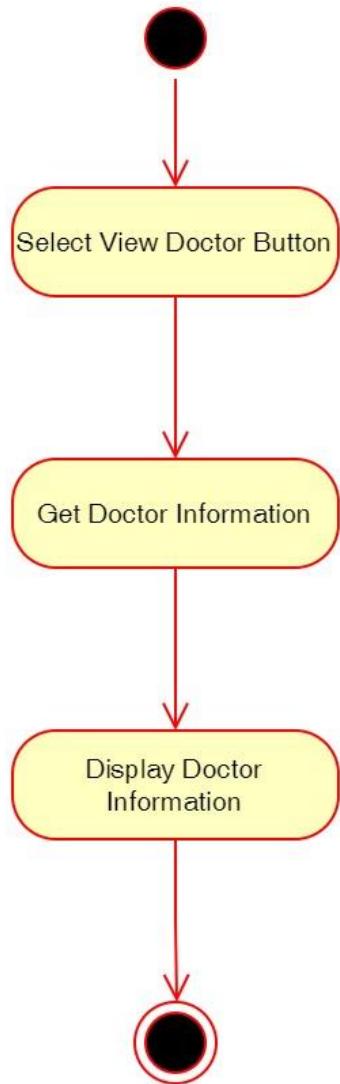


Figure 40: Activity Diagram – User View Doctor

Sequence Diagram

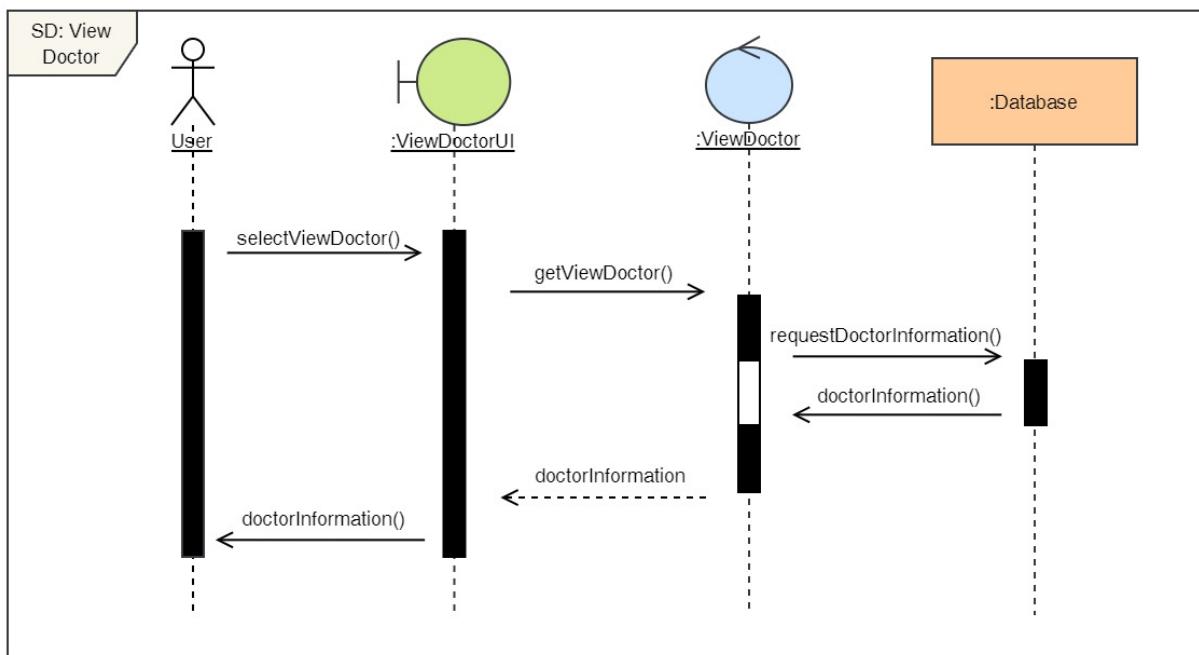


Figure 41: Sequence Diagram – User View Doctor

9. View report

High Level Use Case Description

Use Case:	View Report
Actor:	User
Description:	Users can view report about their visit to the clinic. They can download it as well.

Table 10: High Level Use Case – User View Report

Use Case Diagram

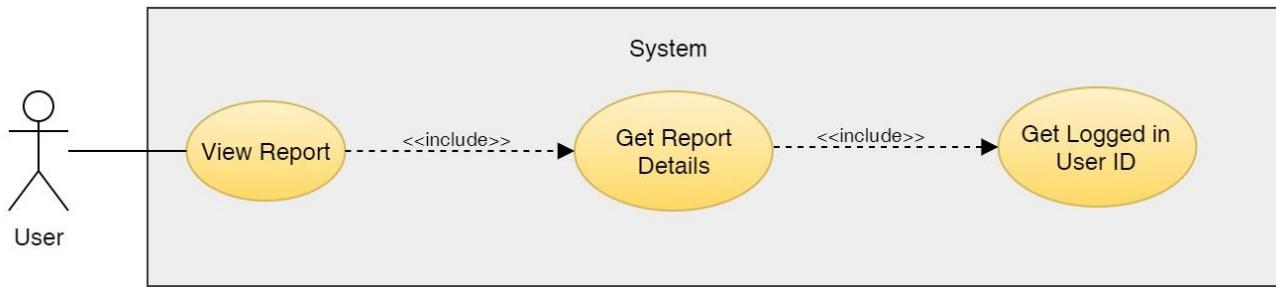


Figure 42: Use Case Diagram – User View Report

Collaboration Diagram

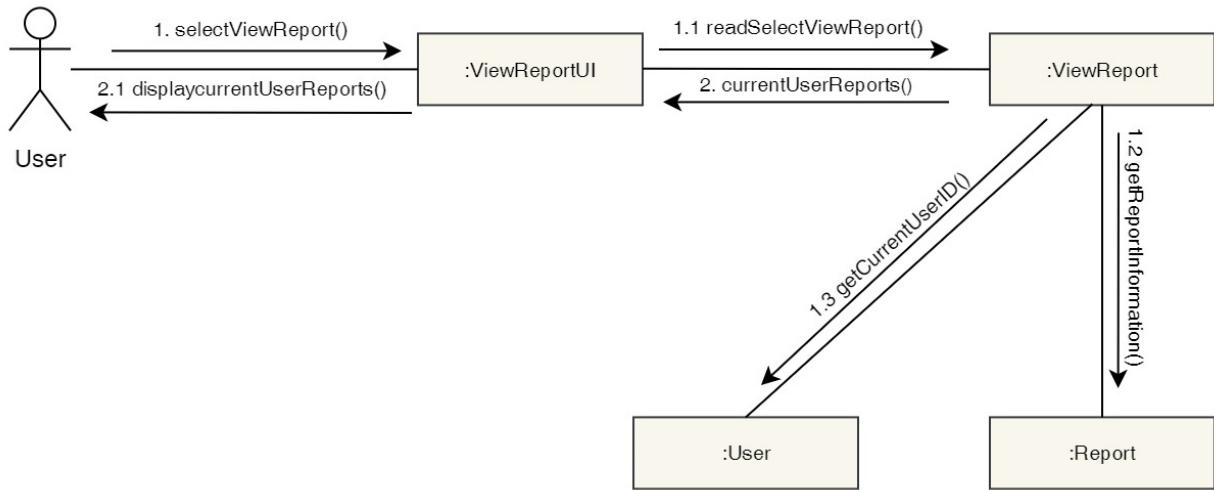


Figure 43: Collaboration Diagram – User View Report

Activity Diagram

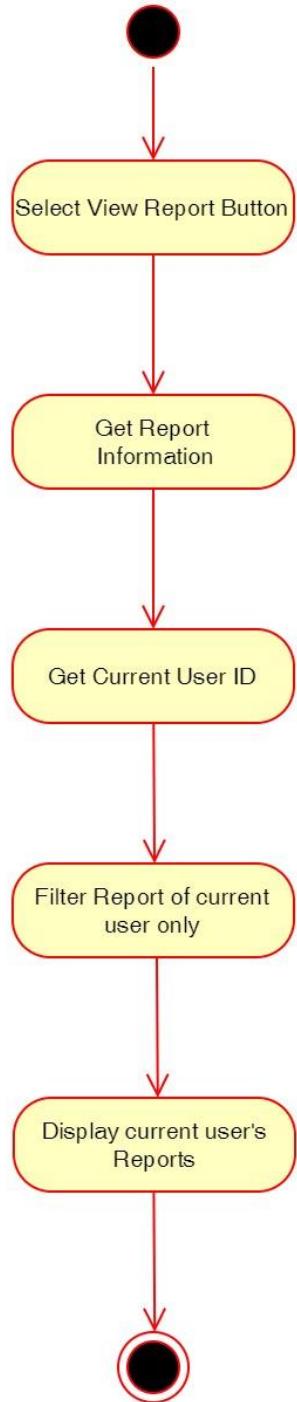


Figure 44: Activity Diagram – User View Report

Sequence Diagram

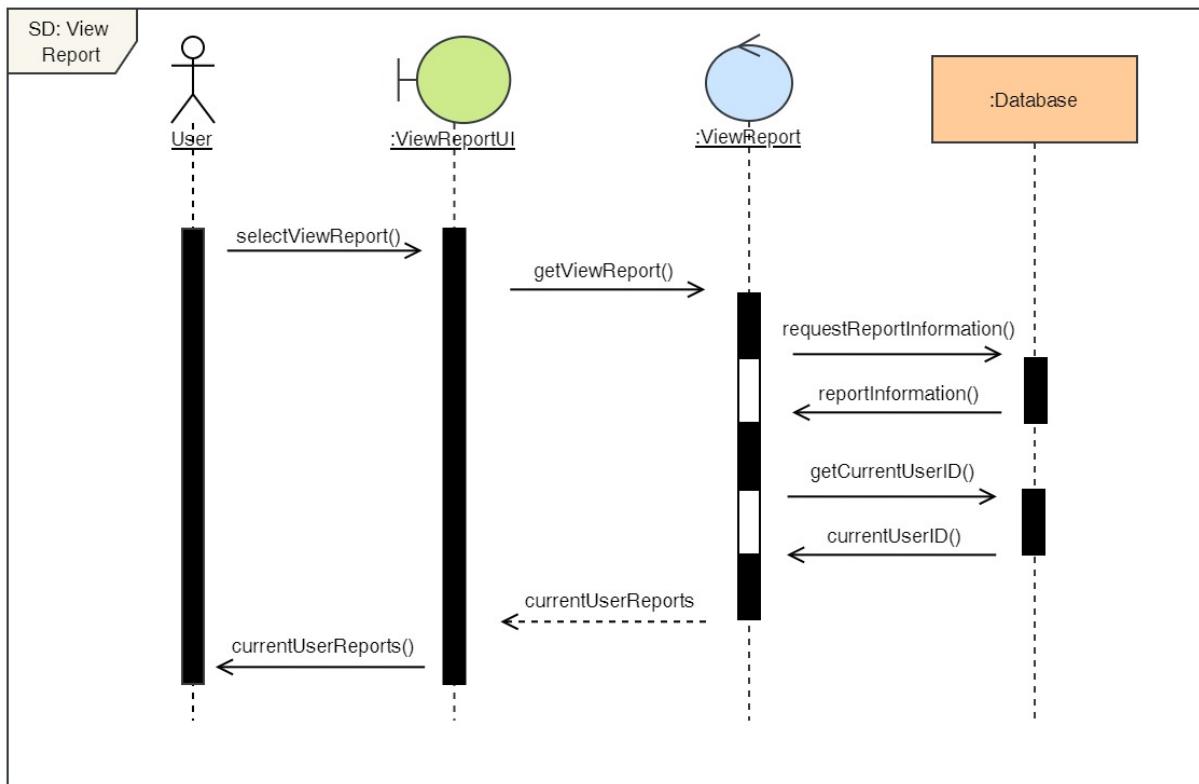


Figure 45: Sequence Diagram – User View Report

10. View services

High Level Use Case Description

Use Case:	View Services
Actor:	User
Description:	Users can view the list of services available along with the cost of it and description.

Table 11: High Level Use Case – User View Services

Use Case Diagram

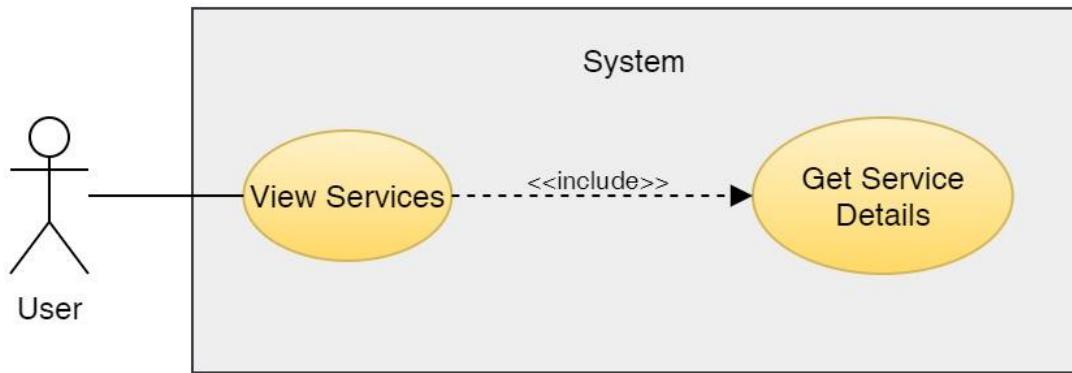


Figure 46: Use Case Diagram – User View Services

Collaboration Diagram

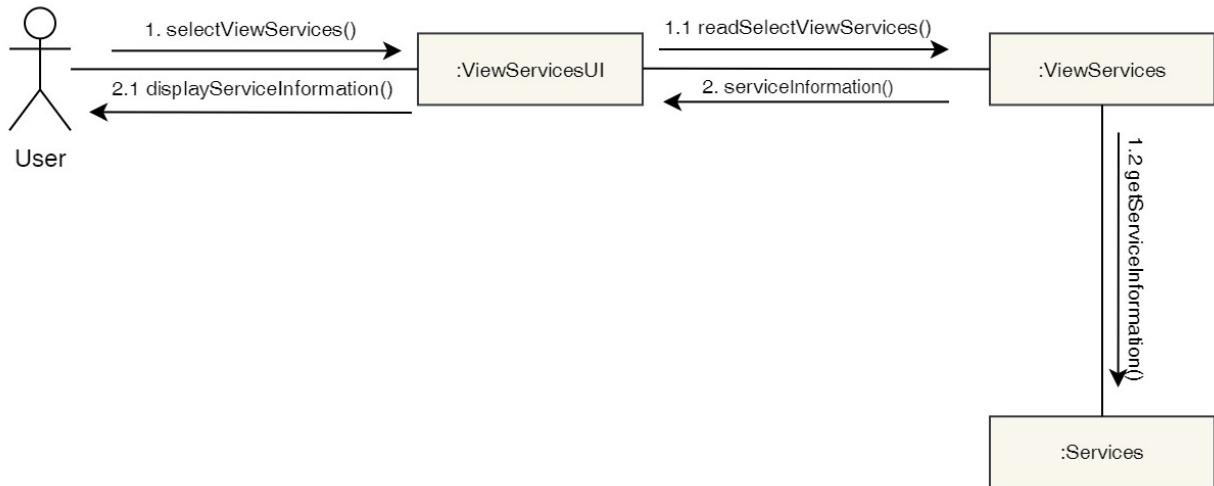


Figure 47: Collaboration Diagram – User View Services

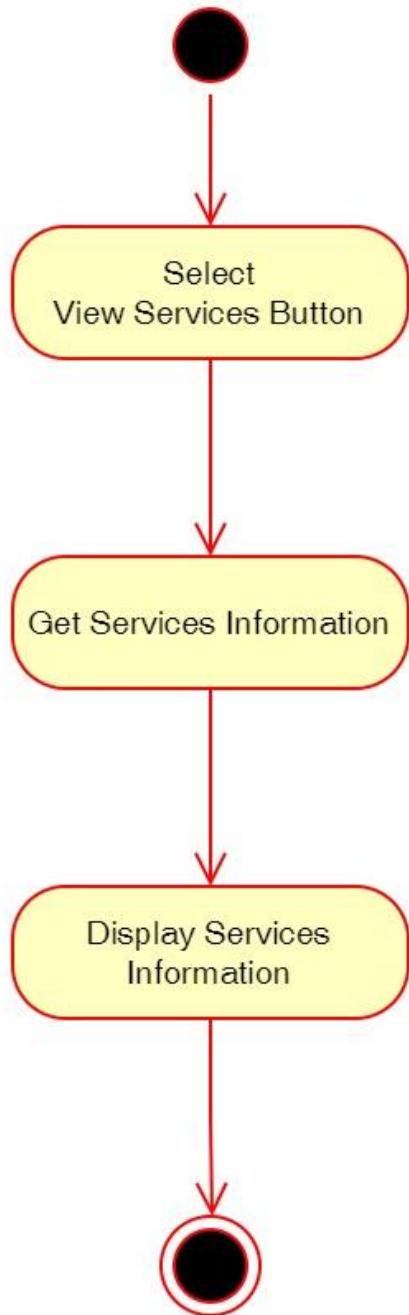
Activity Diagram

Figure 48: Activity Diagram – User View Services

Sequence Diagram

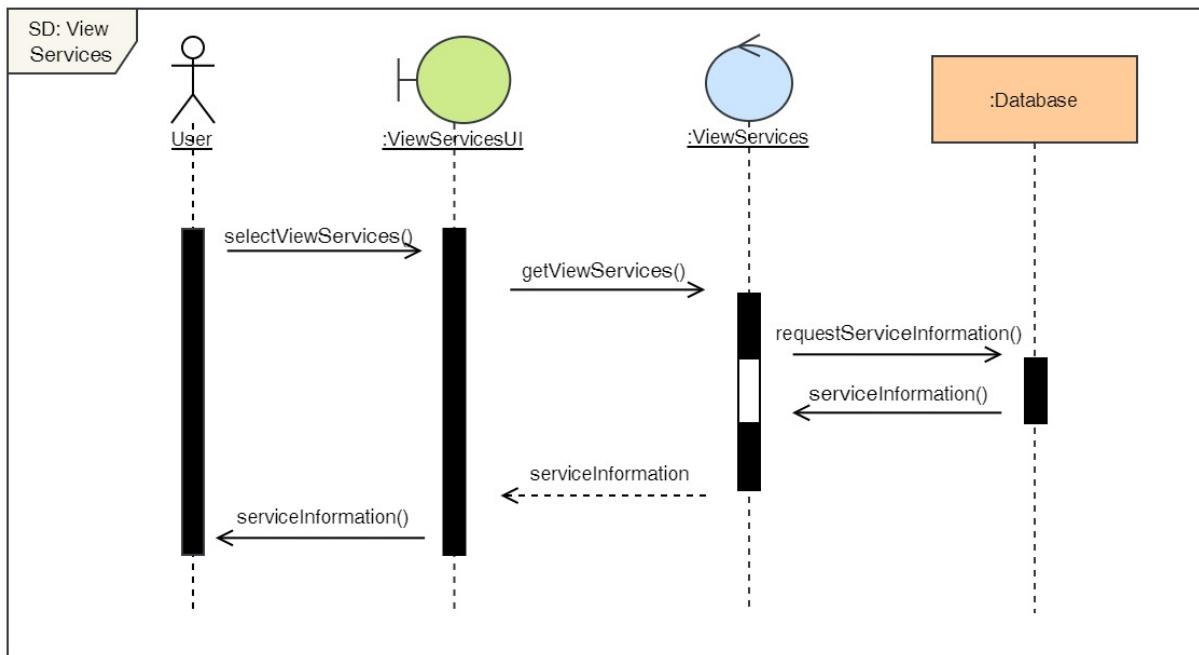


Figure 49: Sequence Diagram – User View Services

[Click here to view the rest of the functional UML diagrams of the admin side.](#)

3.7 Implementation

3.7.1 System Architecture

A system architecture is a conceptual model that describes a system's structure and behaviour. This project is done on Django Framework which is based on MVT (Model-View-Template) Architecture. MVT is a web application development software design pattern. It has three major components, which are listed as follows:

Model: The model is going to be your data's interface. It is in charge of data management. The database (SQLite3) represents the logical data structure that underpins the whole program.

View: It is the user interface that we see in our browser when we render a website. HTML/CSS/JavaScript/Bootstrap are used to represent it.

Template: It is made up of both static sections of the desired HTML output and specific syntax that describes how dynamic information will be included.

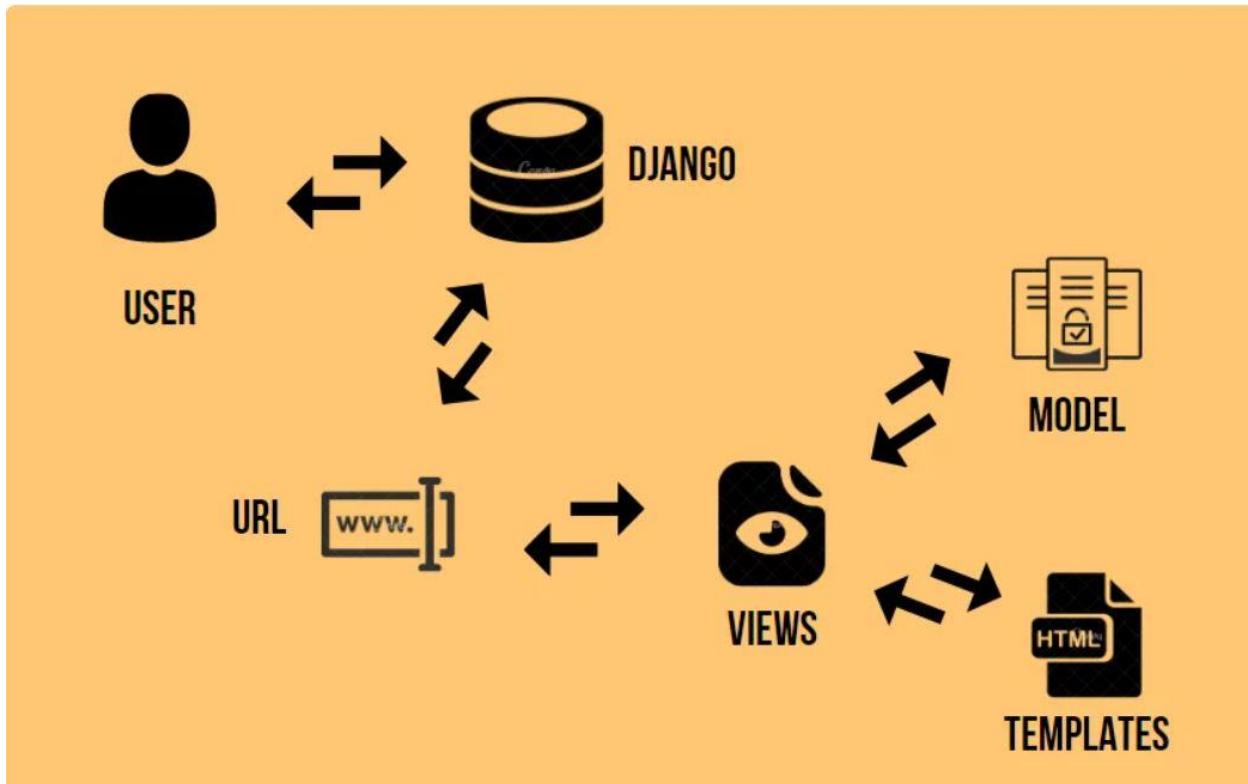
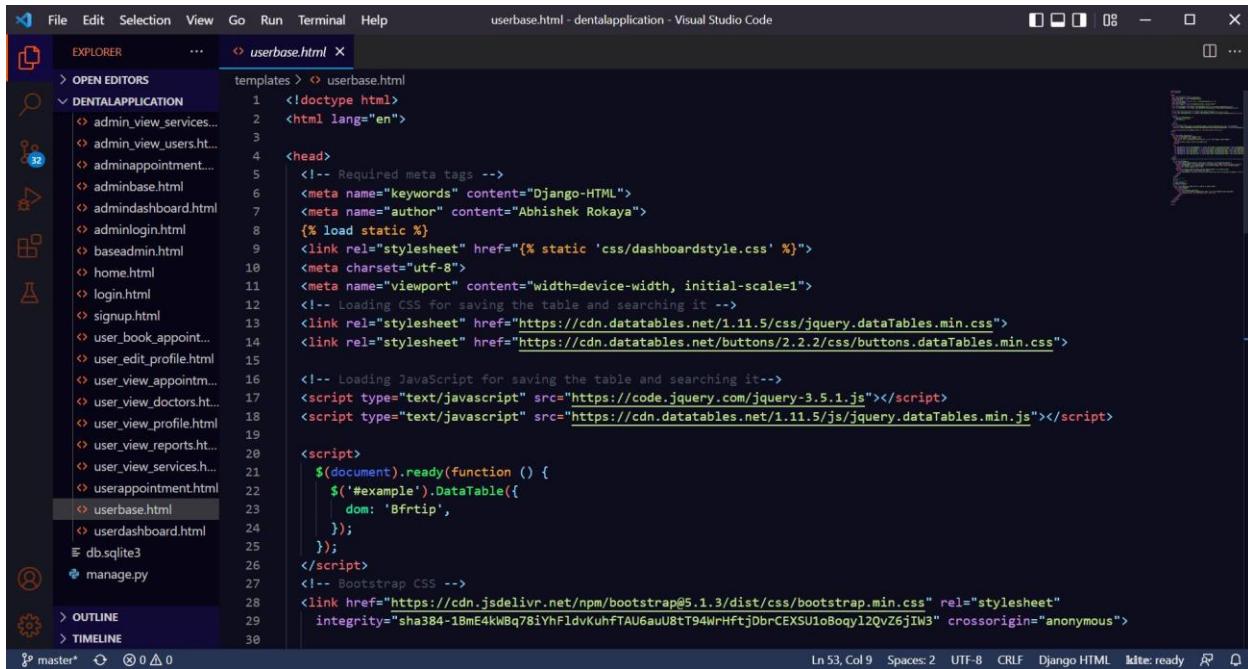


Figure 50: System Architecture (AskPython, 2022)

In the above image, we can see the control flow of the MVT Architecture in Django. The user first sends a URL request for a resource to Django. Then the Django framework searches up the URL resource. If the URL path points to a certain View, that View is called. After that, the View will communicate with the Model and obtain the necessary data from the database. The View then returns to the user with an appropriate template and the obtained data. In this way, the MVT Architecture works in Django Framework.

3.7.2 Frontend Implementation

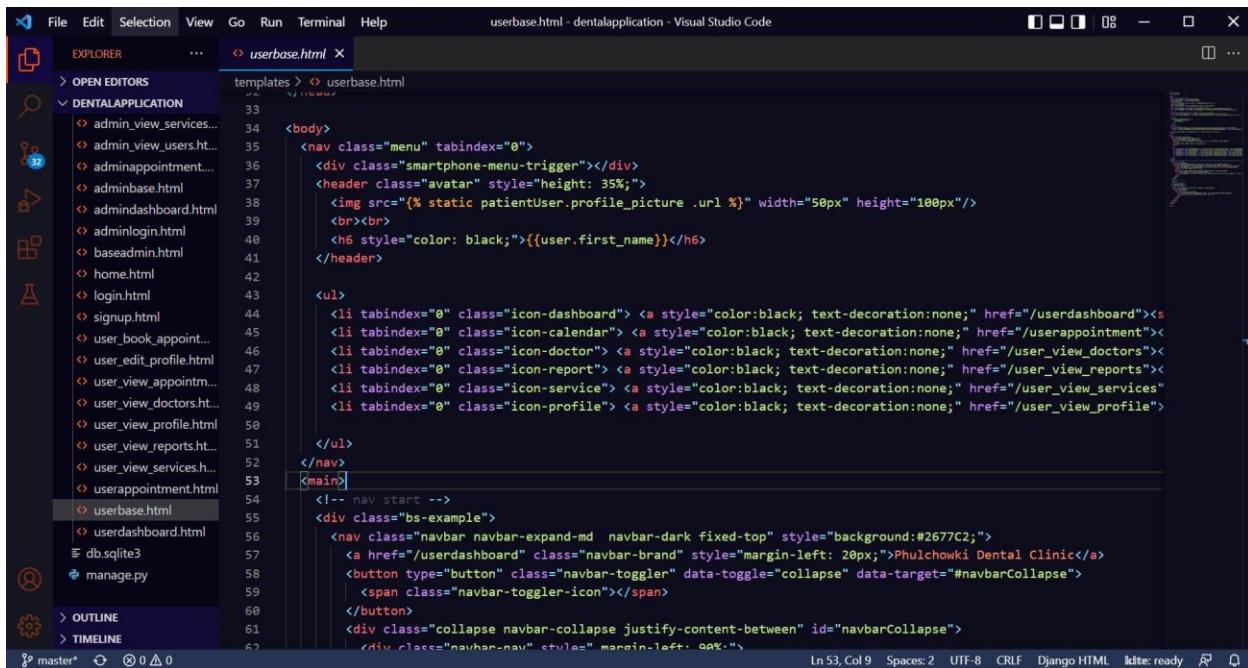
User base



The screenshot shows the Visual Studio Code interface with the 'userbase.html' file open in the editor. The file contains HTML code for a user base page, including meta tags, a head section with CSS and JavaScript imports, and a script block for initializing a DataTable. The code is color-coded for syntax highlighting.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <!-- Required meta tags -->
    <meta name="keywords" content="Django-HTML">
    <meta name="author" content="Abhishek Rokaya">
    {% load static %}
    <link rel="stylesheet" href="{% static 'css/dashboardstyle.css' %}">
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <!-- Loading CSS for saving the table and searching it -->
    <link rel="stylesheet" href="https://cdn.datatables.net/1.11.5/css/jquery.dataTables.min.css">
    <link rel="stylesheet" href="https://cdn.datatables.net/buttons/2.2.2/css/buttons.dataTables.min.css">
    <!-- Loading JavaScript for saving the table and searching it-->
    <script type="text/javascript" src="https://code.jquery.com/jquery-3.5.1.js"></script>
    <script type="text/javascript" src="https://cdn.datatables.net/1.11.5/js/jquery.dataTables.min.js"></script>
<script>
$(document).ready(function () {
    $('#example').DataTable({
        dom: 'Bfrtip',
    });
});
</script>
<!-- Bootstrap CSS -->
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-Eqzq+8u0DlrxTFZjDbrCEXSU1oBoqyl2QvZ6jIW3" crossorigin="anonymous">
```

Figure 51: Frontend Implementation Code – 1



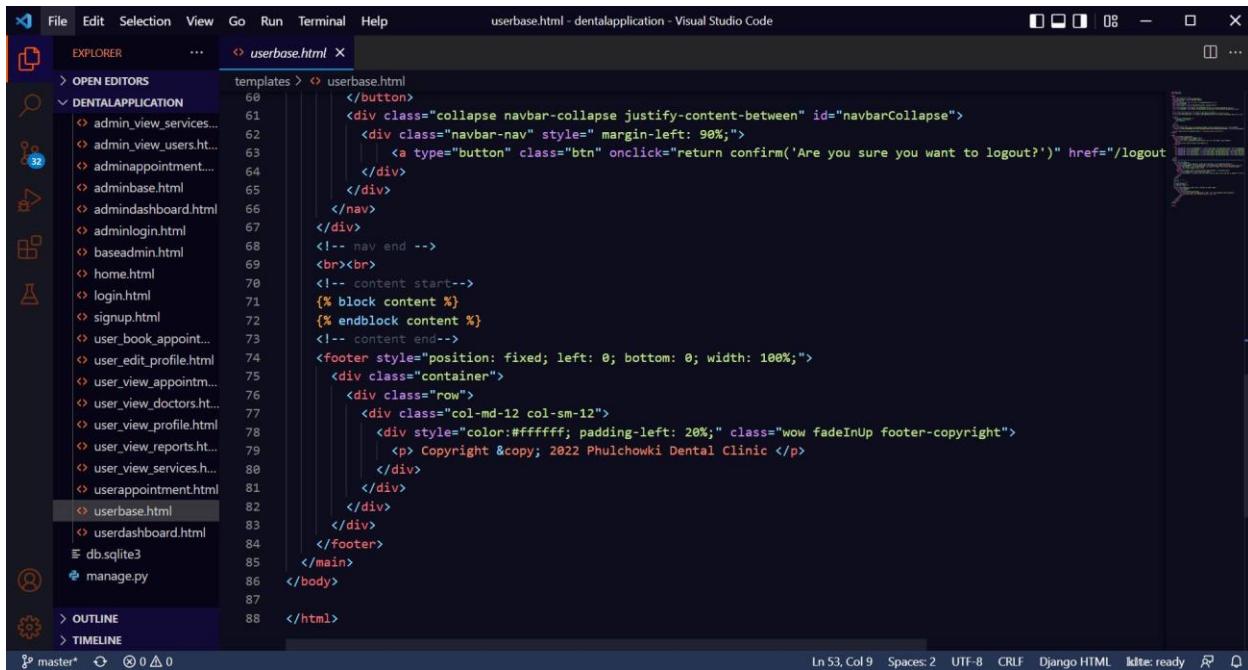
```

File Edit Selection View Go Run Terminal Help userbase.html - dentalapplication - Visual Studio Code
EXPLORER OPEN EDITORS DENTALAPPLICATION templates > userbase.html
admin_view_services.html
admin_view_users.html
adminappointment.html
adminbase.html
admindashboard.html
adminlogin.html
baseadmin.html
home.html
login.html
signup.html
user_book_appoint...
user_edit_profile.html
user_view_appointm...
user_view_doctors.h...
user_view_reports.h...
user_view_services.h...
userappointment.html
userbase.html
userdashboard.html
db.sqlite3
manage.py
OUTLINE TIMELINE
33
34 <body>
35   <nav class="menu" tabindex="0">
36     <div class="smartphone-menu-trigger"></div>
37     <header class="avatar" style="height: 35%;">
38       
39     <br><br>
40     <h6 style="color: black;">{{user.first_name}}</h6>
41   </header>
42
43   <ul>
44     <li tabindex="0" class="icon-dashboard"> <a style="color: black; text-decoration:none;" href="/userdashboard"><s>
45     <li tabindex="0" class="icon-calendar"> <a style="color: black; text-decoration:none;" href="/userappointment"><s>
46     <li tabindex="0" class="icon-doctor"> <a style="color: black; text-decoration:none;" href="/user_view_doctors"><s>
47     <li tabindex="0" class="icon-report"> <a style="color: black; text-decoration:none;" href="/user_view_reports"><s>
48     <li tabindex="0" class="icon-service"> <a style="color: black; text-decoration:none;" href="/user_view_services"><s>
49     <li tabindex="0" class="icon-profile"> <a style="color: black; text-decoration:none;" href="/user_view_profile"><s>
50
51   </ul>
52 </nav>
53 <!-- nav start -->
54 <div class="bs-example">
55   <nav class="navbar navbar-expand-md navbar-dark fixed-top" style="background:#2677C2;">
56     <a href="/userdashboard" class="navbar-brand" style="margin-left: 20px;">Phulchowki Dental Clinic</a>
57     <button type="button" class="navbar-toggler" data-toggle="collapse" data-target="#navbarCollapse">
58       <span class="navbar-toggler-icon"></span>
59     </button>
60   <div class="collapse navbar-collapse justify-content-between" id="navbarCollapse">
61     <div class="navbar-nav" style="margin-left: 20px;">
62
63   <!-- nav end -->
64   <div class="content" style="margin-top: 20px;">
65     <h1>Welcome to Phulchowki Dental Clinic</h1>
66     <p>A modern dental clinic management system</p>
67     <div class="row">
68       <div class="col-md-4">
69         <h2>Patient Services</h2>
70         <ul>
71           <li>Appointment Booking</li>
72           <li>Patient Profile Management</li>
73           <li>Treatment History</li>
74           <li>Prescription Management</li>
75         </ul>
76       </div>
77       <div class="col-md-4">
78         <h2>Administrative Tools</h2>
79         <ul>
80           <li>User Management</li>
81           <li>Report Generation</li>
82           <li>Setting Configuration</li>
83         </ul>
84       </div>
85     </div>
86   </div>
87 </div>
88 </html>

```

Ln 53, Col 9 Spaces: 2 UTF-8 CRLF Django HTML Idle: ready

Figure 52: Frontend Implementation Code – 2



```

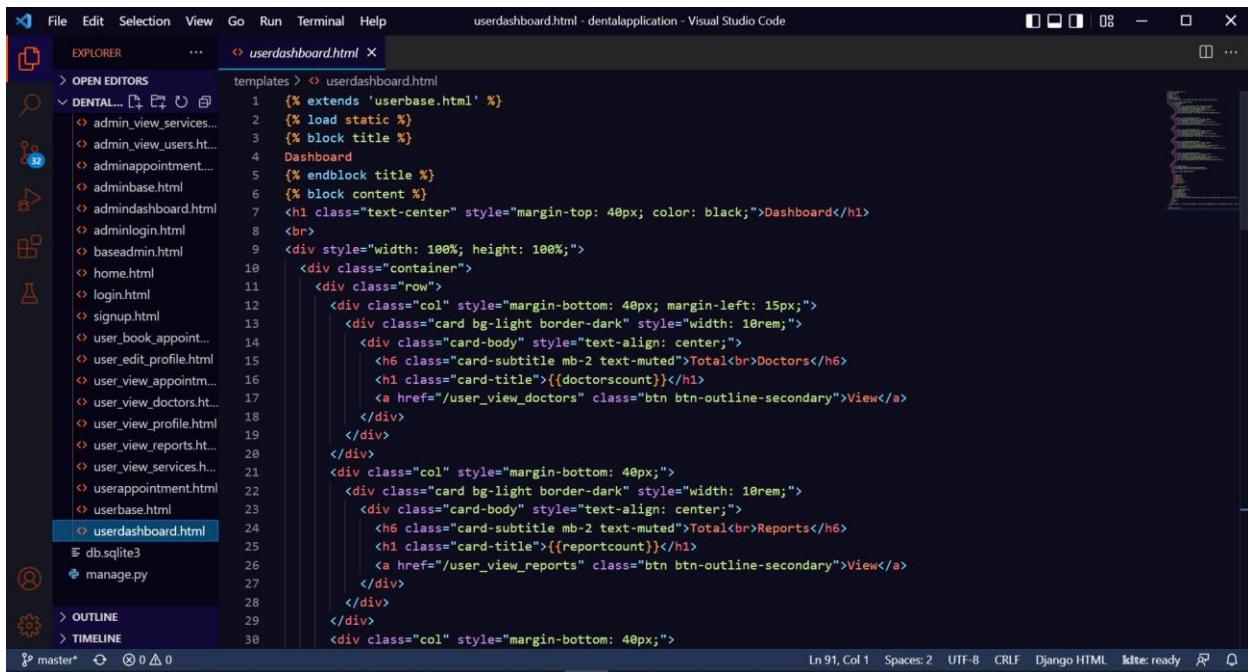
File Edit Selection View Go Run Terminal Help userbase.html - dentalapplication - Visual Studio Code
EXPLORER OPEN EDITORS DENTALAPPLICATION templates > userbase.html
admin_view_services.html
admin_view_users.html
adminappointment.html
adminbase.html
admindashboard.html
adminlogin.html
baseadmin.html
home.html
login.html
signup.html
user_book_appoint...
user_edit_profile.html
user_view_appointm...
user_view_doctors.h...
user_view_reports.h...
user_view_services.h...
userappointment.html
userbase.html
userdashboard.html
db.sqlite3
manage.py
OUTLINE TIMELINE
60
61   <button>
62     <div class="collapse navbar-collapse justify-content-between" id="navbarCollapse">
63       <div class="navbar-nav" style="margin-left: 90%; ">
64         <a type="button" class="btn" onclick="return confirm('Are you sure you want to logout?')" href="/logout">Logout</a>
65       </div>
66     </div>
67   </div>
68 <!-- nav end -->
69 <br><br>
70 <!-- content start-->
71 {%
72   block content %
73   %}
74 <!-- content end-->
75 <footer style="position: fixed; left: 0; bottom: 0; width: 100%;">
76   <div class="container">
77     <div class="row">
78       <div class="col-md-12 col-sm-12">
79         <div style="color:#ffffff; padding-left: 20%; class="wow fadeInUp footer-copyright">
80           <p>Copyright &copy; 2022 Phulchowki Dental Clinic </p>
81         </div>
82       </div>
83     </div>
84   </div>
85 </main>
86 </body>
87 </html>

```

Ln 53, Col 9 Spaces: 2 UTF-8 CRLF Django HTML Idle: ready

Figure 53: Frontend Implementation Code – 3

User Dashboard Code



The screenshot shows the Visual Studio Code interface with the file "userdashboard.html" open in the editor. The code is a Django template for a user dashboard, featuring Bootstrap cards and navigation links. The code includes logic for doctor and report counts.

```
% extends 'userbase.html' %}
{%- load static %}
{% block title %}
Dashboard
{% endblock title %}
{% block content %}
<h1 class="text-center" style="margin-top: 40px; color: black;">Dashboard</h1>
<div style="width: 100%; height: 100%;">
<div class="container">
<div class="row">
<div class="col" style="margin-bottom: 40px; margin-left: 15px;">
<div class="card bg-light border-dark" style="width: 10rem;">
<div class="card-body" style="text-align: center;">
<h6 class="card-subtitle mb-2 text-muted">Total<br>Doctors</h6>
<h1 class="card-title">{{doctorscount}}</h1>
<a href="/user_view_doctors" class="btn btn-outline-secondary">View</a>
</div>
</div>
</div>
<div class="col" style="margin-bottom: 40px;">
<div class="card bg-light border-dark" style="width: 10rem;">
<div class="card-body" style="text-align: center;">
<h6 class="card-subtitle mb-2 text-muted">Total<br>Reports</h6>
<h1 class="card-title">{{reportcount}}</h1>
<a href="/user_view_reports" class="btn btn-outline-secondary">View</a>
</div>
</div>
</div>
<div class="col" style="margin-bottom: 40px;">

```

Figure 54: Frontend Implementation Code – 4

```

File Edit Selection View Go Run Terminal Help
userdashboard.html - dentalapplication - Visual Studio Code
EXPLORER ... templates > userdashboard.html
OPEN EDITORS > DENTAL... admin_view_services...
admin_view_users.ht... adminappointment...
adminbase.html admindashboard.html
adminlogin.html adminlogin.html
baseadmin.html baseadmin.html
home.html login.html
signup.html user_book_appoint...
user_edit_profile.html user_view_appointment...
user_view_doctors.ht... user_view_profile.html
user_view_reports.ht... user_view_services.h...
userappointment.html userbase.html
userdashboard.html db.sqlite3 manage.py
> OUTLINE > TIMELINE
Ln 91, Col 1 Spaces: 2 UTF-8 CRLF Django HTML kite: ready

```

```

</div>
<div class="col" style="margin-bottom: 40px;">
  <div class="card bg-light border-dark" style="width: 10rem;">
    <div class="card-body" style="text-align: center;">
      <h6 class="card-subtitle mb-2 text-muted">Total Services<br></h6>
      <h1 class="card-title">{{servicecount}}</h1>
      <a href="/user_view_services" class="btn btn-outline-secondary">View</a>
    </div>
  </div>
</div>
<div class="col" style="margin-bottom: 40px;">
  <div class="card bg-light border-dark" style="width: 10rem;">
    <div class="card-body" style="text-align: center;">
      <h6 class="card-subtitle mb-2 text-muted">Finished Appointments</h6>
      <h1 class="card-title">{{finishedappointmentcount}}</h1>
      <a href="/user_book_appointment" class="btn btn-outline-secondary">Book New</a>
    </div>
  </div>
</div>
<div class="col" style="margin-bottom: 40px;">
  <div class="card bg-light border-dark" style="width: 10rem;">
    <div class="card-body" style="text-align: center;">
      <h6 class="card-subtitle mb-2 text-muted">Pending Appointments</h6>
      <h1 class="card-title">{{pendingappointmentcount}}</h1>
      <a href="/user_view_appointment" class="btn btn-outline-secondary">View</a>
    </div>
  </div>
</div>
<div style="margin-left: 1%; ">
  <div class="panel-heading" style="text-align:center;">
    <h4 class="panel-title">Upcoming Appointments</h4><br>
  </div>
  <table class="table table-bordered">
    <thead >
      <tr>
        <th>Name</th>
        <th>Date</th>
        <th>Time</th>
        <th>Service</th>
        <th>Doctor</th>
        <th>Description</th>
        <th>Status</th>
      </tr>
    </thead>
    <% for a in appointment %>
    <tr>
      <td>{{a.patientUser}}</td>
      <td>{{a.date}}</td>
      <td>{{a.time}}</td>
      <td>{{a.service}}</td>
      <td>{{a.doctor}}</td>
      <td>{{a.description}}</td>
      <td>{{a.status}}</td>
    </tr>
    <% endfor %>
  </table>
</div>

```

Figure 55: Frontend Implementation Code – 5

```

File Edit Selection View Go Run Terminal Help
userdashboard.html - dentalapplication - Visual Studio Code
EXPLORER ... templates > userdashboard.html
OPEN EDITORS > DENTAL... admin_view_services...
admin_view_users.ht... adminappointment...
adminbase.html admindashboard.html
adminlogin.html adminlogin.html
baseadmin.html baseadmin.html
home.html login.html
signup.html user_book_appoint...
user_edit_profile.html user_view_appointment...
user_view_doctors.ht... user_view_profile.html
user_view_reports.ht... user_view_services.h...
userappointment.html userbase.html
userdashboard.html db.sqlite3 manage.py
> OUTLINE > TIMELINE
Ln 91, Col 1 Spaces: 2 UTF-8 CRLF Django HTML kite: ready

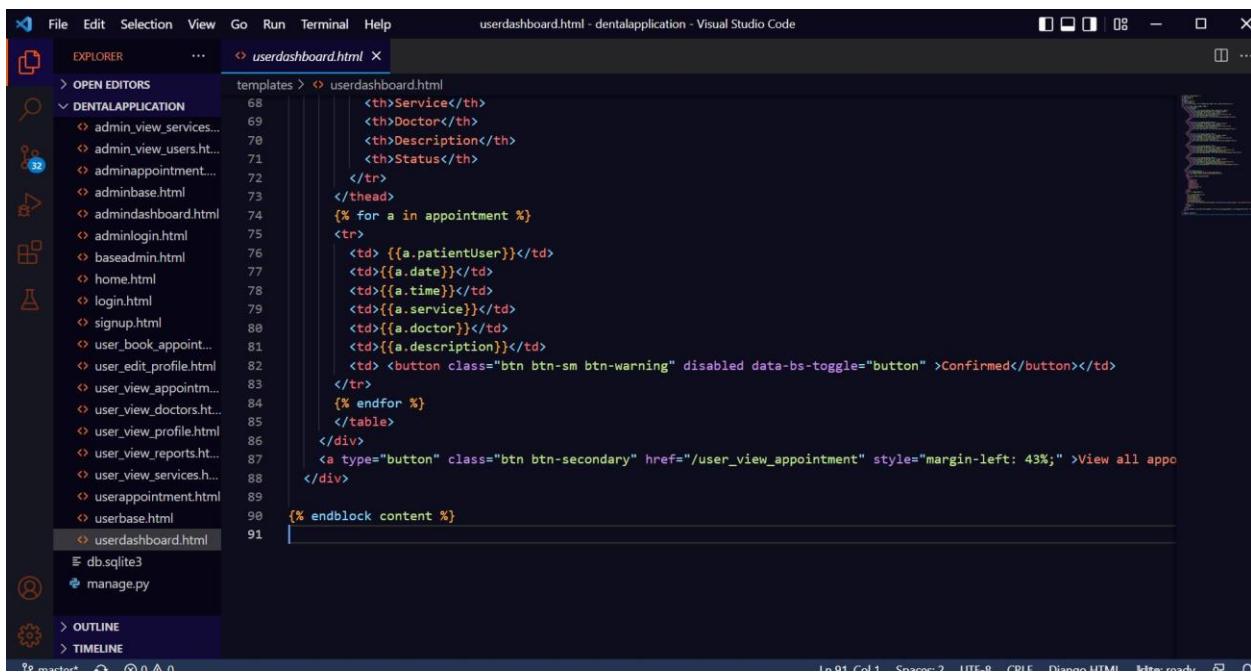
```

```

<h1 class="card-title">{{pendingappointmentcount}}</h1>
<a href="/user_view_appointment" class="btn btn-outline-secondary">View</a>
</div>
</div>
</div>
<div style="margin-left: 1%; ">
  <div class="panel-heading" style="text-align:center;">
    <h4 class="panel-title">Upcoming Appointments</h4><br>
  </div>
  <table class="table table-bordered">
    <thead >
      <tr>
        <th>Name</th>
        <th>Date</th>
        <th>Time</th>
        <th>Service</th>
        <th>Doctor</th>
        <th>Description</th>
        <th>Status</th>
      </tr>
    </thead>
    <% for a in appointment %>
    <tr>
      <td>{{a.patientUser}}</td>
      <td>{{a.date}}</td>
      <td>{{a.time}}</td>
      <td>{{a.service}}</td>
      <td>{{a.doctor}}</td>
      <td>{{a.description}}</td>
      <td>{{a.status}}</td>
    </tr>
    <% endfor %>
  </table>
</div>

```

Figure 56: Frontend Implementation Code – 6



The screenshot shows the Visual Studio Code interface with the file "userdashboard.html" open in the editor. The code is a template for a user dashboard, displaying a table of appointments with columns for Service, Doctor, Description, Status, Date, Time, and Service. A button for confirming an appointment is included. The code uses Django's templating syntax ({{ }} and {% %}). The Explorer sidebar on the left shows other files in the "DENTALAPPLICATION" folder, including admin-related files like "admin_view_services.html" and "adminappointment.html". The bottom status bar indicates the file is 16.91 KB, has 2 changes, and is in UTF-8 encoding.

```
<thead>
  <tr>
    <th>Service</th>
    <th>Doctor</th>
    <th>Description</th>
    <th>Status</th>
  </tr>
</thead>
{% for a in appointment %}
<tr>
  <td> {{a.patientUser}}</td>
  <td> {{a.date}}</td>
  <td> {{a.time}}</td>
  <td> {{a.service}}</td>
  <td> {{a.doctor}}</td>
  <td> {{a.description}}</td>
  <td> <button class="btn btn-sm btn-warning" disabled data-bs-toggle="button" >Confirmed</button></td>
</tr>
{% endfor %}
</table>
</div>
<a type="button" class="btn btn-secondary" href="/user_view_appointment" style="margin-left: 43%;" >View all appo
</div>
{% endblock content %}
```

Figure 57: Frontend Implementation Code – 7

User Dashboard View

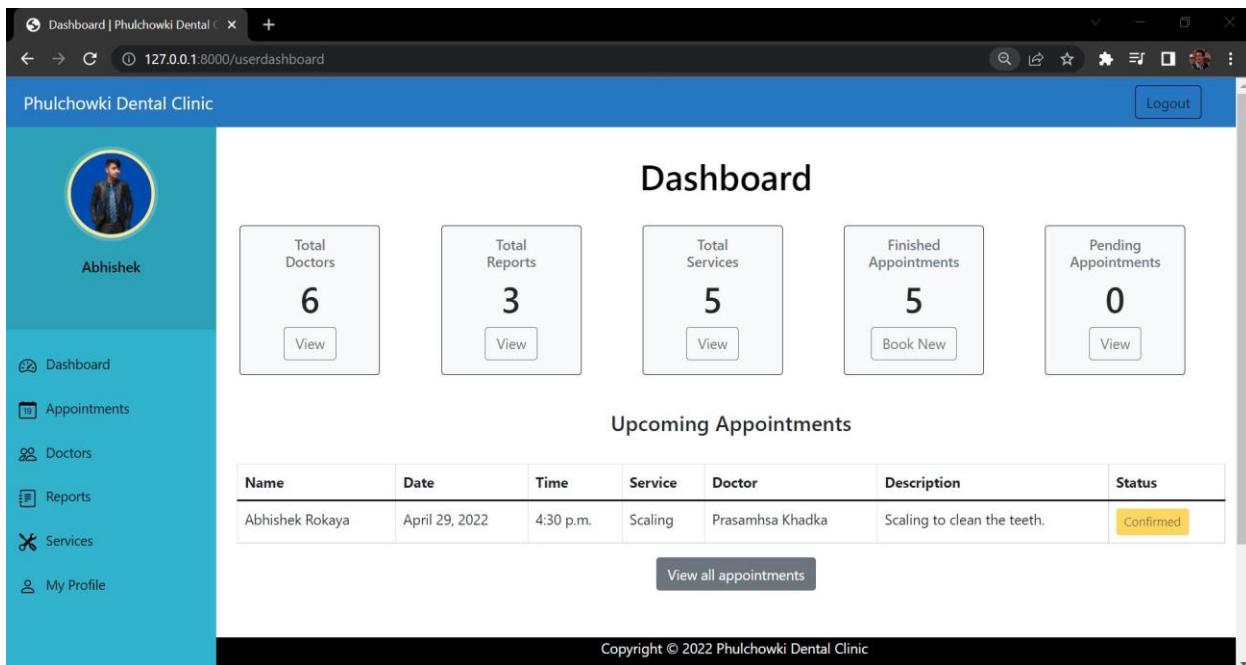
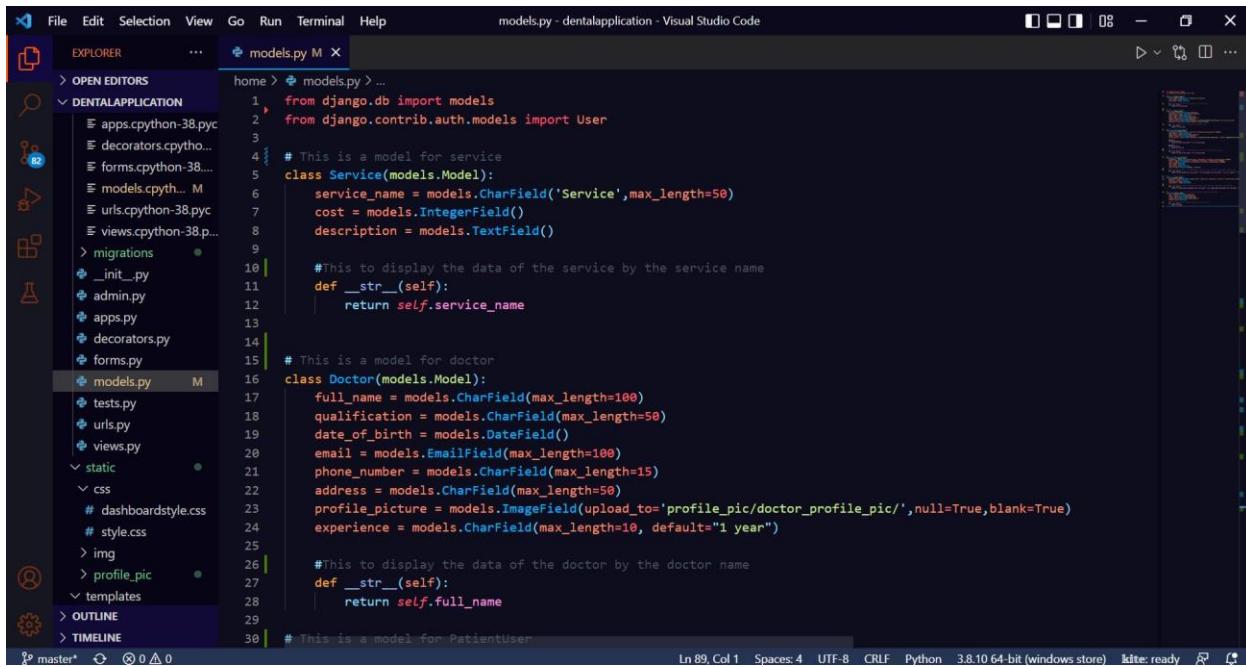


Figure 58: Frontend View

[Click here to view all the codes for the UI.](#)

3.7.3 Database Implementation



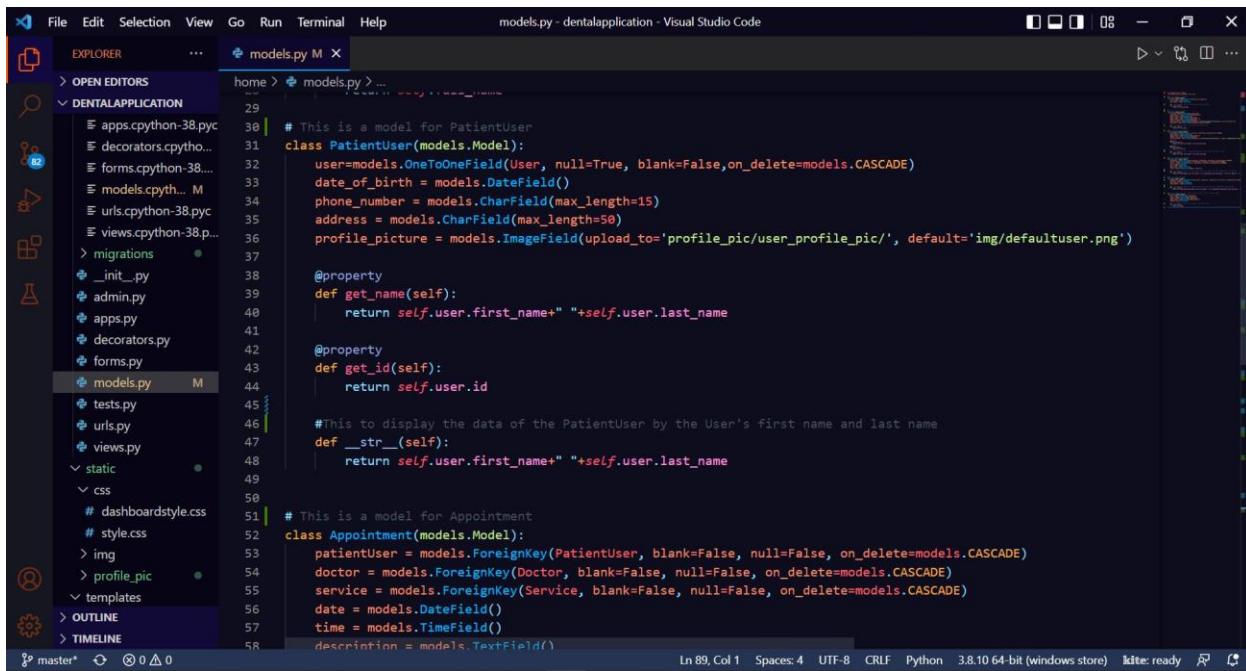
The screenshot shows the Visual Studio Code interface with the file `models.py` open in the editor. The code defines two Django models: `Service` and `Doctor`. The `Service` model has fields for service name (CharField), cost (IntegerField), and description (TextField). The `Doctor` model has fields for full name (CharField), qualification (CharField), date of birth (DateField), email (EmailField), phone number (CharField), address (CharField), profile picture (ImageField), and experience (CharField). Both models have a `__str__` method returning their respective names.

```

1  from django.db import models
2  from django.contrib.auth.models import User
3
4  # This is a model for service
5  class Service(models.Model):
6      service_name = models.CharField('Service', max_length=50)
7      cost = models.IntegerField()
8      description = models.TextField()
9
10     #This to display the data of the service by the service name
11     def __str__(self):
12         return self.service_name
13
14     # This is a model for doctor
15     class Doctor(models.Model):
16         full_name = models.CharField(max_length=100)
17         qualification = models.CharField(max_length=50)
18         date_of_birth = models.DateField()
19         email = models.EmailField(max_length=100)
20         phone_number = models.CharField(max_length=15)
21         address = models.CharField(max_length=50)
22         profile_picture = models.ImageField(upload_to='profile_pic/doctor_profile_pic/', null=True, blank=True)
23         experience = models.CharField(max_length=10, default="1 year")
24
25         #This to display the data of the doctor by the doctor name
26         def __str__(self):
27             return self.full_name
28
29     # This is a model for PatientUser
30
31     # This is a model for PatientUser
32     class PatientUser(models.Model):
33         user=models.OneToOneField(User, null=True, blank=False, on_delete=models.CASCADE)
34         date_of_birth = models.DateField()
35         phone_number = models.CharField(max_length=15)
36         address = models.CharField(max_length=50)
37         profile_picture = models.ImageField(upload_to='profile_pic/user_profile_pic/', default='img/defaultuser.png')
38
39         @property
40         def get_name(self):
41             return self.user.first_name+ " "+self.user.last_name
42
43         @property
44         def get_id(self):
45             return self.user.id
46
47         #This to display the data of the PatientUser by the User's first name and last name
48         def __str__(self):
49             return self.user.first_name+ " "+self.user.last_name
50
51     # This is a model for Appointment
52     class Appointment(models.Model):
53         patientUser = models.ForeignKey(PatientUser, blank=False, null=False, on_delete=models.CASCADE)
54         doctor = models.ForeignKey(Doctor, blank=False, null=False, on_delete=models.CASCADE)
55         service = models.ForeignKey(Service, blank=False, null=False, on_delete=models.CASCADE)
56         date = models.DateField()
57         time = models.TimeField()
58         description = models.TextField()

```

Figure 59: Database Implementation Code – 1



The screenshot shows the Visual Studio Code interface with the file `models.py` open in the editor. The code defines two Django models: `PatientUser` and `Appointment`. The `PatientUser` model has a one-to-one relationship with the `User` model, with fields for date of birth, phone number, and address. It also has properties for `get_name` and `get_id`. The `Appointment` model has foreign key relationships with `PatientUser`, `Doctor`, and `Service`, along with fields for date, time, and description.

```

29
30     # This is a model for PatientUser
31     class PatientUser(models.Model):
32         user=models.OneToOneField(User, null=True, blank=False, on_delete=models.CASCADE)
33         date_of_birth = models.DateField()
34         phone_number = models.CharField(max_length=15)
35         address = models.CharField(max_length=50)
36         profile_picture = models.ImageField(upload_to='profile_pic/user_profile_pic/', default='img/defaultuser.png')
37
38         @property
39         def get_name(self):
40             return self.user.first_name+ " "+self.user.last_name
41
42         @property
43         def get_id(self):
44             return self.user.id
45
46         #This to display the data of the PatientUser by the User's first name and last name
47         def __str__(self):
48             return self.user.first_name+ " "+self.user.last_name
49
50
51     # This is a model for Appointment
52     class Appointment(models.Model):
53         patientUser = models.ForeignKey(PatientUser, blank=False, null=False, on_delete=models.CASCADE)
54         doctor = models.ForeignKey(Doctor, blank=False, null=False, on_delete=models.CASCADE)
55         service = models.ForeignKey(Service, blank=False, null=False, on_delete=models.CASCADE)
56         date = models.DateField()
57         time = models.TimeField()
58         description = models.TextField()

```

Figure 60: Database Implementation Code - 2

```

File Edit Selection View Go Run Terminal Help
models.py - dentalapplication - Visual Studio Code
OPEN EDITORS
DENTALAPPLICATION
apps.cpython-38.pyc
decorators.cpython-38...
forms.cpython-38...
models.cpython-38...
urls.cpython-38.pyc
views.cpython-38.p...
migrations
__init__.py
admin.py
apps.py
decorators.py
forms.py
models.py M
tests.py
urls.py
views.py
static
css
# dashboardstyle.css
# style.css
img
profile_pic
templates
home > models.py ...
51 | # This is a model for Appointment
52 | class Appointment(models.Model):
53 |     patientUser = models.ForeignKey(PatientUser, blank=False, null=False, on_delete=models.CASCADE)
54 |     doctor = models.ForeignKey(Doctor, blank=False, null=False, on_delete=models.CASCADE)
55 |     service = models.ForeignKey(Service, blank=False, null=False, on_delete=models.CASCADE)
56 |     date = models.DateField()
57 |     time = models.TimeField()
58 |     description = models.TextField()
59 |     status = models.IntegerField('Status', default=0)
60 |
61 |     # This is to display the data of the appointment by the user's name, last name, appointment's date and time
62 |     def __str__(self):
63 |         return self.patientUser.user.first_name+" "+self.patientUser.user.last_name+" ("+str(self.date)+") ("+str(s
64 |
65 |     # This is a model for Report
66 |     class Report(models.Model):
67 |         appointment = models.OneToOneField(Appointment, unique=True, blank=False, null=False, on_delete=models.CASCADE)
68 |         problems = models.TextField()
69 |         prescription = models.TextField()
70 |         other_details = models.TextField()
71 |
72 |         # This is to display the data of the report by the user's name, last name, appointment's date and time
73 |         def __str__(self):
74 |             return self.appointment.patientUser.user.first_name+" "+self.appointment.patientUser.user.last_name+" ("+str(
75 |
76 |     # This is a model to collect the data from the contact us form in the home page
77 |     class contact(models.Model):
78 |         # This is a model to collect the data from the contact us form in the home page
79 |         def __str__(self):
80 |             return self.name
81 |
82 |
83 |
84 |
85 |
86 |
87 |
88 |
89 |
Ln 89, Col 1  Spaces: 4  UTF-8  CRLF  Python  3.8.10 64-bit (windows store)  Idle: ready

```

Figure 61: Database Implementation Code – 3

```

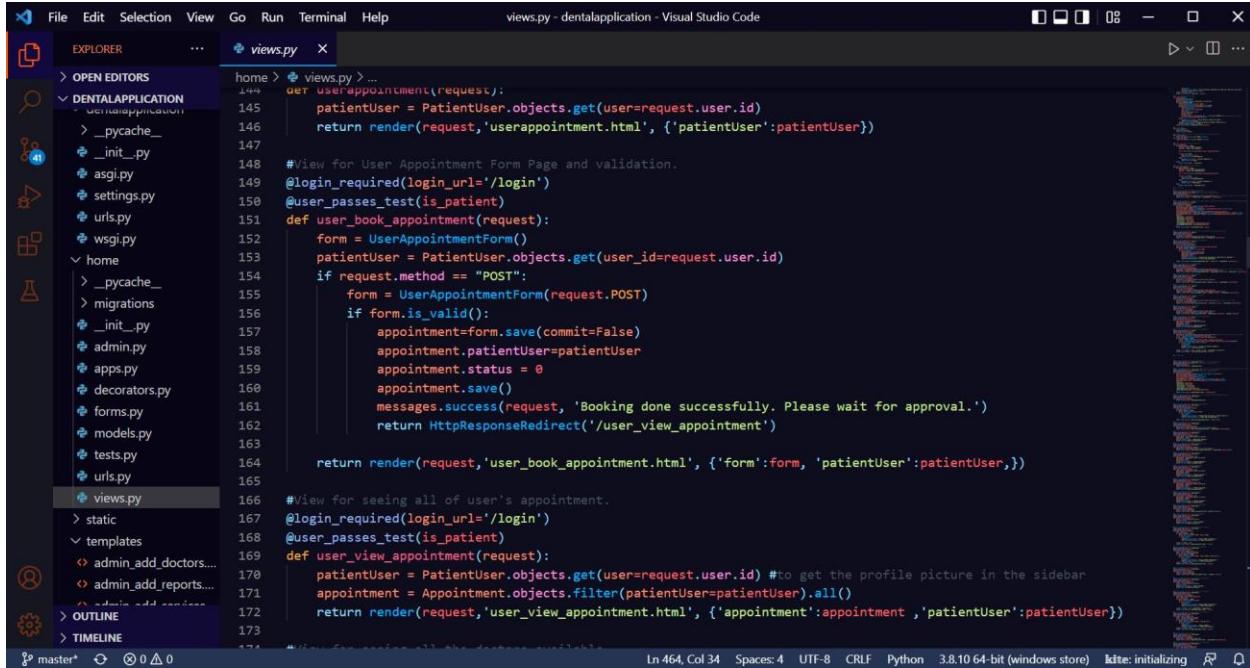
File Edit Selection View Go Run Terminal Help
models.py - dentalapplication - Visual Studio Code
OPEN EDITORS
DENTALAPPLICATION
apps.cpython-38.pyc
decorators.cpython-38...
forms.cpython-38...
models.cpython-38...
urls.cpython-38.pyc
views.cpython-38.p...
migrations
__init__.py
admin.py
apps.py
decorators.py
forms.py
models.py M
tests.py
urls.py
views.py
static
css
# dashboardstyle.css
# style.css
img
profile_pic
templates
home > models.py ...
74 |     def __str__(self):
75 |         return self.appointment.patientUser.user.first_name+" "+self.appointment.patientUser.user.last_name+" ("+str(
76 |
77 |
78 |     # This is a model to collect the data from the contact us form in the home page
79 |     class contact(models.Model):
80 |         name = models.CharField(max_length=100)
81 |         email = models.EmailField(max_length=100)
82 |         subject = models.CharField(max_length=100)
83 |         phone_number = models.CharField(max_length=15)
84 |         message = models.TextField()
85 |
86 |
87 |
88 |
89 |
Ln 89, Col 1  Spaces: 4  UTF-8  CRLF  Python  3.8.10 64-bit (windows store)  Idle: ready

```

Figure 62: Database Implementation Code – 4

3.7.4 Backend Implementation

User Book Appointment and View Appointment



```

File Edit Selection View Go Run Terminal Help
views.py - dentalapplication - Visual Studio Code
OPEN EDITORS
DENTALAPPLICATION
  - dentalapplication
    > __pycache__
      __init__.py
      asgi.py
      settings.py
      urls.py
      wsgi.py
    home
      > __pycache__
      __init__.py
      admin.py
      apps.py
      decorators.py
      forms.py
      models.py
      tests.py
      urls.py
    views.py
      static
      templates
        admin_add_doctors...
        admin_add_reports...
        admin_add_reviews...
    OUTLINE
    TIMELINE
views.py
home > views.py > ...
144     user_appointment(request):
145         patientUser = PatientUser.objects.get(user=request.user.id)
146         return render(request,'userappointment.html', {'patientUser':patientUser})
147
148     #View for User Appointment Form Page and validation.
149     @login_required(login_url='/Login')
150     @user_passes_test(is_patient)
151     def user_book_appointment(request):
152         form = UserAppointmentForm()
153         patientUser = PatientUser.objects.get(user_id=request.user.id)
154         if request.method == "POST":
155             form = UserAppointmentForm(request.POST)
156             if form.is_valid():
157                 appointment=form.save(commit=False)
158                 appointment.patientUser=patientUser
159                 appointment.status = 0
160                 appointment.save()
161                 messages.success(request, 'Booking done successfully. Please wait for approval.')
162                 return HttpResponseRedirect('/user_view_appointment')
163
164     return render(request,'user_book_appointment.html', {'form':form, 'patientUser':patientUser})
165
166     #View for seeing all of user's appointment.
167     @login_required(login_url='/Login')
168     @user_passes_test(is_patient)
169     def user_view_appointment(request):
170         patientUser = PatientUser.objects.get(user=request.user.id) #to get the profile picture in the sidebar
171         appointment = Appointment.objects.filter(patientUser=patientUser).all()
172         return render(request,'user_view_appointment.html', {'appointment':appointment , 'patientUser':patientUser})
173

```

Ln 464, Col 34 Spaces: 4 UTF-8 CRLF Python 3.8.10 64-bit (windows store) Idle: initializing

Figure 63: Backend Implementation Code - 1

User Cancel Appointment and User Edit Profile

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

- File Explorer (Left):** Shows the project structure under "DENTALAPPLICATION". The "views.py" file is currently selected.
- Code Editor (Center):** Displays the content of "views.py". The code defines two main functions: `user_cancel_appointment` and `user_edit_profile`.
 - `user_cancel_appointment` handles a POST request to cancel a user's appointment. It retrieves the appointment by pk, sets its status to 3, saves it, and then shows a success message.
 - `user_edit_profile` handles a POST request to edit a user's profile. It retrieves the patient user, creates forms for updating user and patient user details, and saves them if validation passes. It then shows a success message and redirects to the user's profile view.
- Bottom Status Bar:** Shows the current file is "views.py - dentalapplication - Visual Studio Code", along with other status indicators like line numbers (Ln 464), column numbers (Col 34), and file type (Python).

Figure 64: Backend Implementation Code - 2

[Click here to view all the codes for the automation script \(backend\).](#)

4. Testing and analysis

Software Testing is a method to check whether the actual software product matches expected requirements and to ensure that software product is Defect free. (Hamilton, 2022) Any software developer's ultimate goal is to create apps that are fast, stable, and simple to use. Each application must be properly tested in order to achieve this goal.

4.1 Test Plan

There are 5 types of testing that are done for this application. They are as follows:

4.1.1 Black Box Testing, Test Plan

Black Box Testing is a software testing method in which the functionalities of software applications are tested without having knowledge of internal code structure, implementation details and internal paths. (Hamilton, 2022) It majorly focuses on the input and output of software applications and is totally driven by software requirements and specifications.

Test Cases	Objectives
Test Case 1	To view user's profile picture in the sidebar.
Test Case 2	To book an appointment by the user.
Test Case 3	To view user's appointment by user.
Test Case 4	To check if we can access admin page without logging in.
Test Case 5	To view user's report by the user.
Test Case 6	To book an appointment outside the clinic time.
Test Case 7	To view pending appointment by the admin.
Test Case 8	To delete user from the system by the admin.

Table 12: Black Box Testing Test Plan

4.1.2 White Box Testing, Test Plan

White box testing is an approach that allows testers to inspect and verify the inner workings of a software system—its code, infrastructure, and integrations with external systems. (Imperva, 2022) In a modern Continuous Integration/Continuous Delivery (CI/CD) development pipeline, white box testing is an important aspect of automated build operations.

Test Cases	Objectives
Test Case 1	To register user.
Test Case 2	To log in user.
Test Case 3	To enter invalid login details.
Test Case 4	To log in admin.
Test Case 5	To cancel appointment by user.
Test Case 6	To view doctor by user.
Test Case 7	To view service by user.
Test Case 8	To download the report in pdf by the user.
Test Case 9	To sort the appointment according to the date.
Test Case 10	To search the report.
Test Case 11	To view user's profile.
Test Case 12	To logout.
Test Case 13	To approve appointment of the user by the admin.
Test Case 14	To reject appointment of the user by the admin.
Test Case 15	To cancel the confirmed appointment of the user by the admin.
Test Case 16	To finish the appointment of the user by the admin.
Test Case 17	To add another report of the same appointment.

Table 13: White Box Testing Test Plan

4.1.3 Integration Testing, Test Plan

Integration testing is a type of software testing in which the different units, modules or components of a software application are tested as a combined entity. (Awati, 2022)

Test Cases	Objectives
Test Case 1	To update the user profile by the user.
Test Case 2	To add report by the admin.
Test Case 3	To update the report details by the admin.
Test Case 4	To delete the report by the admin.
Test Case 5	To add service by the admin.
Test Case 6	To update the service details by the admin.
Test Case 7	To delete the service by the admin.
Test Case 8	To add doctor by the admin.
Test Case 9	To update the doctor details by the admin.
Test Case 10	To delete the doctor by the admin

Table 14: Integration Testing Test Plan

4.1.4 System Testing, Test Plan

System Testing is a type of software testing that is performed on a complete integrated system to evaluate the compliance of the system with the corresponding requirements. (Pankaj, 2019) As majority of the tests will be performed in the black box, white box, and integration testing, system testing is performed to determine whether or not the system is operational.

Test Cases	Objectives
Test Case 1	To run the system from VS Code.

Table 15: System Testing Test Plan

4.1.5 Client Evaluation Testing, Test Plan

Client evaluation testing is a type of testing done by the client to ensure that the software system is working properly before it is deployed to the production environment. After all other testing has been completed, it is done in the last step of testing.

Link to client evaluation testing form: <https://forms.gle/PR2q5TZUwZUAEcyu9>

4.2 Black Box Testing

Test No.1	
Objective	To view user's profile picture in the sidebar.
Action	The user logged in and the 'View Services' button was pressed.
Expected Output	The profile picture of the user would be shown in the sidebar.
Actual Output	The profile picture of the user was not shown in the sidebar.
Result	The test was unsuccessful.

Table 16: Black Box Test 1 (Unsuccessful)

Name	Cost	Description
Braces	28000	We do all kinds of braces implantation to align your teeth. The base price starts from Rs.28000.
General Visit	350	We do a thorough check up on this and recommend necessary treatments.
RCT	4500	Root Canal Treatment (RCT) is a procedure done when the tooth decay reaches the pulp (innermost layer of the tooth) causing inflammation.
Scaling	1500	Scaling is the cleaning of the teeth.
Tooth Extraction	15000	Tooth extraction is done when the teeth needs to be removed. The costs started from 15000 and can be raised according to the complexity.

Figure 65: Black Box Test 1 (Unsuccessful)

Code that caused the error:

```
#View for seeing all of the services available.
@login_required(login_url='/login')
@user_passes_test(is_patient)
def user_view_services(request):
    services = Service.objects.all
    return render(request,'user_view_services.html', {'services':services})
```

Figure 66: Black Box Test 1 - Error Code

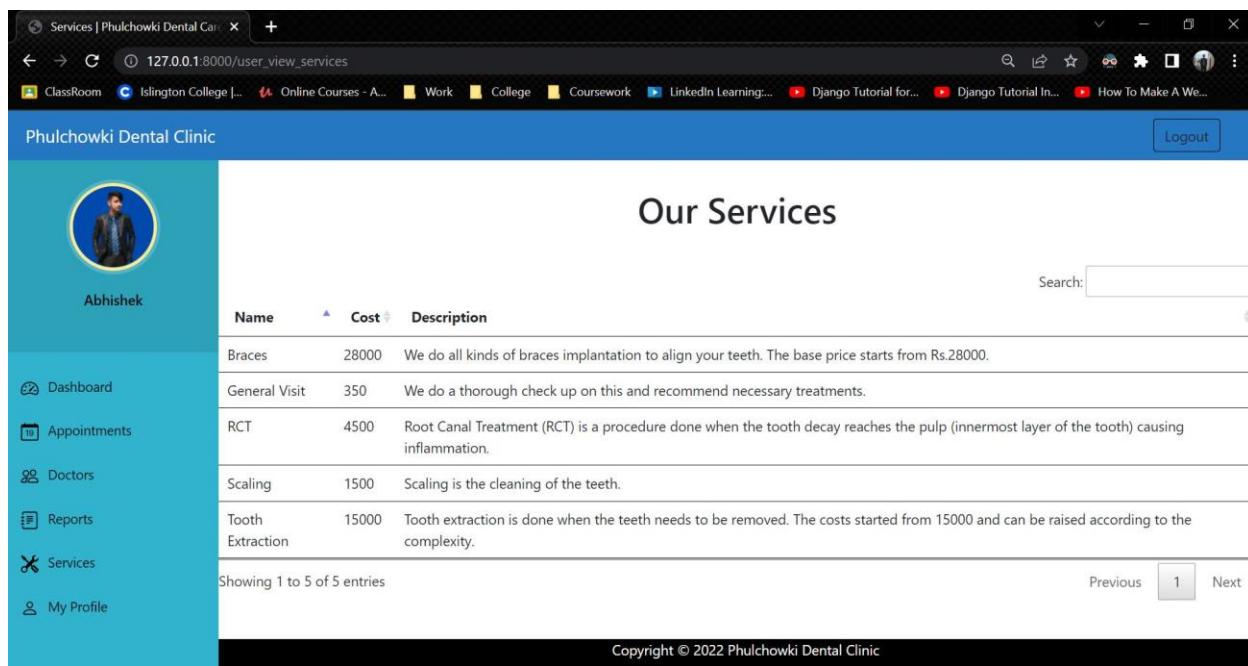
Fixing the code:

```
#view for seeing all of the services available.
@login_required(login_url='/login')
@user_passes_test(is_patient)
def user_view_services(request):
    patientUser = PatientUser.objects.get(user=request.user.id) #to get logged in user's image |
    services = Service.objects.all
    return render(request,'user_view_services.html', {'services':services , 'patientUser':patientUser})
```

Figure 67: Black Box Test 1 - Fixed Code

Test No.1	
Objective	To view user's profile picture in the sidebar.
Action	The user logged in and the 'View Services' button was pressed.
Expected Output	The profile picture of the user would be shown in the sidebar.
Actual Output	The profile picture of the user was not shown in the sidebar.
Result	The test was successful.
Cause of Error	The current user's ID wasn't

Table 17: Black Box Test 1 (Successful)



The screenshot shows a web browser window with the title 'Services | Phulchowki Dental Care'. The address bar displays '127.0.0.1:8000/user_view_services'. The page content is titled 'Our Services' and lists five dental services with their names, costs, and descriptions. A sidebar on the left contains links for Dashboard, Appointments, Doctors, Reports, Services, and My Profile. The user's profile picture, 'Abhishek', is visible in the sidebar. The bottom of the page includes copyright information and navigation buttons for 'Previous', 'Next', and page number '1'.

Name	Cost	Description
Braces	28000	We do all kinds of braces implantation to align your teeth. The base price starts from Rs.28000.
General Visit	350	We do a thorough check up on this and recommend necessary treatments.
RCT	4500	Root Canal Treatment (RCT) is a procedure done when the tooth decay reaches the pulp (innermost layer of the tooth) causing inflammation.
Scaling	1500	Scaling is the cleaning of the teeth.
Tooth Extraction	15000	Tooth extraction is done when the teeth needs to be removed. The costs started from 15000 and can be raised according to the complexity.

Figure 68: Black Box Test 1 (Successful)

Test No.2	
Objective	To book an appointment by the user.
Action	The booking form is filled and submitted by the user.
Expected Output	The booking would be done.
Actual Output	The booking was not done.
Result	The test was unsuccessful.

Table 18: Black Box Test 2 (Unsuccessful)

The screenshot shows a web browser window with the title 'Book Appointments | Phulchowki'. The URL in the address bar is '127.0.0.1:8000/user.book.appointment'. The page itself is titled 'Book an Appointment' and contains fields for 'Doctor' (set to 'Deeplab Acharya'), 'Service' (set to 'General Visit'), 'Date' (set to '04/26/2022'), and 'Time' (set to '03:45 PM'). Below these fields is a text area containing the message 'Need to get my teeth checked'. The left sidebar of the application is visible, showing a profile picture of 'Kushal' and a navigation menu with options like 'Dashboard', 'Appointments', 'Doctors', 'Reports', 'Services', and 'My Profile'. A 'Logout' button is located in the top right corner of the main content area. At the bottom of the page, a copyright notice reads 'Copyright © 2022 Phulchowki Dental Clinic'.

Figure 69: Black Box Test 2 (Unsuccessful) - 1

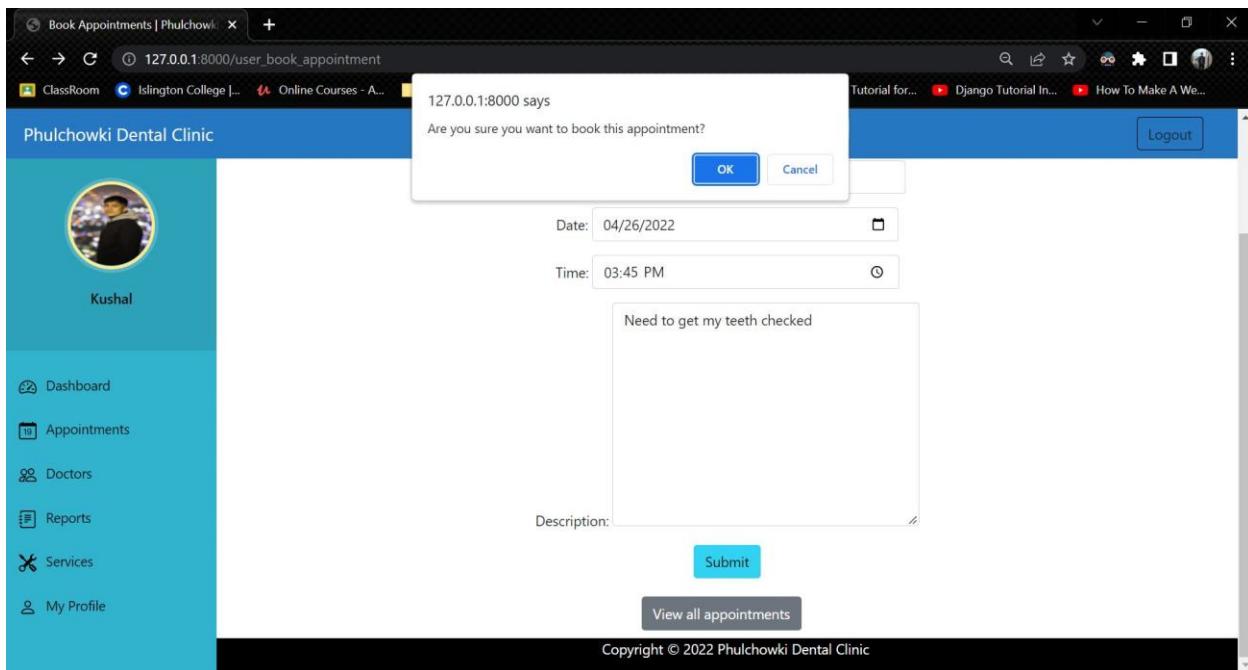


Figure 70: Black Box Test 2 (Unsuccessful) - 2

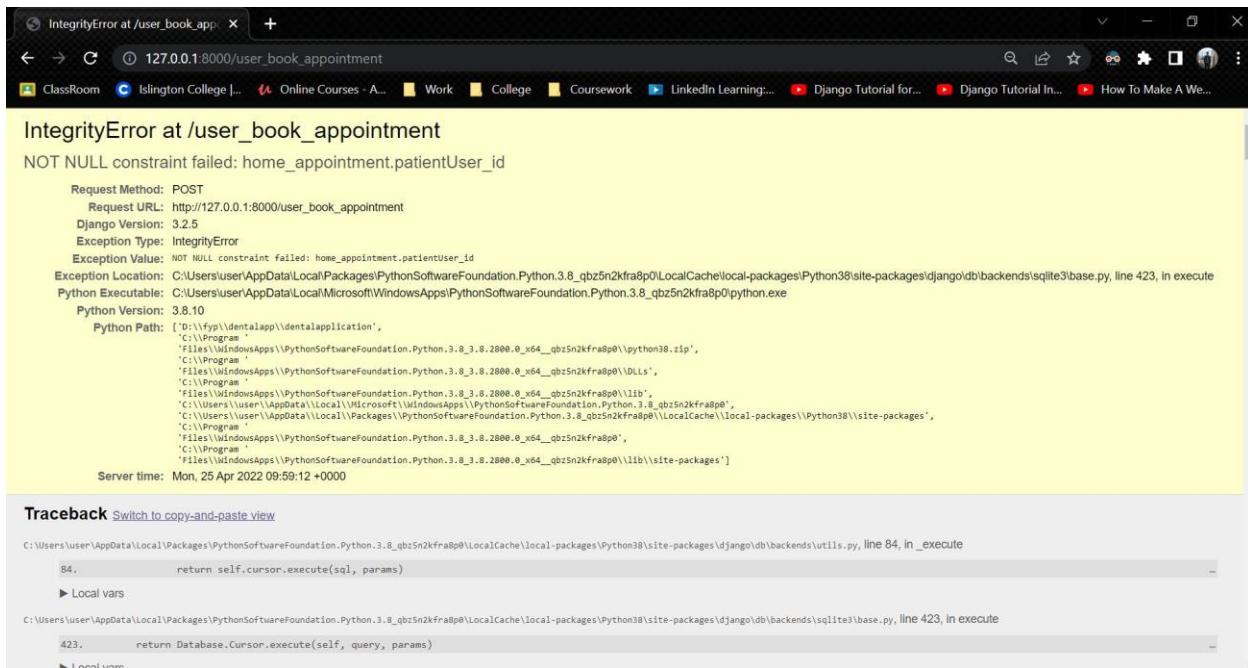


Figure 71: Black Box Test 2 (Unsuccessful) – 3

Code that caused the error:

```
#View for User Appointment Form Page and validation.
@login_required(login_url='/login')
@user_passes_test(is_patient)
def user_book_appointment(request):
    form = UserAppointmentForm()
    patientUser = PatientUser.objects.get(user_id=request.user.id)
    if request.method == "POST":
        form = UserAppointmentForm(request.POST)
        if form.is_valid():
            appointment=form.save()
            appointment.patientUser=patientUser
            appointment.status = 0
            appointment.save()
            messages.success(request, 'Booking done successfully. Please wait for approval.')
            return HttpResponseRedirect('/user_view_appointment')

    return render(request,'user_book_appointment.html', {'form':form, 'patientUser':patientUser,})
```

Figure 72: Black Box Test 2 - Error Code

Fixing the code:

```
#View for User Appointment Form Page and validation.
@login_required(login_url='/login')
@user_passes_test(is_patient)
def user_book_appointment(request):
    form = UserAppointmentForm()
    patientUser = PatientUser.objects.get(user_id=request.user.id)
    if request.method == "POST":
        form = UserAppointmentForm(request.POST)
        if form.is_valid():
            appointment=form.save(commit=False)
            appointment.patientUser=patientUser
            appointment.status = 0
            appointment.save()
            messages.success(request, 'Booking done successfully. Please wait for approval.')
            return HttpResponseRedirect('/user_view_appointment')

    return render(request,'user_book_appointment.html', {'form':form, 'patientUser':patientUser,})
```

Figure 73: Black Box Test 2 - Fixed Code

Test No.2	
Objective	To book an appointment by the user.
Action	The booking form is filled and submitted by the user.
Expected Output	The booking would be done.
Actual Output	The booking was done.
Result	The test was successful.
Cause of Error	The form was saved before getting the current user ID.

Table 19: Black Box Test 2 (Successful)

Figure 74: Black Box Test 2 (Successful) - 1

The screenshot shows a web browser window for 'Phulchowki Dental Clinic'. The URL is 127.0.0.1:8000/user_view_appointment. The page displays a message 'Booking done successfully. Please wait for approval.' and a table titled 'Your Appointments'.

Left Sidebar:

- Profile picture of 'Kushal'
- Navigation menu:
 - Dashboard
 - Appointments
 - Doctors
 - Reports
 - Services
 - My Profile

Main Content Area:

Message: Booking done successfully. Please wait for approval.

Your Appointments

Name	Date	Time	Service	Doctor	Description	Status
Kushal Maharjan	April 26, 2022	3:45 p.m.	General Visit	Deeplab Acharya	Need to get my teeth checked	Pending

Showing 1 to 1 of 1 entries

Search:

Buttons: Previous, Next, Book new appointment

Copyright © 2022 Phulchowki Dental Clinic

Figure 75: Black Box Test 2 (Successful) - 2

Test No.3	
Objective	To view user's appointment by user.
Action	The view appointment button in the sidebar was clicked.
Expected Output	The list of user's appointment would be shown.
Actual Output	The list of user's appointment was not shown.
Result	The test was unsuccessful.

Table 20: Black Box Test 3 (Unsuccessful)

The screenshot shows a web application interface for a dental clinic named 'Phulchowki Dental Clinic'. The user is logged in as 'Abhishek'. The main page is titled 'Your Appointments'. On the left, there is a sidebar with a user profile picture and the name 'Abhishek'. Below the profile are links to 'Dashboard', 'Appointments', 'Doctors', 'Reports', 'Services', and 'My Profile'. The main content area has a heading 'Your Appointments' and a table with columns: Name, Date, Time, Service, Doctor, Description, and Status. A message 'No data available in table' is displayed. At the bottom of the page, there is a search bar, a copyright notice 'Copyright © 2022 Phulchowki Dental Clinic', and a 'Book new appointment' button.

Figure 76: Black Box Test 3 (Unsuccessful)

Code that caused the error:

```
#View for seeing all of user's appointment.  
@login_required(login_url='/login')  
@user_passes_test(is_patient)  
def user_view_appointment(request):  
    patientUser = PatientUser.objects.get(user=request.user.id) #to get the profile picture in the sidebar  
    appointment = Appointment.objects.filter(patientUser=request.user.id).all()  
    return render(request,'user_view_appointment.html', {'appointment':appointment , 'patientUser':patientUser})
```

Figure 77: Black Box Test 3 - Error Code

Fixing the code:

```
#View for seeing all of user's appointment.  
@login_required(login_url='/login')  
@user_passes_test(is_patient)  
def user_view_appointment(request):  
    patientUser = PatientUser.objects.get(user=request.user.id) #to get the profile picture in the sidebar  
    appointment = Appointment.objects.filter(patientUser=patientUser).all()  
    return render([request,'user_view_appointment.html', {'appointment':appointment , 'patientUser':patientUser}])
```

Figure 78: Black Box Test 3 - Fixed Code

Test No.3	
Objective	To view user's appointment by user.
Action	The view appointment button in the sidebar was clicked.
Expected Output	The list of user's appointment would be shown.
Actual Output	The list of user's appointment was shown.
Result	The test was successful.
Cause of Error	The filter query was wrong so it didn't display anything.

Table 21: Black Box Test 3 (Successful)

Your Appointments

Name	Date	Time	Service	Doctor	Description	Status
Abhishek Rokaya	April 28, 2022	2:45 p.m.	Tooth Extraction	Prasamhsa Khadka	I need to extract my wisdom tooth.	Finished
Abhishek Rokaya	April 29, 2022	12:50 p.m.	General Visit	Animesh Basnet	General checkup needed.	Finished
Abhishek Rokaya	May 2, 2022	2 p.m.	Scaling	Animesh Basnet	I need to get my teeth cleaned.	Finished
Abhishek Rokaya	May 6, 2022	4 p.m.	General Visit	Rabina Lama	Need a general visit.	Pending
Abhishek Rokaya	April 26, 2022	2:01 p.m.	General Visit	Prasamhsa Khadka	A general visit with the doctor	Finished
Abhishek Rokaya	April 26, 2022	4:30 p.m.	Tooth Extraction	Prasamhsa Khadka	need to get an extraction.	Confirmed

Showing 1 to 6 of 6 entries

Copyright © 2022 Phulchowki Dental Clinic

Figure 79: Black Box Test 3 (Successful)

Test No.4	
Objective	To check if we can access admin page without logging in.
Action	The path to admin dashboard page was entered while not being logged in.
Expected Output	The admin login page would be shown.
Actual Output	The admin dashboard page was shown.
Result	The test was unsuccessful.

Table 22: Black Box Test 4 (Unsuccessful)

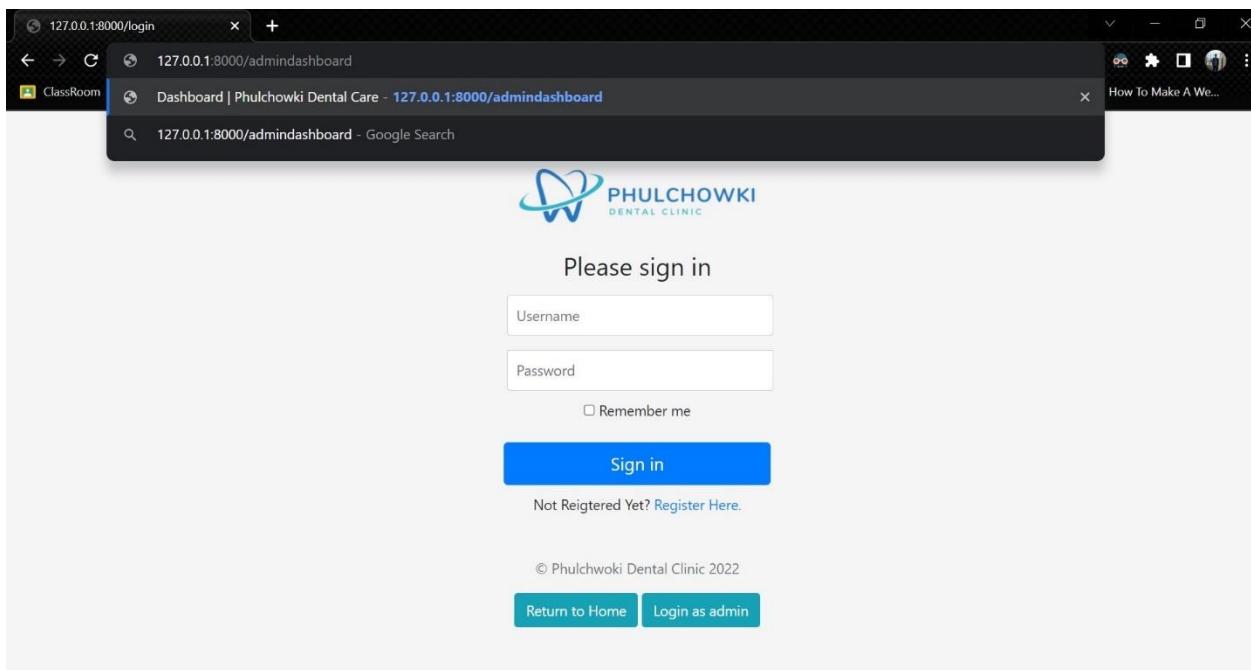


Figure 80: Black Box Test 4 (Unsuccessful) - 1

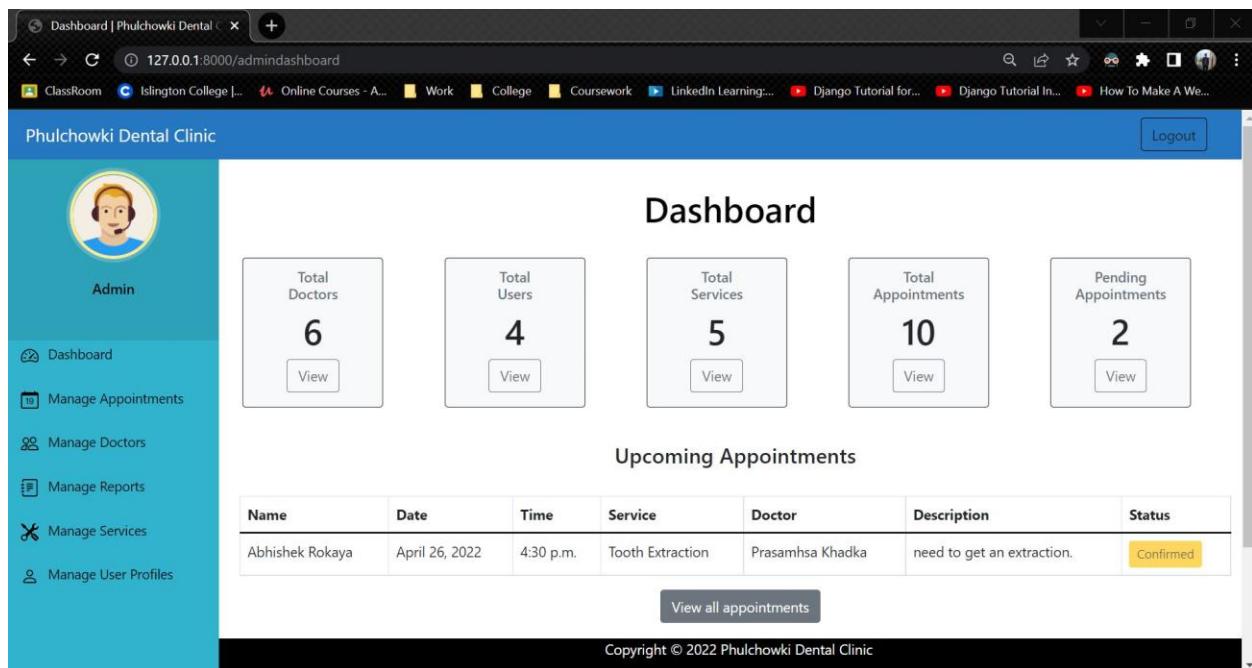


Figure 81: Black Box Test 4 (Unsuccessful) – 2

Code that caused the error:

```
def admindashboard(request):
    appointment=Appointment.objects.all().order_by('-id').filter(status=1)
    doctorcount=Doctor.objects.all().count()
    patientcount=PatientUser.objects.all().count()
    servicecount=Service.objects.all().count()
    appointmentcount = Appointment.objects.all().filter(status__gte=1).count()
    pendingappointmentcount = Appointment.objects.all().filter(status=0).count()
    mydict={
        'appointment':appointment,
        'doctorcount':doctorcount,
        'patientcount': patientcount,
        'servicecount': servicecount,
        'appointmentcount':appointmentcount,
        'pendingappointmentcount':pendingappointmentcount,
    }
    return render(request,'admindashboard.html', context=mydict)
```

Figure 82: Black Box Test 4 - Error Code

Fixing the code:

```
#Admin's dashboard where we can see the total doctor, service, patient and appointment count along with the upcoming appointments.
@login_required(login_url='/adminlogin') #Decorators for authentication
@user_passes_test(is_admin) #Checking if the user is admin or not
def admindashboard(request):
    appointment=Appointment.objects.all().order_by('-id').filter(status=1)
    doctorcount=Doctor.objects.all().count()
    patientcount=PatientUser.objects.all().count()
    servicecount=Service.objects.all().count()
    appointmentcount = Appointment.objects.all().filter(status__gte=1).count()
    pendingappointmentcount = Appointment.objects.all().filter(status=0).count()
    mydict={
        'appointment':appointment,
        'doctorcount':doctorcount,
        'patientcount': patientcount,
        'servicecount': servicecount,
        'appointmentcount':appointmentcount,
        'pendingappointmentcount':pendingappointmentcount,
    }
    return render(request,'admindashboard.html', context=mydict)
```

Figure 83: Black Box Test 4 - Fixed Code

Test No.4	
Objective	To check if we can access admin page without logging in.
Action	The path to admin dashboard page was entered while not being logged in.
Expected Output	The admin login page would be shown.
Actual Output	The admin login page was shown.
Result	The test was successful.
Cause of Error	The decorators was not declared in the code.

Table 23: Black Box Test 4 (Successful)

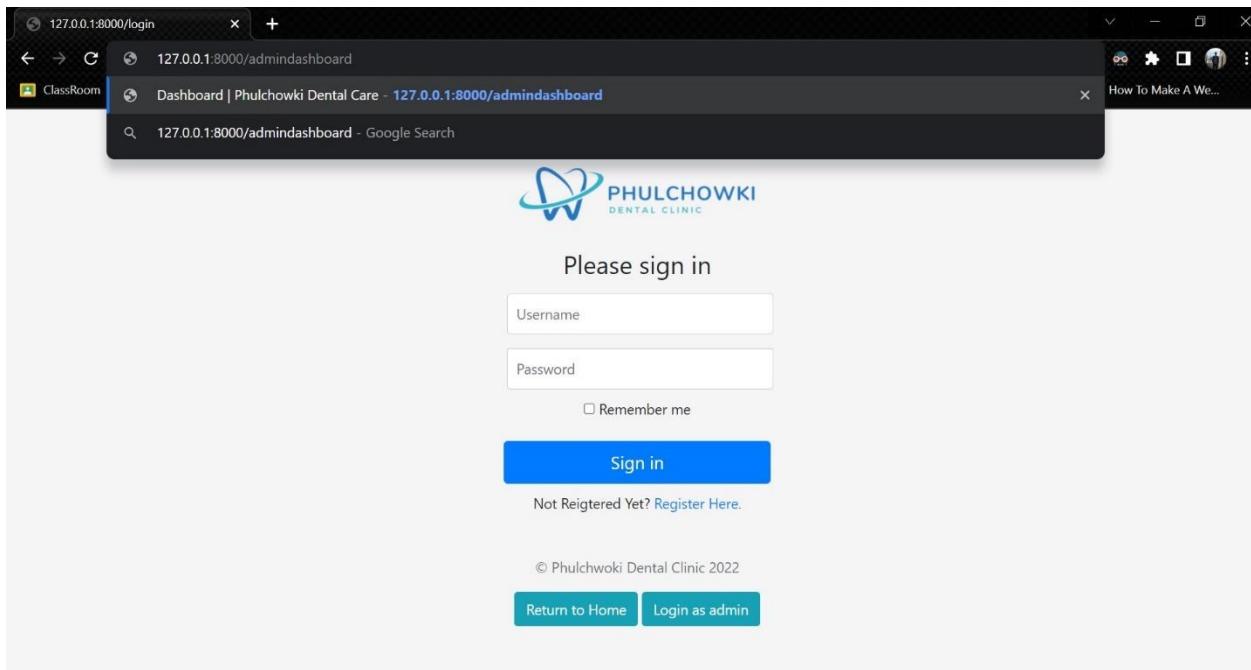


Figure 84: Black Box Test 4 (Successful) - 1

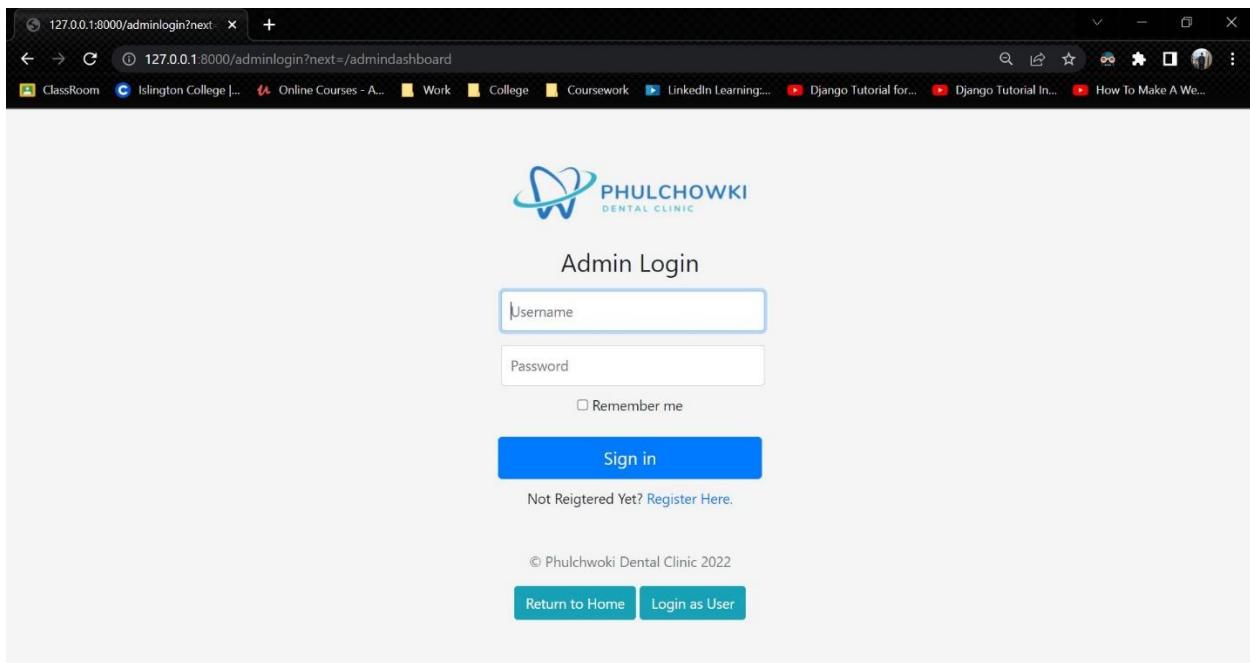


Figure 85: Black Box Test 4 (Successful) - 2

[The rest of the black box testing is continued here. Click here to see the rest of it.](#)

4.3 White Box Testing

Test No.1	
Objective	To register user.
Action	The 'register' button was clicked and the user details were filled and the form was submitted.
Expected Output	The user would be registered to the system.
Actual Output	The user was registered to the system.
Result	The test was successful.

Table 24: White Box Test 1

The screenshot shows a web browser window with the URL `127.0.0.1:8000/signup`. The page title is "Register". The form is titled "Register" and contains the following fields:

- First Name
- Last Name
- Username
- Email
- Password
- mm/dd/yyyy
- Phone Number
- Address
- Choose File (No file chosen)

At the bottom of the form is a blue "Register" button. Below the button, a link says "Already Registered? Sign In Here." At the very bottom of the page, there is footer text: "© Phulchowki Dental Clinic 2022" and a "Return to Home" button.

Figure 86: White Box Test 1 - 1

The screenshot shows a web browser window with the URL `127.0.0.1:8000/signup`. The page title is "Register". It features a logo for "PHULCHOWKI DENTAL CLINIC". The form fields include:

- Name: Kushal
- Middle Name: Maharjan
- Address: itskushal
- Email: kushalmh@gmail.com
- Password: (obscured)
- Date of Birth: 01/07/2003
- Phone Number: 9803289425
- Address: Balaju, Kathmandu
- Profile Picture: Choose File [selected file: kushal.JPG]

A blue "Register" button is at the bottom. Below it, a link says "Already Registered? Sign In Here." and a copyright notice reads "© Phulchowki Dental Clinic 2022".

Figure 87: White Box Test 1 - 2

The screenshot shows a web browser window with the URL `127.0.0.1:8000/login`. The page title is "Please sign in". It features a logo for "PHULCHOWKI DENTAL CLINIC". The form fields include:

- Username: (empty)
- Password: (empty)
- Remember me: (checkbox)

A blue "Sign in" button is at the bottom. Below it, a green box displays the message "Account Successfully Created. Please Log in.". At the bottom, links say "Not Registered Yet? Register Here.", "Return to Home", and "Login as admin".

Figure 88: White Box Test 1 - 3

Test No.2	
Objective	To log in user.
Action	The 'login' button was clicked and the user credentials were filled and the 'Sign in' button was clicked.
Expected Output	The user would be logged in and redirected to the dashboard.
Actual Output	The user was logged in and redirected to the dashboard.
Result	The test was successful.

Table 25: White Box Test 2

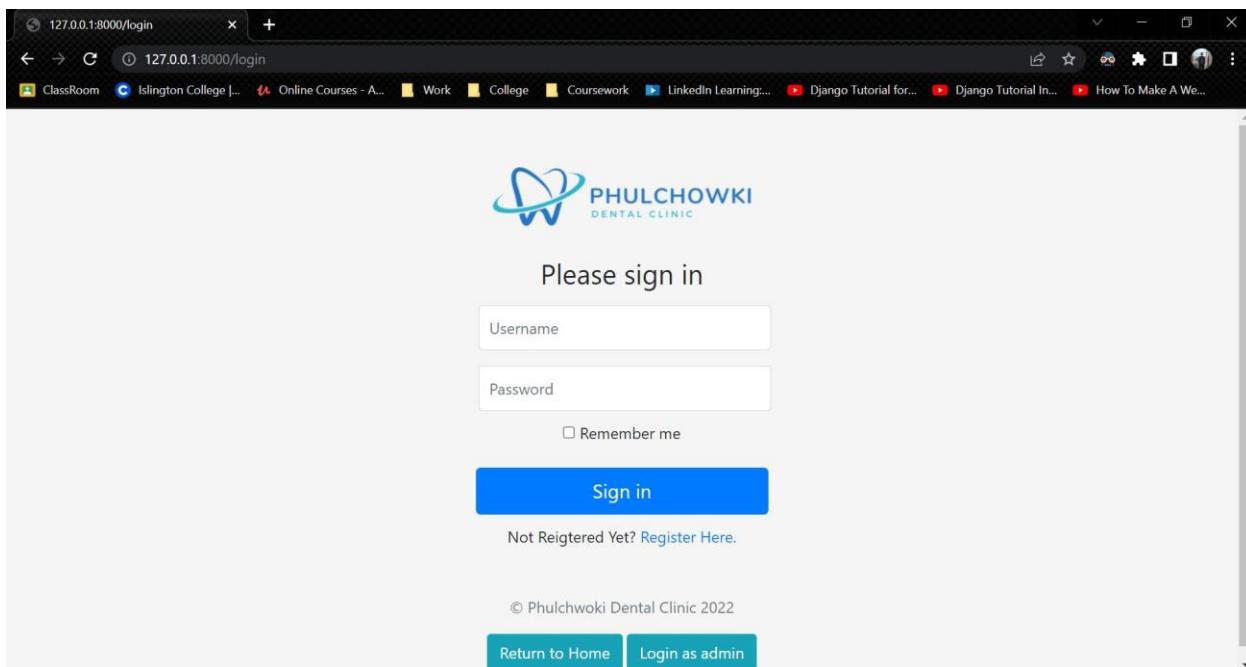


Figure 89: White Box Test 2 - 1

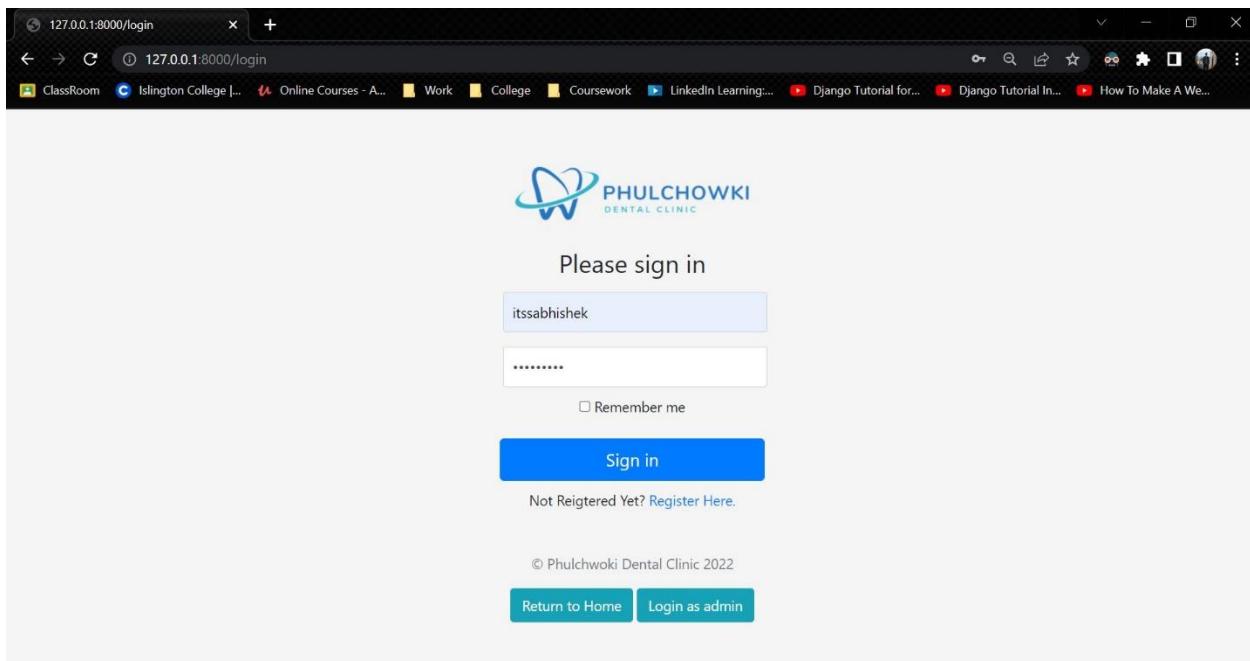


Figure 90: White Box Test 2 - 2

Name	Date	Time	Service	Doctor	Description	Status
Abhishek Rokaya	April 26, 2022	4:30 p.m.	Tooth Extraction	Prasamhsa Khadka	need to get an extraction.	Confirmed

Figure 91: White Box Test 2 - 3

Test No.3	
Objective	To enter invalid login details.
Action	The 'login' button was clicked and the user credentials were filled and the 'Sign in' button was clicked.
Expected Output	An error showing 'Invalid Credentials' would be displayed.
Actual Output	An error showing 'Invalid Credentials' was displayed.
Result	The test was successful.

Table 26: White Box Test 3

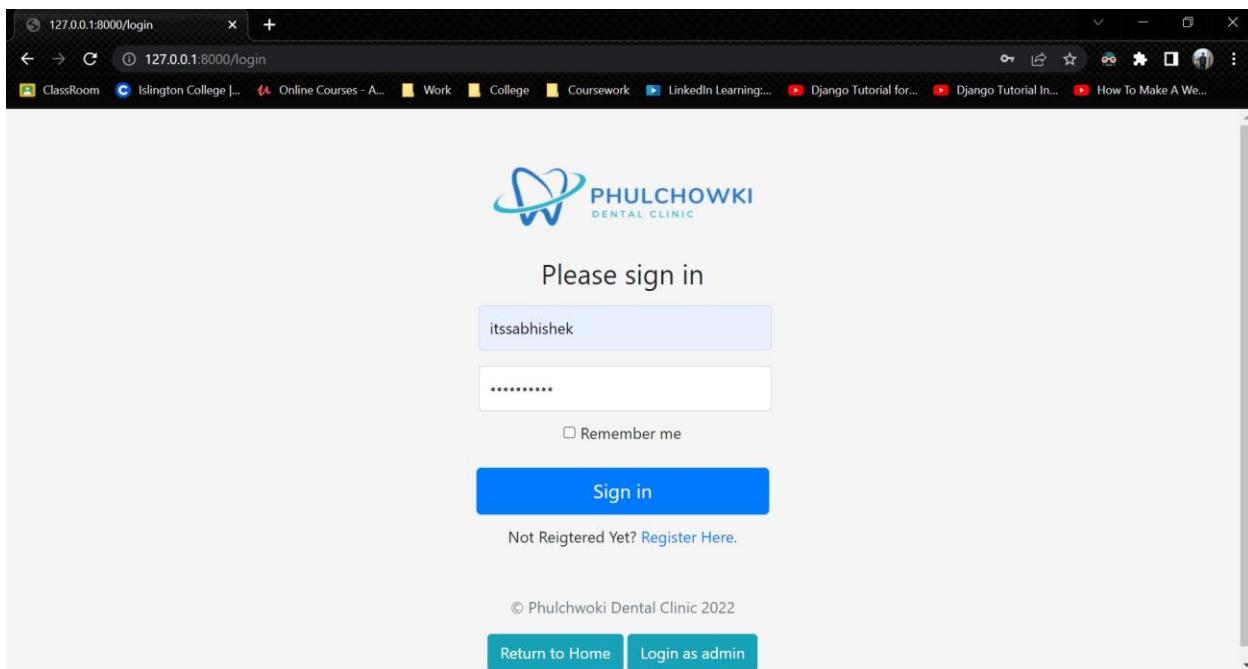


Figure 92: White Box Test 3 - 1

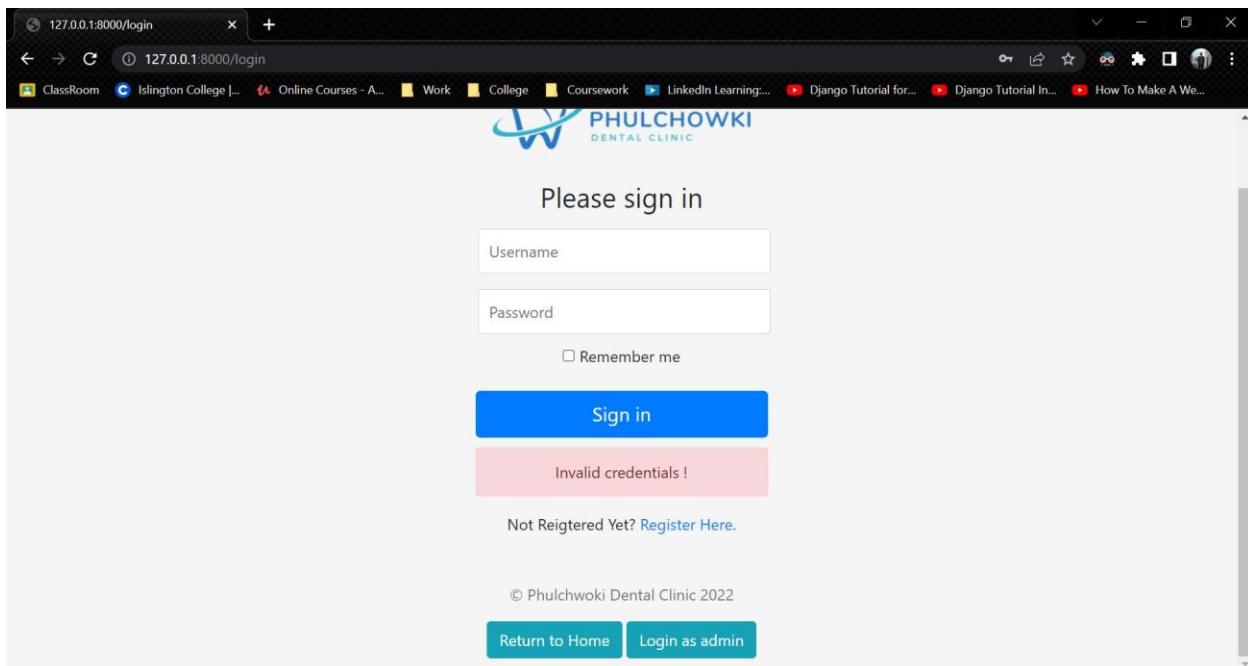


Figure 93: White Box Test 3 - 2

Test No.4	
Objective	To log in admin.
Action	The 'admin login' button was clicked and the admin credentials were filled and the 'Sign in' button was clicked.
Expected Output	The admin would be logged in and redirected to the admin dashboard.
Actual Output	The admin was logged in and redirected to the dashboard.
Result	The test was successful.

Table 27: White Box Test 4

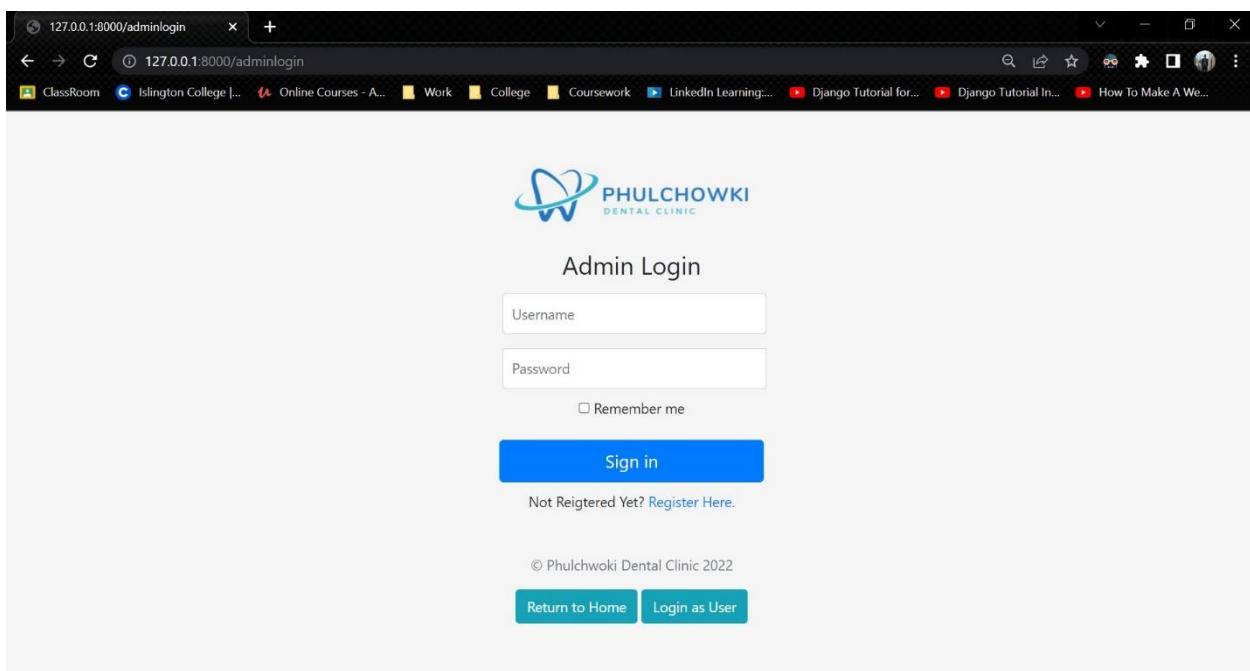


Figure 94: White Box Test 4 - 1

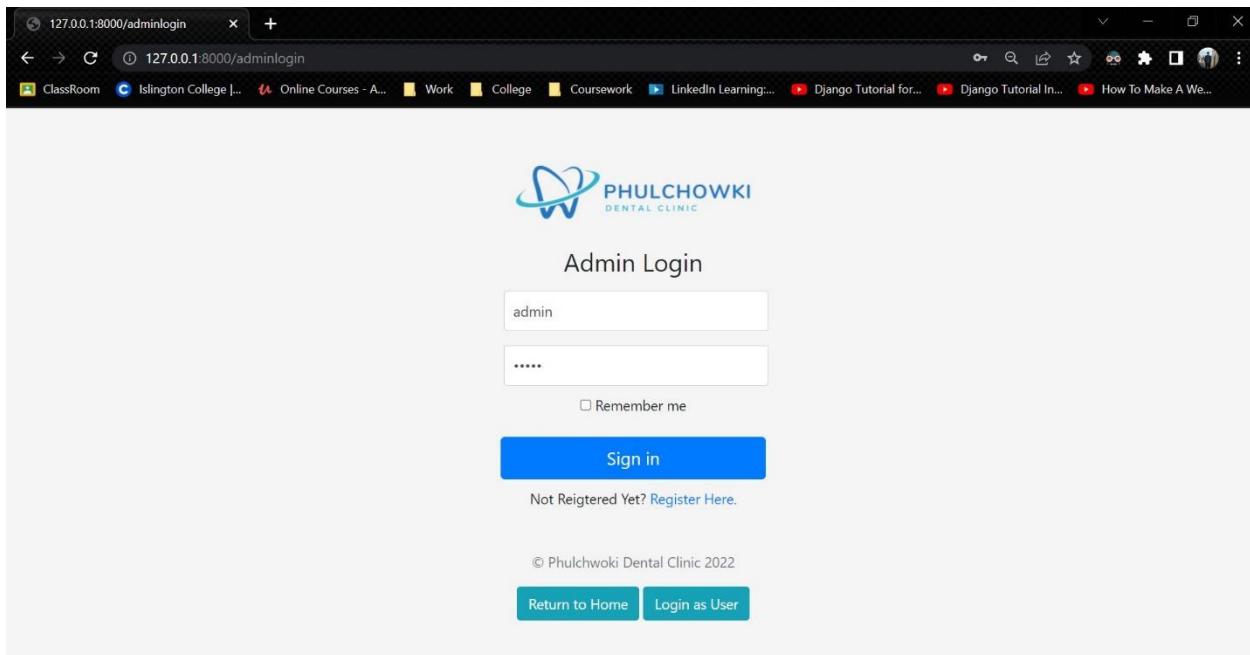


Figure 95: White Box Test 4 - 2

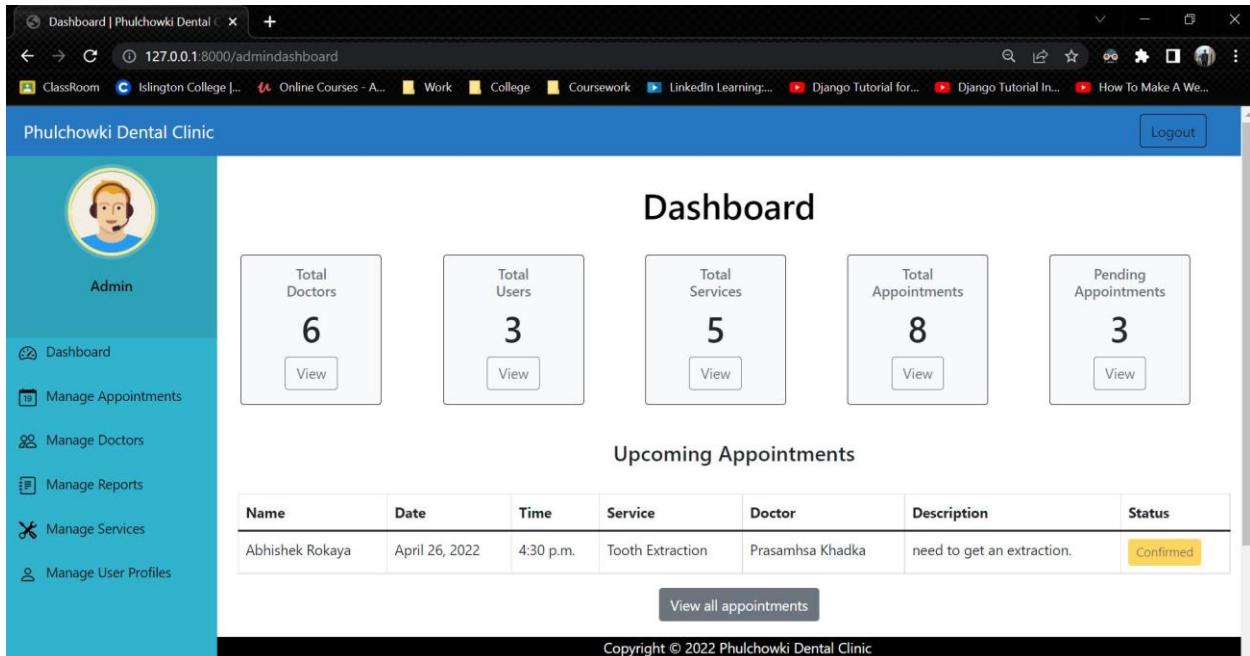


Figure 96: White Box Test 4 - 3

Test No.5	
Objective	To cancel appointment by user.
Action	The 'cancel' button was clicked and the confirmation was clicked.
Expected Output	The appointment would be cancelled.
Actual Output	The appointment was cancelled.
Result	The test was successful.

Table 28: White Box Test 5

Your Appointments

Name	Date	Time	Service	Doctor	Description	Status
Abhishek Rokaya	May 6, 2022	4 p.m.	General Visit	Rabina Lama	Need a general visit.	Pending
Abhishek Rokaya	May 2, 2022	2 p.m.	Scaling	Animesh Basnet	I need to get my teeth cleaned.	Finished
Abhishek Rokaya	April 29, 2022	12:50 p.m.	General Visit	Animesh Basnet	General checkup needed.	Finished
Abhishek Rokaya	April 28, 2022	2:45 p.m.	Tooth Extraction	Prasamhsa Khadka	I need to extract my wisdom tooth.	Finished
Abhishek Rokaya	April 26, 2022	2:01 p.m.	General Visit	Prasamhsa Khadka	A general visit with the doctor	Finished
Abhishek Rokaya	April 26, 2022	4:30 p.m.	Tooth Extraction	Prasamhsa Khadka	need to get an extraction.	Confirmed

Showing 1 to 6 of 6 entries

Copyright © 2022 Phulchowki Dental Clinic

Figure 97: White Box Test 5 - 1

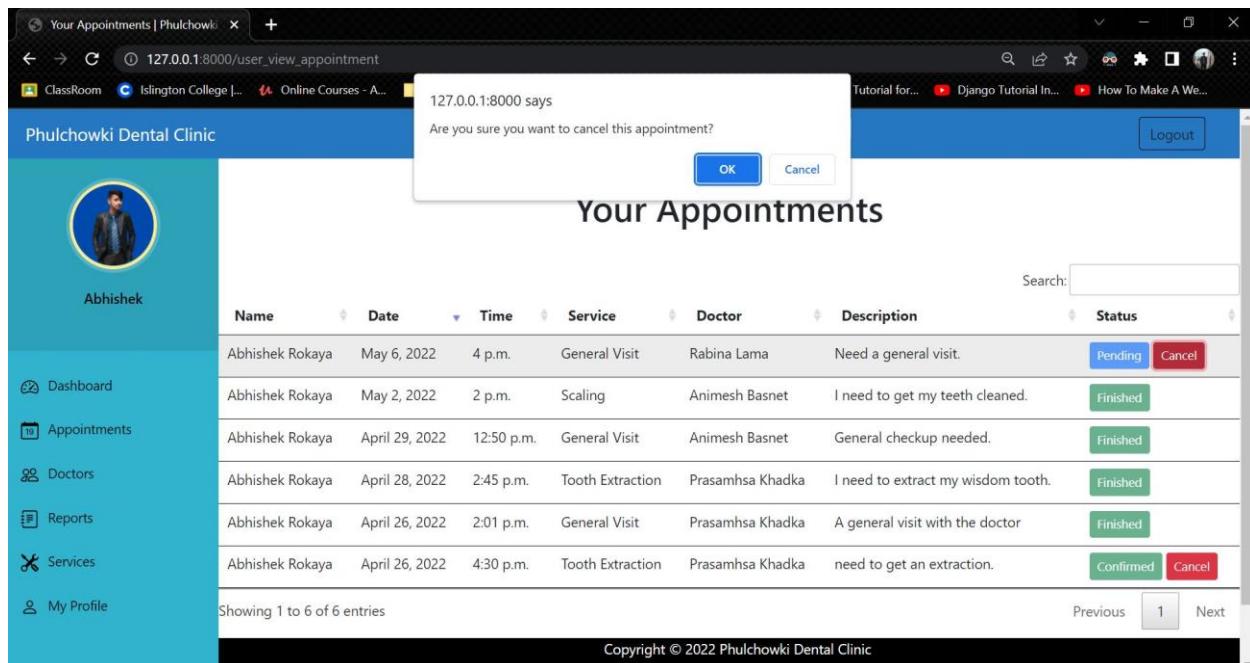


Figure 98: White Box Test 5 – 2

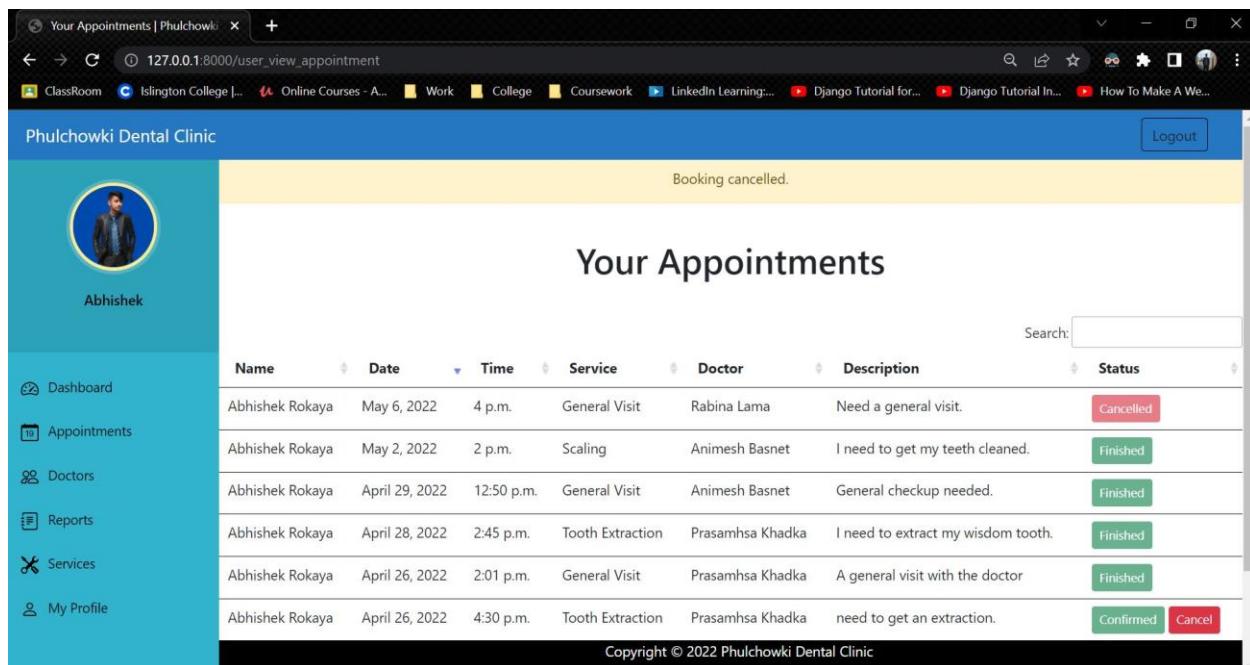


Figure 99: White Box Test 5 – 3

[The rest of the white box testing is continued here. Click here to see the rest of it.](#)

4.4 Integration Testing

Test No.1	
Objective	To update user's profile by the user.
Action	The 'Edit Profile' button was clicked, then the necessary information were updated and the 'update' button was clicked.
Expected Output	The data would be updated.
Actual Output	The data was updated.
Result	The test was successful.

Table 29: Integration Test 1

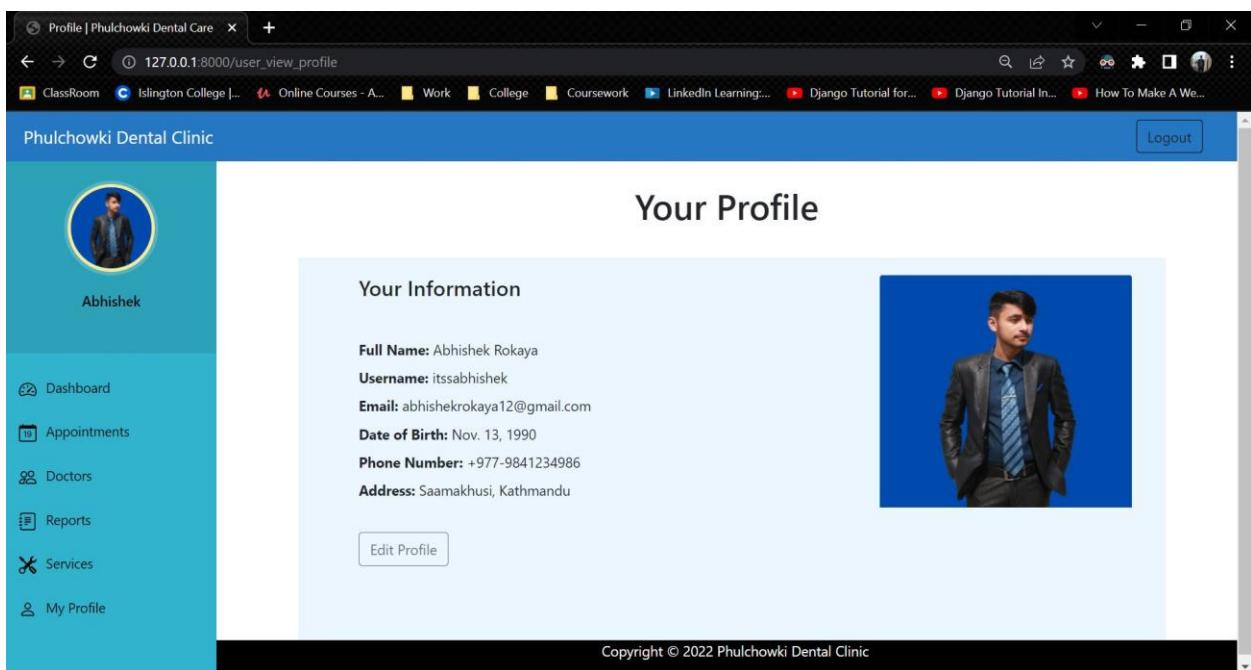


Figure 100: Integration Test 1 - 1

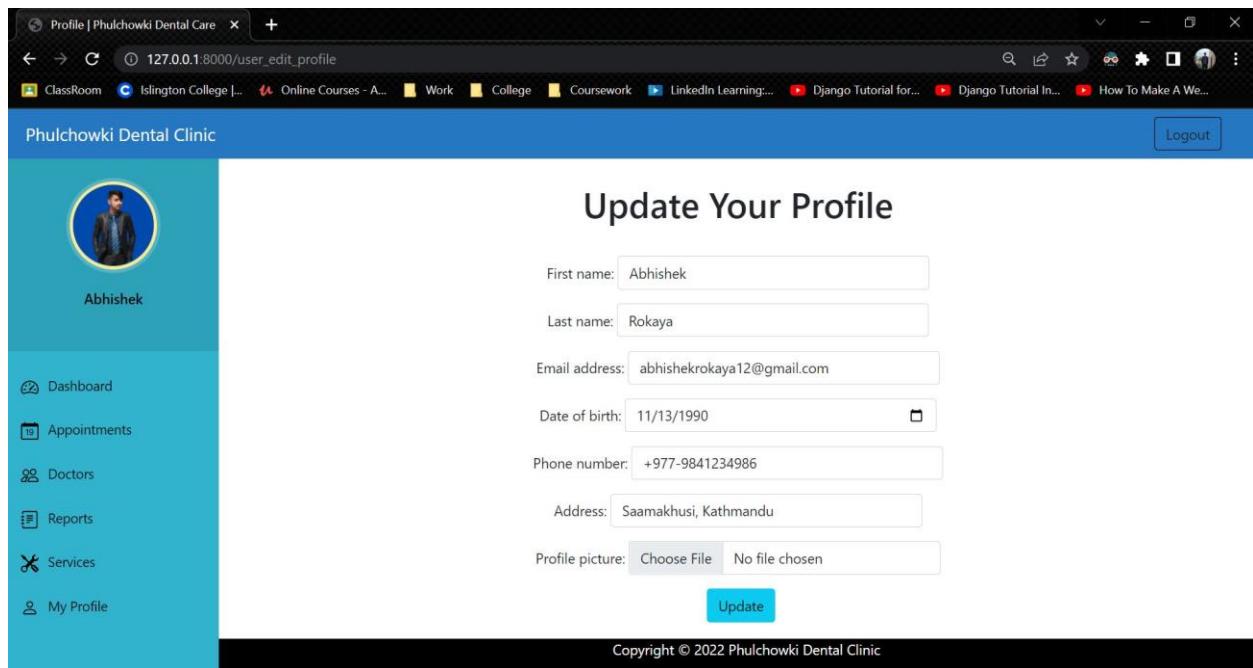


Figure 101: Integration Test 1 - 2

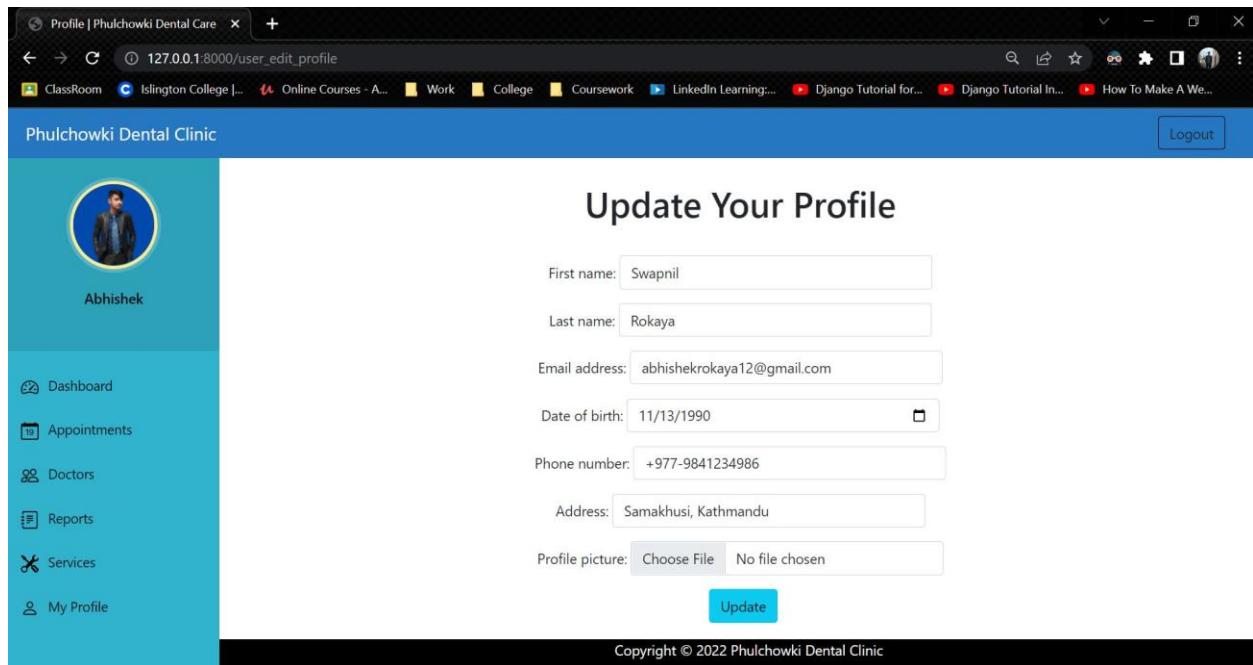


Figure 102: Integration Test 1 - 3

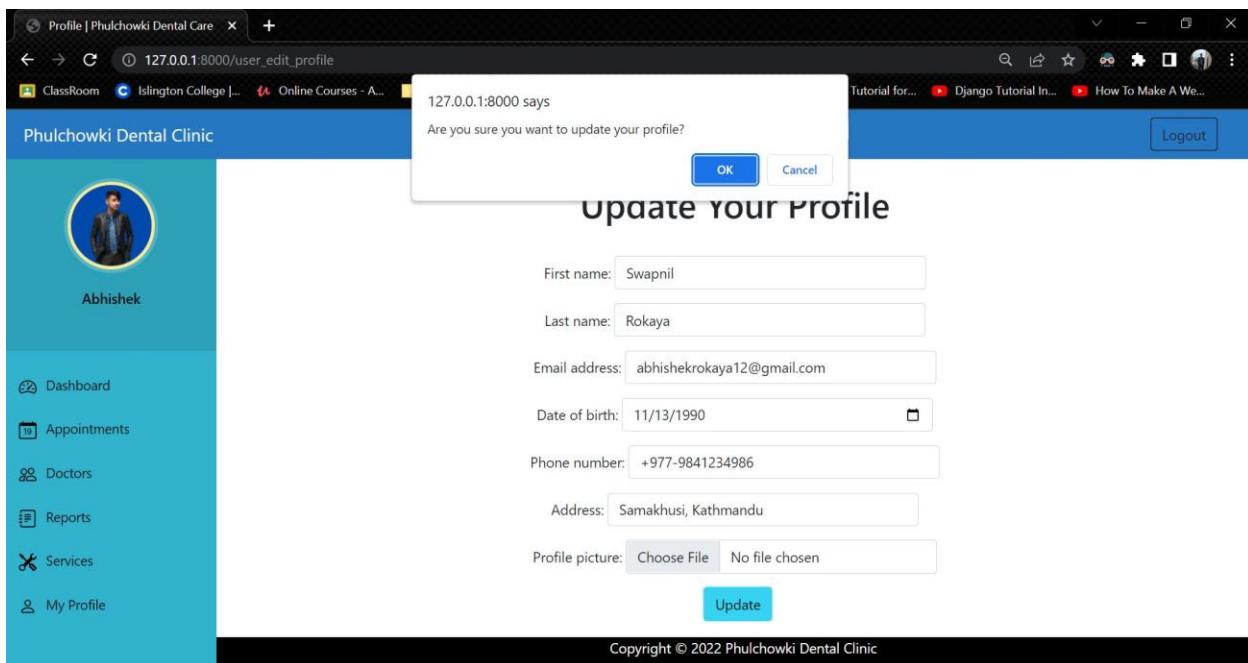


Figure 103: Integration Test 1 – 4

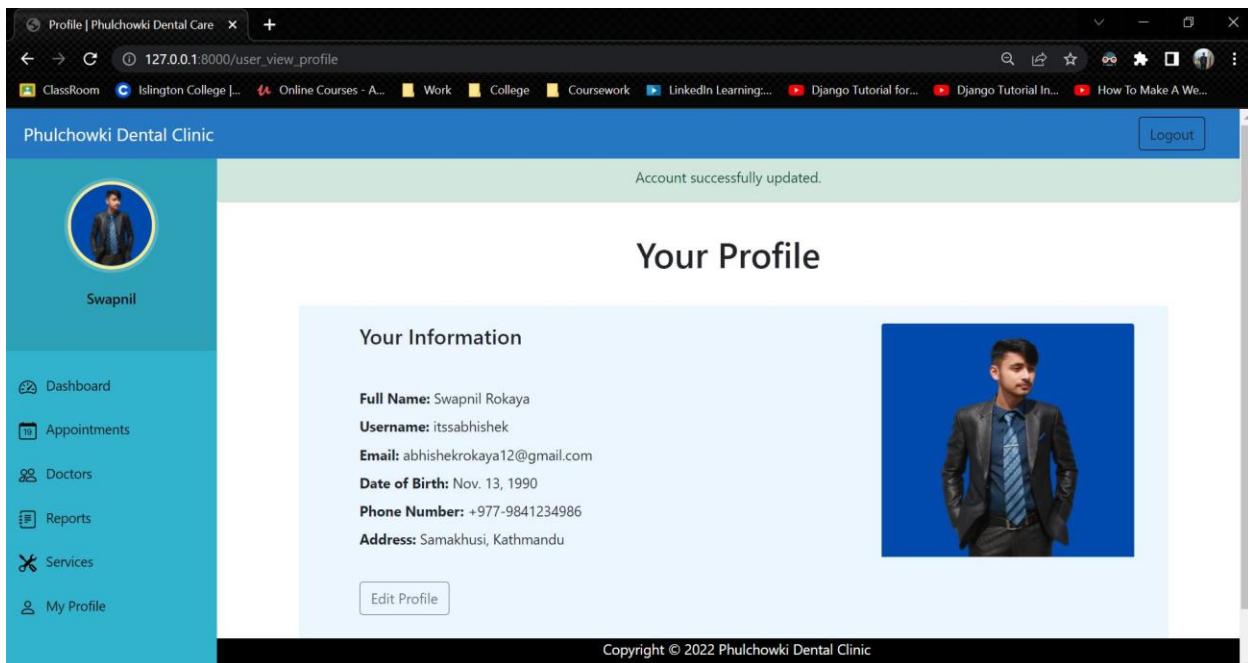


Figure 104: Integration Test 1 - 5

#	id	password	last_login	is_superuser	username	last_name	email	is_staff	is_active	date_joined	first_name
1		pbkdf2_sh...	2022-04-2...	1	admin			1	1	2022-02-0...	
20		pbkdf2_sh...	2022-04-2...	0	itssabhishek	Rokaya	abhishek...	0	1	2022-04-0...	Abhishek
21		pbkdf2_sh...	2022-04-2...	0	leomessi	Messi	leomessi...	0	1	2022-04-0...	Lionel
32		pbkdf2_sh...	2022-04-2...	0	itskushal	Maharjan	kushalmhj...	0	1	2022-04-2...	Kushal

Figure 105: Integration Test 1 - Database Before

#	id	password	last_login	is_superuser	username	last_name	email	is_staff	is_active	date_joined	first_name
1		pbkdf2_sh...	2022-04-2...	1	admin			1	1	2022-02-0...	
20		pbkdf2_sh...	2022-04-2...	0	itssabhishek	Rokaya	abhishek...	0	1	2022-04-0...	Swapnil
21		pbkdf2_sh...	2022-04-2...	0	leomessi	Messi	leomessi...	0	1	2022-04-0...	Lionel
32		pbkdf2_sh...	2022-04-2...	0	itskushal	Maharjan	kushalmhj...	0	1	2022-04-2...	Kushal

Figure 106: Integration Test 1 - Database After

Test No.2	
Objective	To add report by the admin.
Action	The 'Add report' button was clicked, then the necessary information were added and the submit button was clicked.
Expected Output	The report would be added.
Actual Output	The report was added.
Result	The test was successful.

Table 30: Integration Test 2

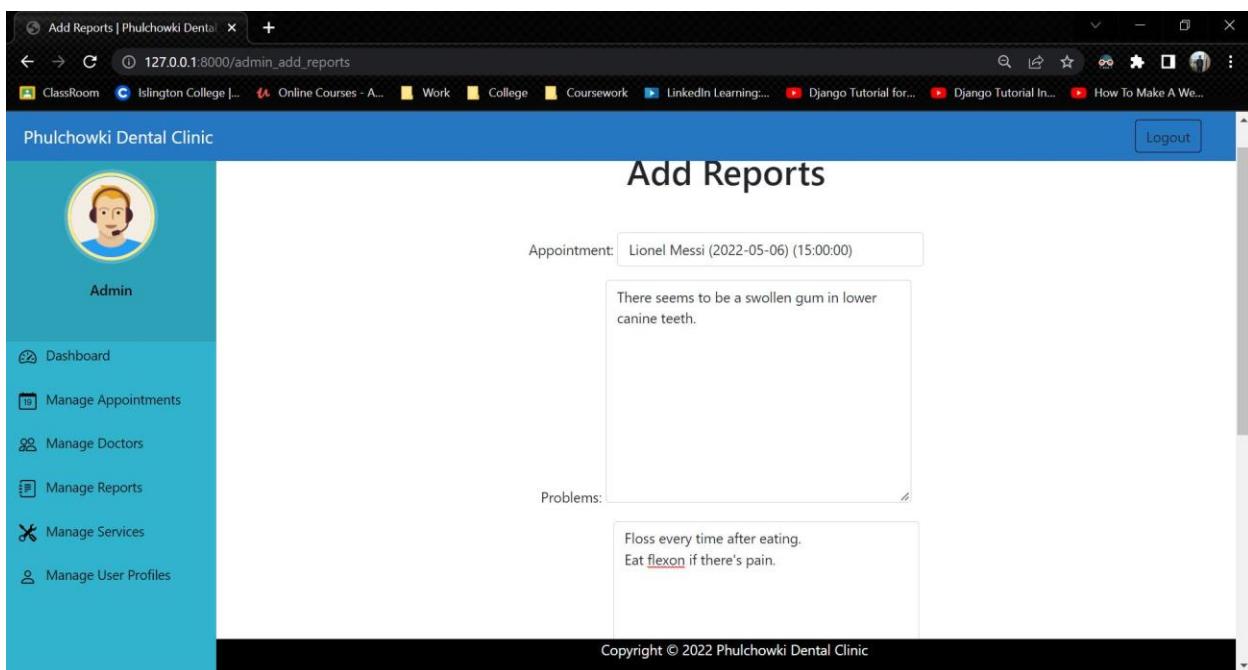


Figure 107: Integration Test 2 - 1

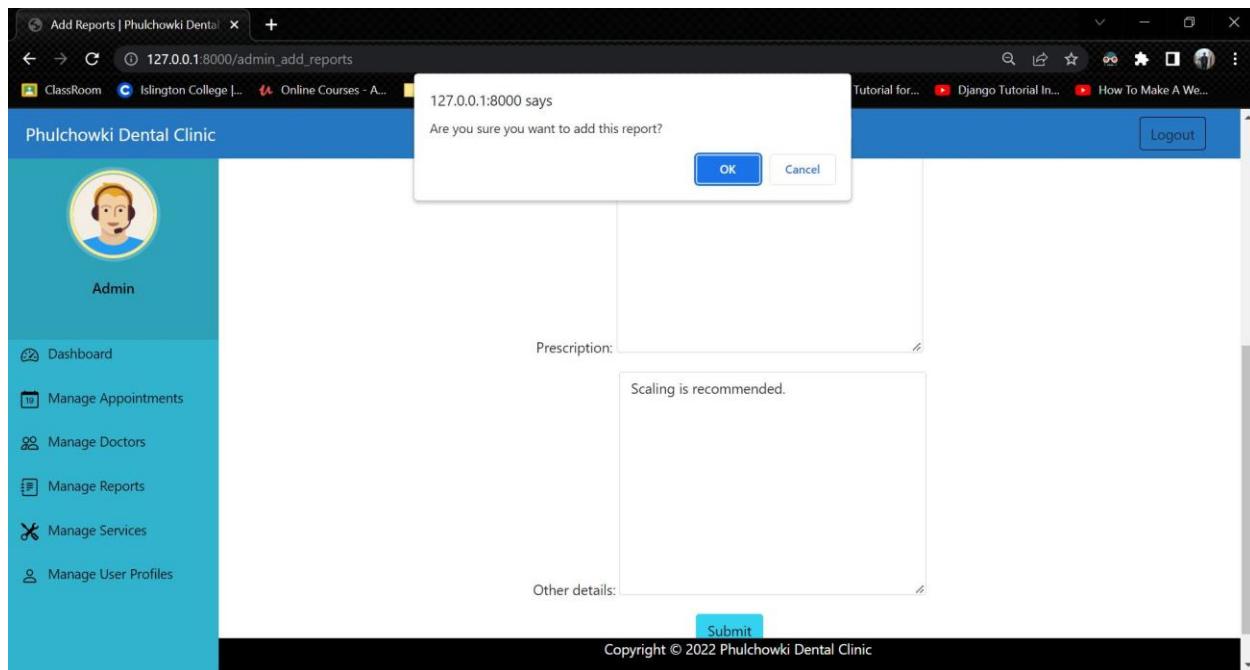


Figure 108: Integration Test 2 – 2

The screenshot shows a Django admin interface for 'Reports' at the URL `127.0.0.1:8000/admin_view_reports`. A success message 'Report added successfully.' is displayed above a table titled 'All Reports'. The table lists three reports with the following data:

Name	Appointment Date	Doctor	Service	Problems	Prescription	Other Details	Total Cost	Update	Delete
Lionel Messi	May 6, 2022	Suresh Magar	General Visit	There seems to be a swollen gum in lower canine teeth.	Floss every time after eating. Eat flexon if there's pain.	Scaling is recommended.	Rs.350		
Swapnil Rokaya	April 29, 2022	Animesh Basnet	General Visit	The teeth seems to be yellowish than normal.	Brush at least 2 times a day.	Scaling is recommended once every six months.	Rs.350		
Swapnil	April 28, 2022	Prasamhsa	Tooth	Wisdom tooth was	Flexon 3-times a	If there is still a lot	Rs.15000		

The left sidebar includes 'Copy', 'Excel', 'CSV', and 'PDF' export buttons. A search bar is also present. The bottom right corner shows the copyright notice 'Copyright © 2022 Phulchowki Dental Clinic'.

Figure 109: Integration Test 2 - 3

<i>i</i>	id	problems	prescription	other_details	appointment_id
	25	Wisdom tooth was causing th...	Flexon 3-times a day after foo...	If there is still a lot of pain, ple...	37
	26	Everything seems to be perfe...	Brush at least 2 times a day.	Scaling is recommended to cl...	41
	27	There is a cavity in the upper r...	The teeth needs to be filled. F...	Avoid drinking cold drinks.	43
	28	The teeth seems to be yellowi...	Brush at least 2 times a day.	Scaling is recommended once...	38

Figure 110: Integration Test 2 - Database Before

<i>i</i>	id	problems	prescription	other_details	appointment_id
	25	Wisdom tooth was causing th...	Flexon 3-times a day after foo...	If there is still a lot of pain, ple...	37
	26	Everything seems to be perfe...	Brush at least 2 times a day.	Scaling is recommended to cl...	41
	27	There is a cavity in the upper r...	The teeth needs to be filled. F...	Avoid drinking cold drinks.	43
	28	The teeth seems to be yellowi...	Brush at least 2 times a day.	Scaling is recommended once...	38
	29	There seems to be a swollen ...	Floss every time after eating. ...	Scaling is recommended.	45

Figure 111: Integration Test 2 - Database After

Test No.3	
Objective	To update the report details by the admin.
Action	The 'Update' button was clicked, then the necessary information were updated and the 'submit' button was clicked.
Expected Output	The report would be updated.
Actual Output	The report was updated.
Result	The test was successful.

Table 31: Integration Test 3

Name	Appointment Date	Doctor	Service	Problems	Prescription	Other Details	Total Cost	Update	Delete
Lionel Messi	May 6, 2022	Suresh Magar	General Visit	There seems to be a swollen gum in lower canine teeth.	Floss every time after eating. Eat flexon if there's pain.	Scaling is recommended.	Rs.350	<input checked="" type="button"/>	<input type="button"/>
Swapnil Rokaya	April 29, 2022	Animesh Basnet	General Visit	The teeth seems to be yellowish than normal.	Brush at least 2 times a day.	Scaling is recommended once every six months.	Rs.350	<input checked="" type="button"/>	<input type="button"/>
Swapnil Rokaya	April 28, 2022	Prasamhsa Khadka	Tooth Extraction	Wisdom tooth was causing the patient a lot of pain.	Flexon 3-times a day after food. Wisgel- 2 times a	If there is still a lot of pain, please contact us as soon	Rs.15000	<input checked="" type="button"/>	<input type="button"/>

Figure 112: Integration Test 3 - 1

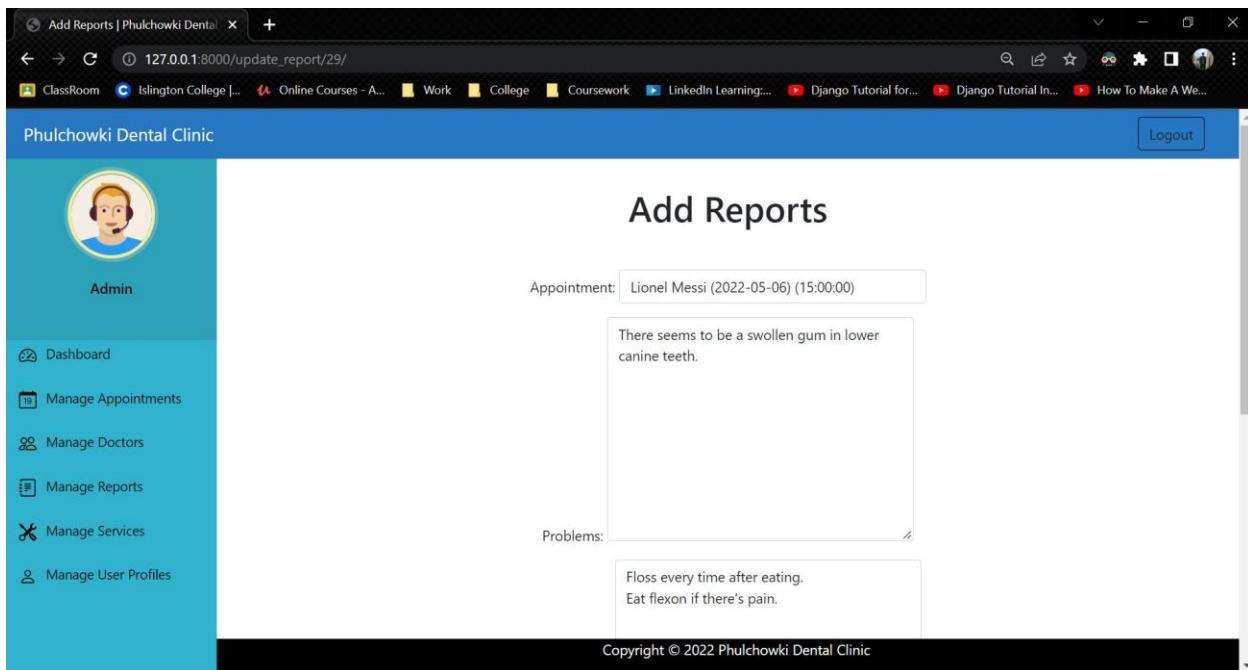


Figure 113: Integration Test 3 – 2

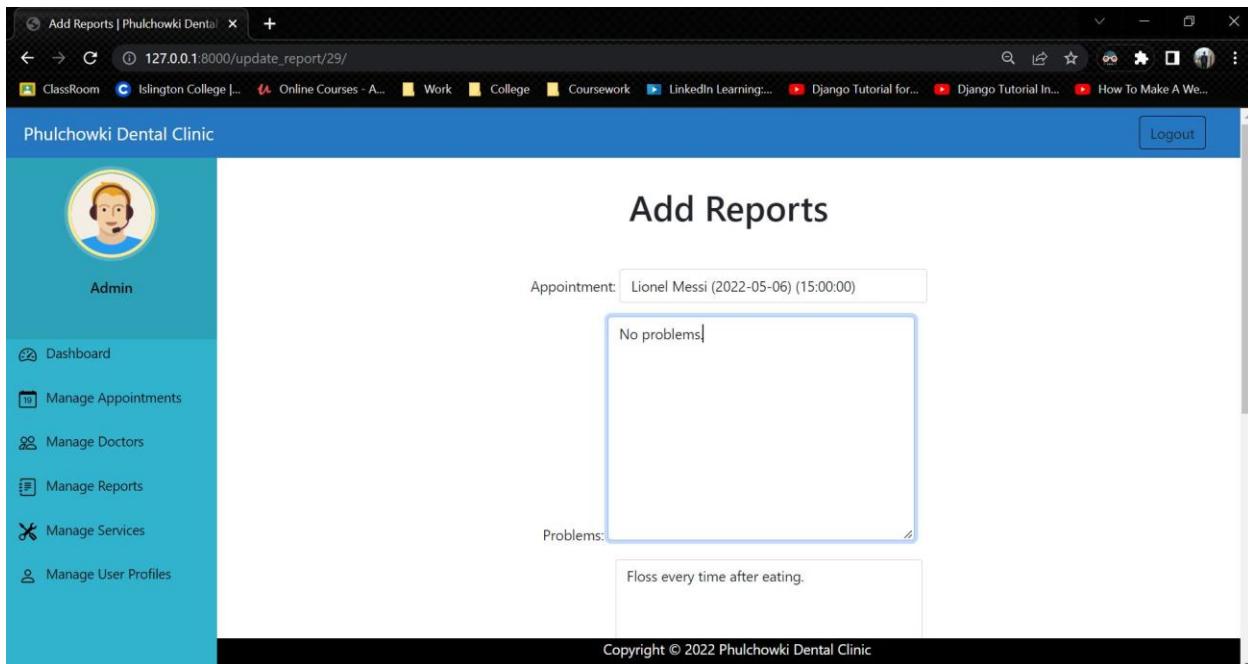


Figure 114: Integration Test 3 – 3

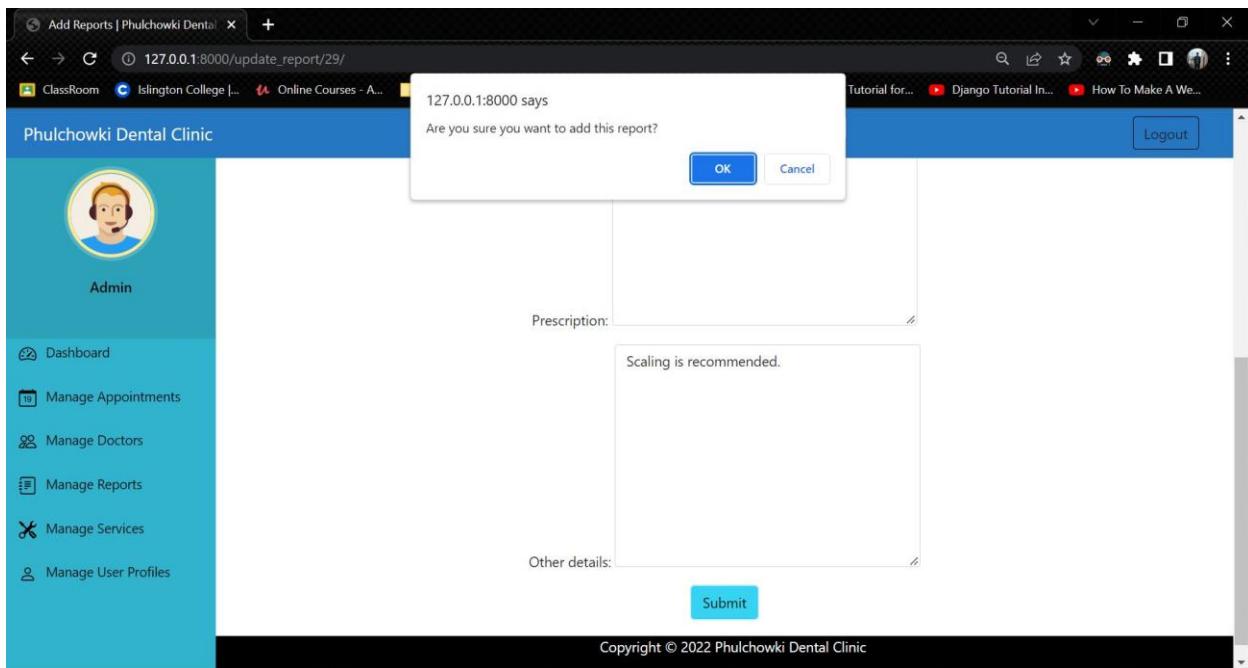


Figure 115: Integration Test 3 – 4

All Reports									
Name	Appointment Date	Doctor	Service	Problems	Prescription	Other Details	Total Cost	Update	Delete
Lionel Messi	May 6, 2022	Suresh Magar	General Visit	No problems.	Floss every time after eating.	Scaling is recommended.	Rs.350		
Swapnil Rokaya	April 29, 2022	Animesh Basnet	General Visit	The teeth seems to be yellowish than normal.	Brush at least 2 times a day.	Scaling is recommended once every six months.	Rs.350		
Swapnil Rokaya	April 28, 2022	Prasamhsa Khadka	Tooth Extraction	Wisdom tooth was causing the patient a lot of pain.	Flexon 3-times a day after food. Wisgel- 2 times a	If there is still a lot of pain, please contact us as soon	Rs.15000		

Figure 116: Integration Test 3 – 5

id	problems	prescription	other_details	appointment_id
25	Wisdom tooth was causing th...	Flexon 3-times a day after foo...	If there is still a lot of pain, ple...	37
26	Everything seems to be perfe...	Brush at least 2 times a day.	Scaling is recommended to cl...	41
27	There is a cavity in the upper r...	The teeth needs to be filled. F...	Avoid drinking cold drinks.	43
28	The teeth seems to be yellowi...	Brush at least 2 times a day.	Scaling is recommended once...	38
29	There seems to be a swollen ...	Floss every time after eating. ...	Scaling is recommended.	45

Figure 117: Integration Test 3 - Database Before

id	problems	prescription	other_details	appointment_id
25	Wisdom tooth was causing th...	Flexon 3-times a day after foo...	If there is still a lot of pain, ple...	37
26	Everything seems to be perfe...	Brush at least 2 times a day.	Scaling is recommended to cl...	41
27	There is a cavity in the upper r...	The teeth needs to be filled. F...	Avoid drinking cold drinks.	43
28	The teeth seems to be yellowi...	Brush at least 2 times a day.	Scaling is recommended once...	38
29	No problems.	Floss every time after eating.	Scaling is recommended.	45

Figure 118: Integration Test 3 - Database After

Test No.4	
Objective	To delete the report by the admin.
Action	The 'Delete' button was clicked, then the confirm button was clicked.
Expected Output	The report would be deleted.
Actual Output	The report was deleted.
Result	The test was successful.

Table 32: Integration Test 4

Name	Appointment Date	Doctor	Service	Problems	Prescription	Other Details	Total Cost	Update	Delete
Lionel Messi	May 6, 2022	Suresh Magar	General Visit	No problems.	Floss every time after eating.	Scaling is recommended.	Rs.350		
Swapnil Rokaya	April 29, 2022	Animesh Basnet	General Visit	The teeth seems to be yellowish than normal.	Brush at least 2 times a day.	Scaling is recommended once every six months.	Rs.350		
Swapnil Rokaya	April 28, 2022	Prasamhsa Khadka	Tooth Extraction	Wisdom tooth was causing the patient a lot of pain.	Flexon 3-times a day after food. Wisgel- 2 times a day while brushing.	If there is still a lot of pain, please contact us as soon as possible.	Rs.15000		
Lionel	April 27, 2022	Animesh	General	There is a cavity in	The teeth needs to	Avoid drinking cold	Rs.250		

Figure 119: Integration Test 4 – 1

The screenshot shows a web browser window for 'Reports | Phulchowki Dental Care' at the URL 127.0.0.1:8000/admin_view_reports. A modal dialog box is centered over the page, displaying the message '127.0.0.1:8000 says Are you sure you want to delete this Report?'. Below the dialog, there are 'OK' and 'Cancel' buttons. The main content area is titled 'All Reports' and contains a table with several rows of patient appointment data. The table columns include Name, Appointment Date, Doctor, Service, Problems, Prescription, Other Details, Total Cost, Update, and Delete. The first row shows Lionel Messi with a general visit and no problems. The second row shows Swapnil Rokaya with a general visit and yellowish teeth. The third row shows Swapnil Rokaya with a tooth extraction and wisdom tooth issues. The fourth row shows Lionel Messi again with a general visit and a cavity. At the bottom of the table, there is a copyright notice: 'Copyright © 2022 Phulchowki Dental Clinic'.

Name	Appointment Date	Doctor	Service	Problems	Prescription	Other Details	Total Cost	Update	Delete
Lionel Messi	May 6, 2022	Suresh Magar	General Visit	No problems.	Floss every time after eating.	Scaling is recommended.	Rs.350		
Swapnil Rokaya	April 29, 2022	Animesh Basnet	General Visit	The teeth seems to be yellowish than normal.	Brush at least 2 times a day.	Scaling is recommended once every six months.	Rs.350		
Swapnil Rokaya	April 28, 2022	Prasamhsa Khadka	Tooth Extraction	Wisdom tooth was causing the patient a lot of pain.	Flexon 3-times a day after food. Wisgel- 2 times a day while brushing.	If there is still a lot of pain, please contact us as soon as possible.	Rs.15000		
Lionel Messi	April 27, 2022	Animesh Basnet	General Visit	There is a cavity in the teeth needs to be treated.	The teeth needs to be treated.	Avoid drinking cold.	Rs.350		

Figure 120: Integration Test 4 – 2

This screenshot shows the same web application interface as Figure 120, but with a success message 'Report deleted.' displayed prominently above the 'All Reports' table. The rest of the page, including the sidebar menu and the table of patient appointments, remains identical to the previous figure.

Figure 121: Integration Test 4 - 3

<code>i id</code>	<code>problems</code>	<code>prescription</code>	<code>other_details</code>	<code>appointment_id</code>
25	Wisdom tooth was causing th...	Flexon 3-times a day after foo...	If there is still a lot of pain, ple...	37
26	Everything seems to be perfe...	Brush at least 2 times a day.	Scaling is recommended to cl...	41
27	There is a cavity in the upper r...	The teeth needs to be filled. F...	Avoid drinking cold drinks.	43
28	The teeth seems to be yellowi...	Brush at least 2 times a day.	Scaling is recommended once...	38
29	No problems.	Floss every time after eating.	Scaling is recommended.	45

Figure 122: Integration Test 4 - Database Before

<code>i id</code>	<code>problems</code>	<code>prescription</code>	<code>other_details</code>	<code>appointment_id</code>
25	Wisdom tooth was causing th...	Flexon 3-times a day after foo...	If there is still a lot of pain, ple...	37
26	Everything seems to be perfe...	Brush at least 2 times a day.	Scaling is recommended to cl...	41
27	There is a cavity in the upper r...	The teeth needs to be filled. F...	Avoid drinking cold drinks.	43
28	The teeth seems to be yellowi...	Brush at least 2 times a day.	Scaling is recommended once...	38

Figure 123: Integration Test 4 - Database After

Test No.5	
Objective	To add service by the admin.
Action	The 'Add Service' button was clicked, then the necessary information were added and the submit button was clicked.
Expected Output	The service would be added.
Actual Output	The service was added.
Result	The test was successful.

Table 33: Integration Test 5

The screenshot shows a web browser window with the URL `127.0.0.1:8000/admin_add_services`. The page title is "Add Services". On the left, there is a sidebar with a user icon and the role "Admin". Below the sidebar, there are several navigation links: "Dashboard", "Manage Appointments", "Manage Doctors", "Manage Reports", "Manage Services" (which is currently selected and highlighted in blue), and "Manage User Profiles". The main content area has input fields for "Service" (containing "Tooth Filling") and "Cost" (containing "3500"). Below these fields is a "Description" box with the text: "It is a procedure to fill the tooth that has been damaged by cavity." At the bottom of the form is a "Submit" button. A copyright notice at the bottom of the page reads "Copyright © 2022 Phulchowki Dental Clinic".

Figure 124: Integration Test 5 - 1

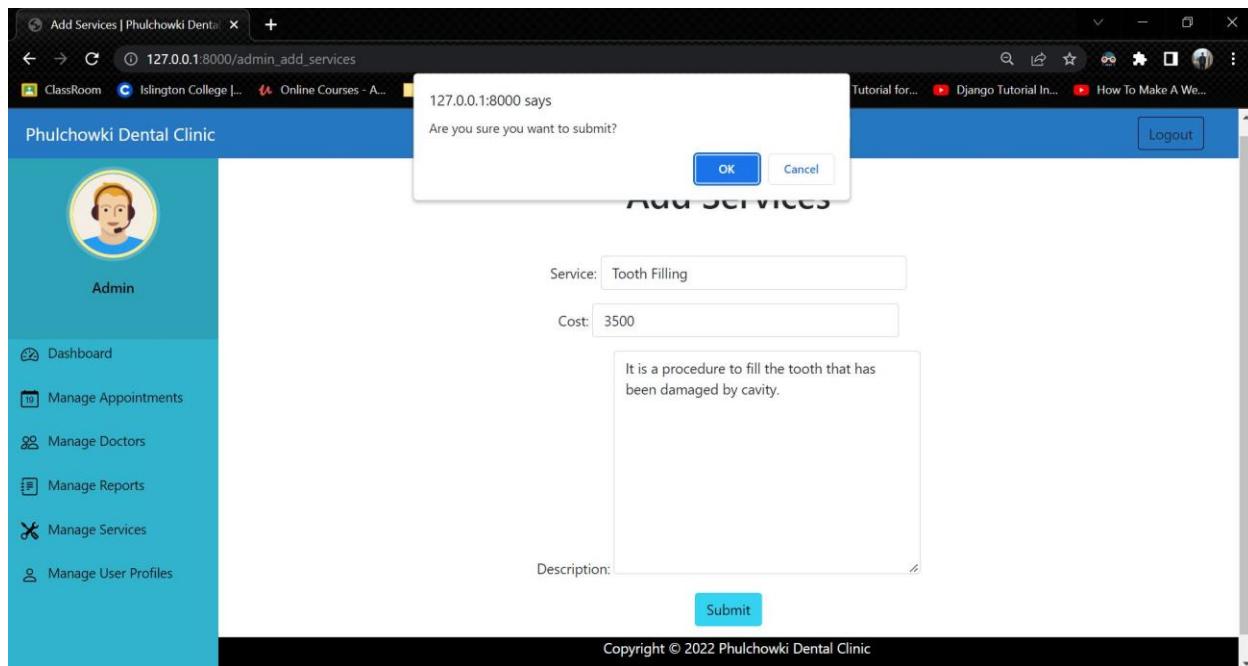


Figure 125: Integration Test 5 - 2

The screenshot shows the 'All Services' page in the Django admin interface for 'Phulchowki Dental Clinic'. The sidebar is identical to Figure 125. The main content area has a title 'All Services'. Below it is a table with the following data:

Name	Cost	Description	Update	Delete
Tooth Filling	Rs.3500	It is a procedure to fill the tooth that has been damaged by cavity.	<input checked="" type="checkbox"/>	<input type="button" value="Delete"/>
RCT	Rs.4500	Root Canal Treatment (RCT) is a procedure done when the tooth decay reaches the pulp (innermost layer of the tooth) causing inflammation.	<input checked="" type="checkbox"/>	<input type="button" value="Delete"/>
Scaling	Rs.1500	Scaling is the cleaning of the teeth.	<input checked="" type="checkbox"/>	<input type="button" value="Delete"/>
Tooth Extraction	Rs.15000	Tooth extraction is done when the teeth needs to be removed. The costs started from 15000 and can be raised according to the complexity.	<input checked="" type="checkbox"/>	<input type="button" value="Delete"/>
General Visit	Rs.350	We do a thorough check up on this and recommend necessary treatments.	<input checked="" type="checkbox"/>	<input type="button" value="Delete"/>

A green bar at the top of the table area says 'Service added successfully.' The URL in the browser is 127.0.0.1:8000/admin/view_services.

Figure 126: Integration Test 5 - 3

<i>i</i>	id	service_name	cost	description
	2	Scaling	1500	Scaling is the cleaning of the teeth.
	3	Tooth Extraction	15000	Tooth extraction is done when the teeth nee...
	4	RCT	4500	Root Canal Treatment (RCT) is a procedure ...
	5	General Visit	350	We do a thorough check up on this and reco...
	6	Braces	28000	We do all kinds of braces implantation to ali...

Figure 127: Integration Test 5 - Database Before

<i>i</i>	id	service_name	cost	description
	2	Scaling	1500	Scaling is the cleaning of the teeth.
	3	Tooth Extraction	15000	Tooth extraction is done when the teeth nee...
	4	RCT	4500	Root Canal Treatment (RCT) is a procedure ..
	5	General Visit	350	We do a thorough check up on this and reco...
	6	Braces	28000	We do all kinds of braces implantation to ali...
	7	Tooth Filling	3500	It is a procedure to fill the tooth that has bee...

Figure 128: Integration Test 5 - Database After

Test No.6	
Objective	To update the service details by the admin.
Action	The 'Update' button was clicked, then the necessary information were updated and the 'submit' button was clicked.
Expected Output	The service would be updated.
Actual Output	The service was updated.
Result	The test was successful.

Table 34: Integration Test 6

Name	Cost	Description	Update	Delete
Tooth Filling	Rs.3500	It is a procedure to fill the tooth that has been damaged by cavity.	<input checked="" type="checkbox"/>	<input type="button" value="Delete"/>
RCT	Rs.4500	Root Canal Treatment (RCT) is a procedure done when the tooth decay reaches the pulp (innermost layer of the tooth) causing inflammation.	<input checked="" type="checkbox"/>	<input type="button" value="Delete"/>
Scaling	Rs.1500	Scaling is the cleaning of the teeth.	<input checked="" type="checkbox"/>	<input type="button" value="Delete"/>
Tooth Extraction	Rs.15000	Tooth extraction is done when the teeth needs to be removed. The costs started from 15000 and can be raised according to the complexity.	<input checked="" type="checkbox"/>	<input type="button" value="Delete"/>
General Visit	Rs.350	We do a thorough check up on this and recommend necessary treatments.	<input checked="" type="checkbox"/>	<input type="button" value="Delete"/>
Braces	Rs.28000	We do all kinds of braces implantation to align your teeth. The base price starts from Rs.28000.	<input checked="" type="checkbox"/>	<input type="button" value="Delete"/>

Figure 129: Integration Test 6 - 1

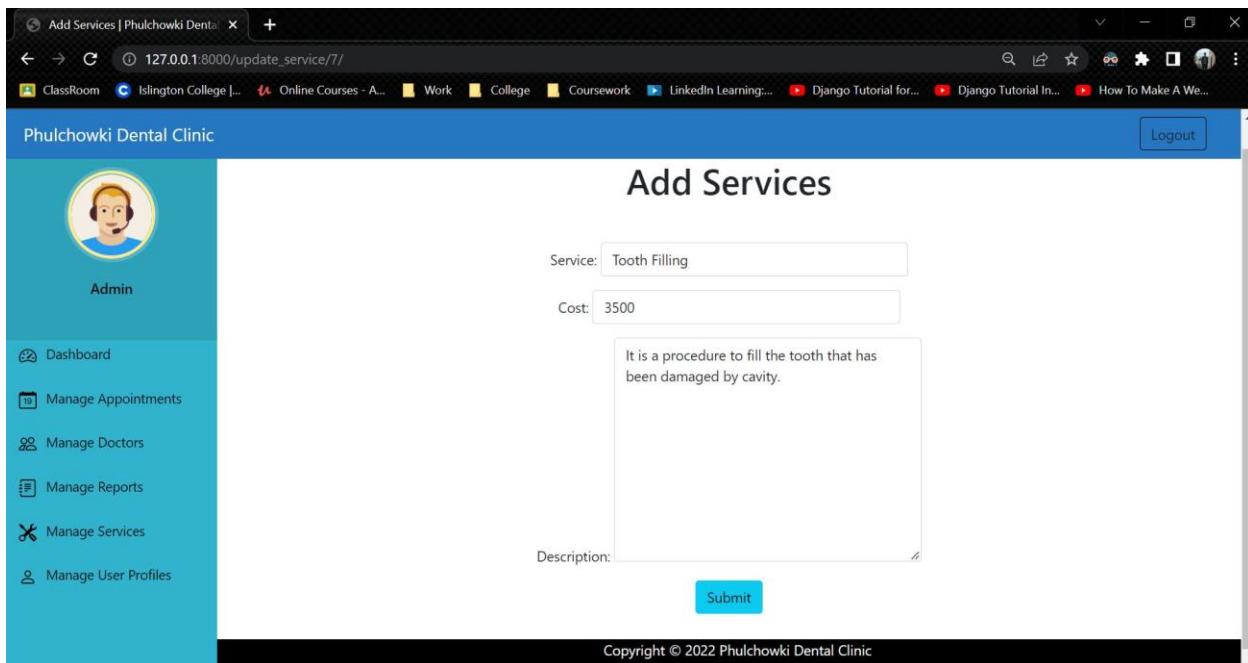


Figure 130: Integration Test 6 – 2

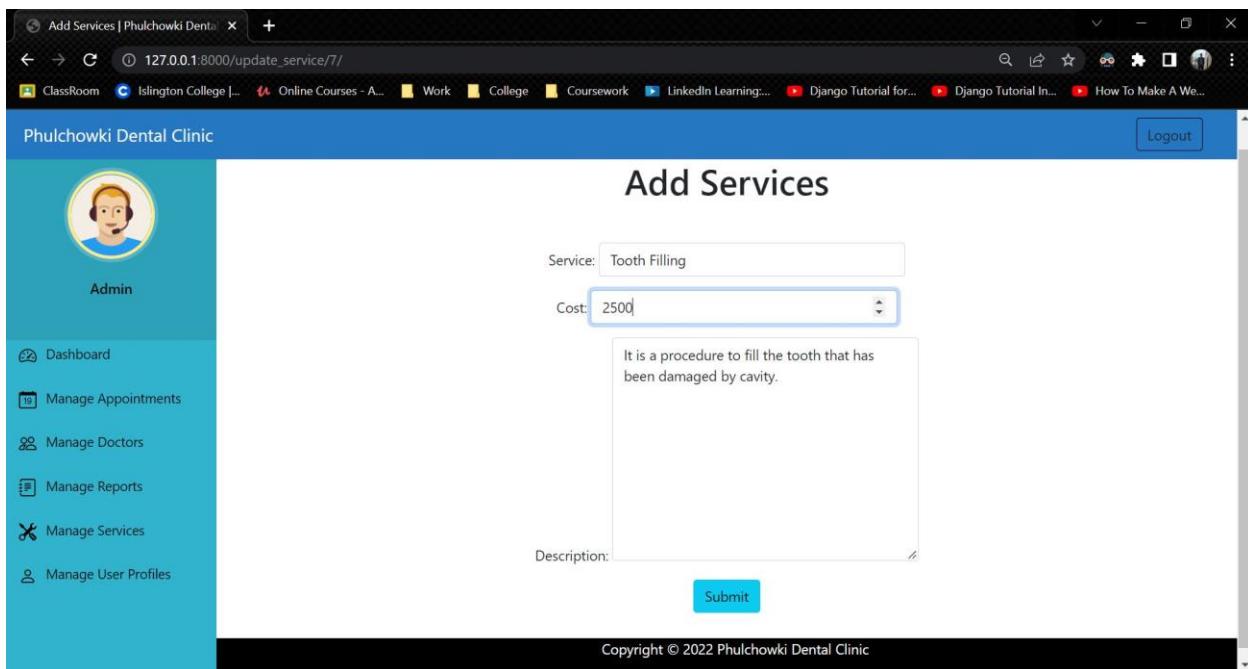


Figure 131: Integration Test 6 - 3

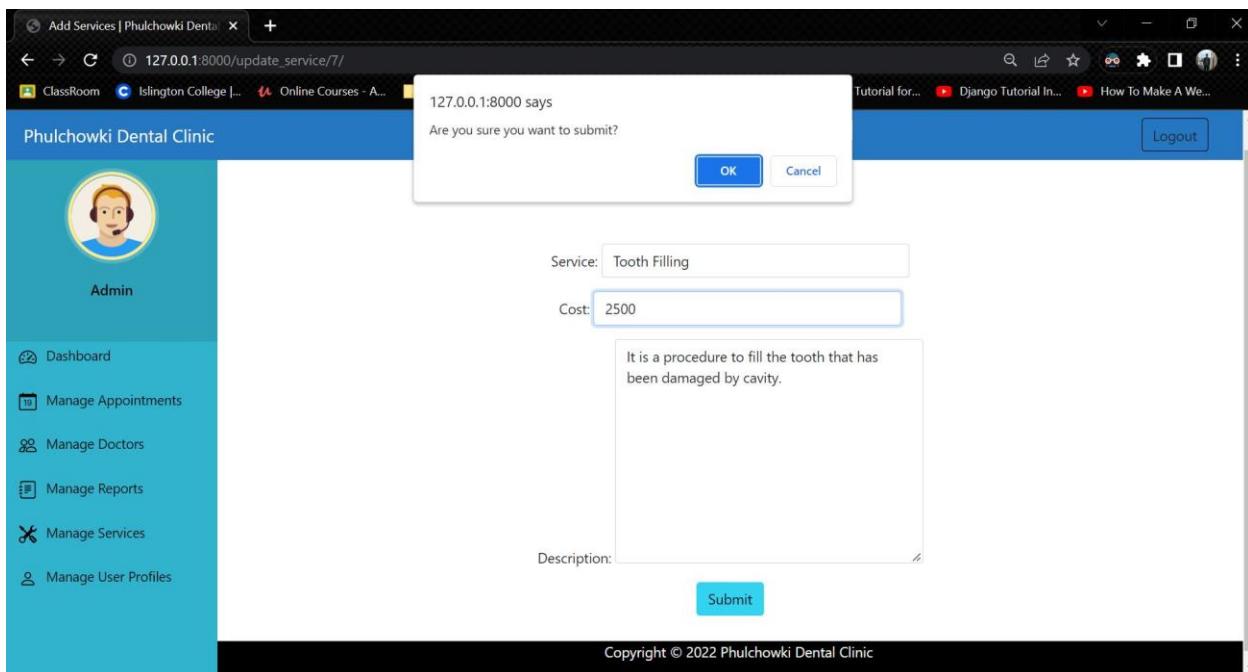


Figure 132: Integration Test 6 - 4

The screenshot shows a web browser window with the URL `127.0.0.1:8000/admin_view_services`. The page title is "All Services". A success message "Service updated." is displayed at the top. Below it is a table listing services:

Name	Cost	Description	Update	Delete
Tooth Filling	Rs.2500	It is a procedure to fill the tooth that has been damaged by cavity.	<input type="checkbox"/>	<input type="checkbox"/>
RCT	Rs.4500	Root Canal Treatment (RCT) is a procedure done when the tooth decay reaches the pulp (innermost layer of the tooth) causing inflammation.	<input type="checkbox"/>	<input type="checkbox"/>
Scaling	Rs.1500	Scaling is the cleaning of the teeth.	<input type="checkbox"/>	<input type="checkbox"/>
Tooth Extraction	Rs.15000	Tooth extraction is done when the teeth needs to be removed. The costs started from 15000 and can be raised according to the complexity.	<input type="checkbox"/>	<input type="checkbox"/>
General Visit	Rs.350	We do a thorough check up on this and recommend necessary treatments.	<input type="checkbox"/>	<input type="checkbox"/>

At the bottom of the page, a copyright notice reads "Copyright © 2022 Phulchowki Dental Clinic".

Figure 133: Integration Test 6 - 5

<code>i</code>	<code>id</code>	<code>service_name</code>	<code>cost</code>	<code>description</code>
2	2	Scaling	1500	Scaling is the cleaning of the teeth.
3	3	Tooth Extraction	15000	Tooth extraction is done when the teeth nee...
4	4	RCT	4500	Root Canal Treatment (RCT) is a procedure ..
5	5	General Visit	350	We do a thorough check up on this and reco...
6	6	Braces	28000	We do all kinds of braces implantation to ali...
7	7	Tooth Filling	3500	It is a procedure to fill the tooth that has bee...

Figure 134: Integration Test 6 - Database Before

<code>i</code>	<code>id</code>	<code>service_name</code>	<code>cost</code>	<code>description</code>
2	2	Scaling	1500	Scaling is the cleaning of the teeth.
3	3	Tooth Extraction	15000	Tooth extraction is done when the teeth nee...
4	4	RCT	4500	Root Canal Treatment (RCT) is a procedure ...
5	5	General Visit	350	We do a thorough check up on this and reco...
6	6	Braces	28000	We do all kinds of braces implantation to ali...
7	7	Tooth Filling	2500	It is a procedure to fill the tooth that has bee...

Figure 135: Integration Test 6 - Database After

Test No.7	
Objective	To delete the service by the admin.
Action	The 'Delete' button was clicked, then the confirm button was clicked.
Expected Output	The service would be deleted.
Actual Output	The service was deleted.
Result	The test was successful.

Table 35: Integration Test 7

Name	Cost	Description	Update	Delete
Tooth Filling	Rs.2500	It is a procedure to fill the tooth that has been damaged by cavity.	<input checked="" type="checkbox"/>	<input type="button" value="Delete"/>
RCT	Rs.4500	Root Canal Treatment (RCT) is a procedure done when the tooth decay reaches the pulp (innermost layer of the tooth) causing inflammation.	<input checked="" type="checkbox"/>	<input type="button" value="Delete"/>
Scaling	Rs.1500	Scaling is the cleaning of the teeth.	<input checked="" type="checkbox"/>	<input type="button" value="Delete"/>
Tooth Extraction	Rs.15000	Tooth extraction is done when the teeth needs to be removed. The costs started from 15000 and can be raised according to the complexity.	<input checked="" type="checkbox"/>	<input type="button" value="Delete"/>
General Visit	Rs.350	We do a thorough check up on this and recommend necessary treatments.	<input checked="" type="checkbox"/>	<input type="button" value="Delete"/>
Braces	Rs.28000	We do all kinds of braces implantation to align your teeth. The base price starts from Rs.28000.	<input checked="" type="checkbox"/>	<input type="button" value="Delete"/>

Copyright © 2022 Phulchowki Dental Clinic

Figure 136: Integration Test 7 - 1

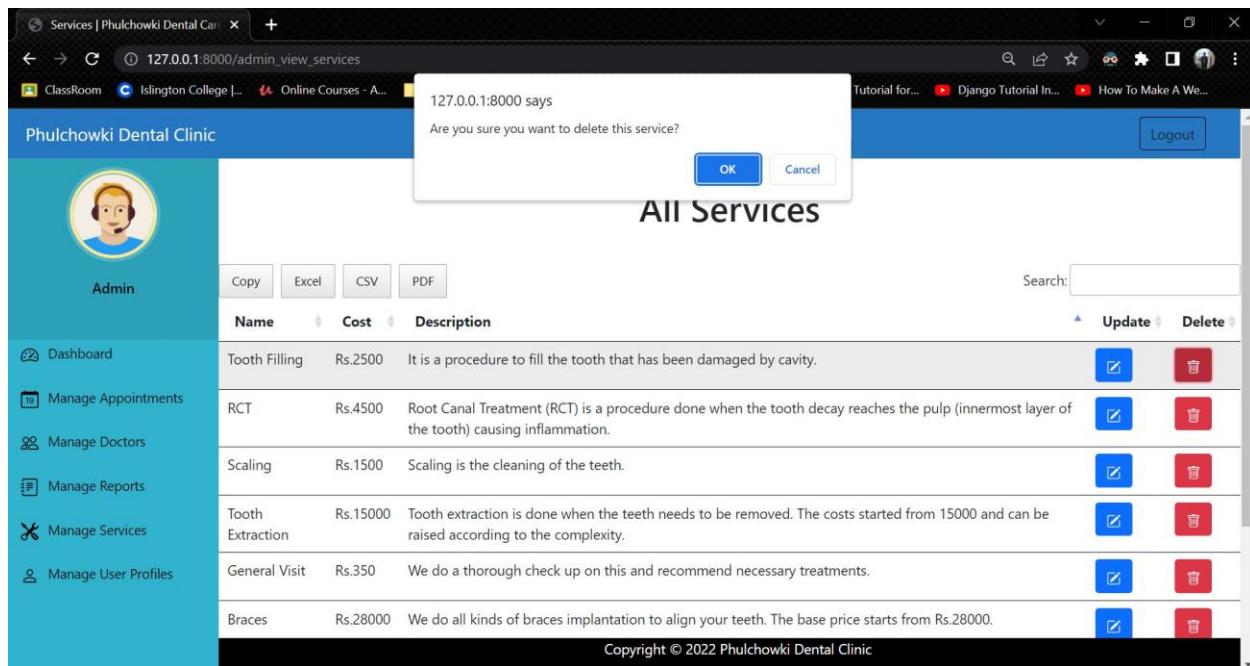


Figure 137: Integration Test 7 – 2

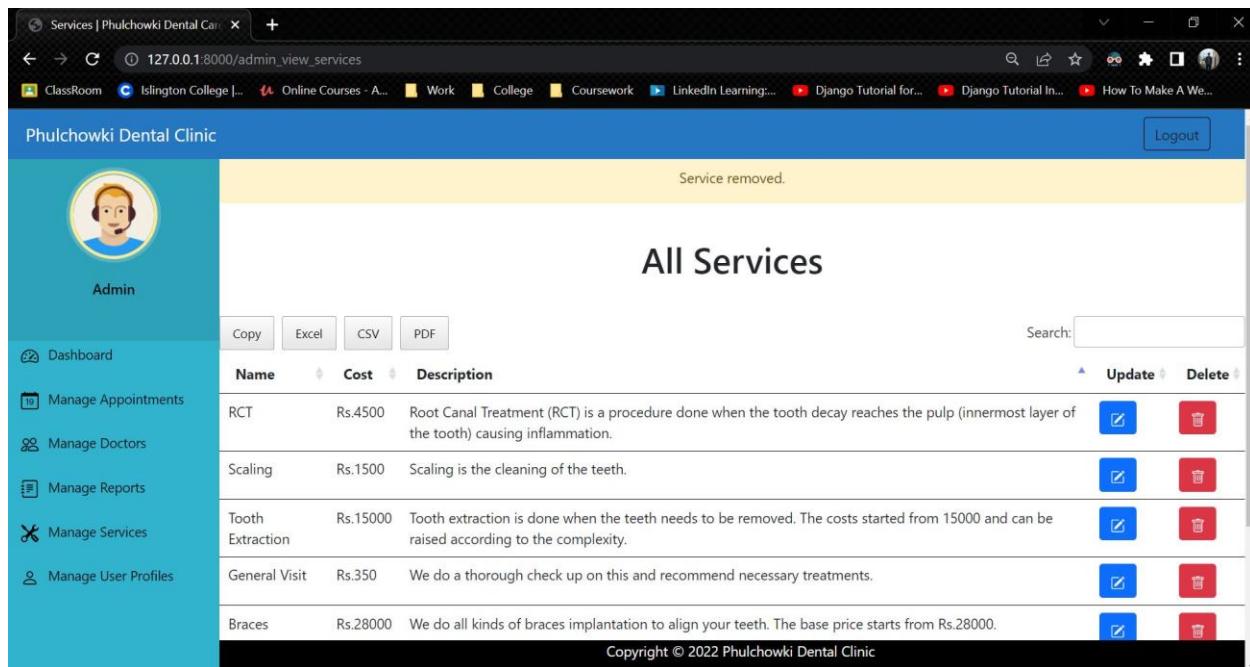


Figure 138: Integration Test 7 – 3

<code>i</code>	<code>id</code>	<code>service_name</code>	<code>cost</code>	<code>description</code>
2		Scaling	1500	Scaling is the cleaning of the teeth.
3		Tooth Extraction	15000	Tooth extraction is done when the teeth nee...
4		RCT	4500	Root Canal Treatment (RCT) is a procedure ...
5		General Visit	350	We do a thorough check up on this and reco...
6		Braces	28000	We do all kinds of braces implantation to ali...
7		Tooth Filling	2500	It is a procedure to fill the tooth that has bee...

Figure 139: Integration Test 7 - Database Before

<code>i</code>	<code>id</code>	<code>service_name</code>	<code>cost</code>	<code>description</code>
2		Scaling	1500	Scaling is the cleaning of the teeth.
3		Tooth Extraction	15000	Tooth extraction is done when the teeth nee...
4		RCT	4500	Root Canal Treatment (RCT) is a procedure ...
5		General Visit	350	We do a thorough check up on this and reco...
6		Braces	28000	We do all kinds of braces implantation to ali...

Figure 140: Integration Test 7 - Database After

Test No.8	
Objective	To add doctor by the admin.
Action	The 'Add Doctor' button was clicked, then the necessary information were added and the submit button was clicked.
Expected Output	The doctor would be added.
Actual Output	The doctor was added.
Result	The test was successful.

Table 36: Integration Test 8

The screenshot shows a web browser window with the URL `127.0.0.1:8000/admin_add_doctors`. The page title is "Add Doctors". On the left, there is a sidebar with a user icon labeled "Admin" and navigation links: "Dashboard", "Manage Appointments", "Manage Doctors", "Manage Reports", "Manage Services", and "Manage User Profiles". The main content area contains fields for entering doctor information:

- Full name: Arun Baniya
- Qualification: Bachelors in Dental Science
- Date of birth: 06/11/1996
- Email: arun.baniya@gmail.com
- Phone number: +977-9813293848
- Address: Samakhosi, Kathmandu
- Profile picture: Choose File arun.JPG
- Experience: 2 years

A "Submit" button is at the bottom right of the form. The footer of the page includes the copyright notice "Copyright © 2022 Phulchowki Dental Clinic".

Figure 141: Integration Test 8 – 1

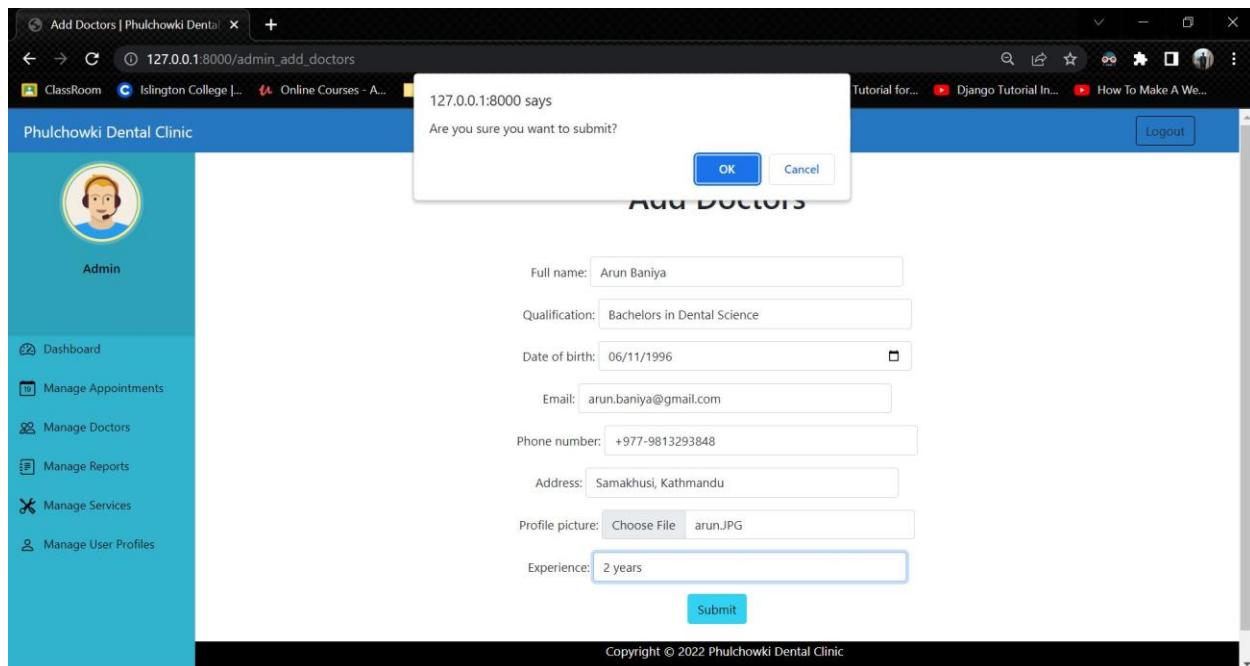


Figure 142: Integration Test 8 - 2

Name	Profile Picture	Qualification	Experience	Date of Birth	Email	Mobile	Address	Update	Delete
Arun Baniya		Bachelors in Dental Science	2 years	June 11, 1996	arun.baniya@gmail.com	+977-9813293848	Samakhosi, Kathmandu		
Suresh Magar		Bachelors in Dental Surgery	1 year	Nov. 20, 1995	sureshbdom@gmail.com	+977-9815283756	Kalanki, Kathmandu		
Deeplab Acharya		Masters in Dental Hygiene	4 years	Jan. 4, 1993	deeplove123@gmail.com	+977-9818273946	Chabahil, Kathmandu		

Figure 143: Integration Test 8 - 3

#	id	full_name	qualification	date_of_birth	email	phone_num...	address	experience	profile_picture
43	Prasamhsa K...	Masters in De...	1992-05-05	prasma12@gm...	+977-986123...	Thaiba, Lalitpur	5 years	profile_pic/doctor_pr...	
44	Animesh Basnet	Bachelors in ...	1998-01-01	animesh.b@g...	+977-984127...	Satdobato, Lal...	3 years	profile_pic/doctor_pr...	
45	Deeplab Acha...	Masters in De...	1993-01-04	deeplove123...	+977-981827...	Chabahil, Kat...	4 years	profile_pic/doctor_pr...	
46	Subikhyat Ghi...	Bachelors in ...	1995-05-30	subi.gh@gmai...	+977-9851117...	Gairidhara, Ka...	2 years	profile_pic/doctor_pr...	
47	Rabina Lama	Masters in De...	1991-01-17	rabina.lama@...	+977-986628...	Lamatar, Lalitpur	6 years	profile_pic/doctor_pr...	
48	Suresh Magar	Bachelors in ...	1995-11-20	sureshbdm@...	+977-981528...	Kalanki, Kath...	1 year	profile_pic/doctor_pr...	

Figure 144: Integration Test 8 - Database before

#	id	full_name	qualification	date_of_birth	email	phone_num...	address	experience	profile_picture
43	Prasamhsa K...	Masters in De...	1992-05-05	prasma12@gm...	+977-986123...	Thaiba, Lalitpur	5 years	profile_pic/doctor_pr...	
44	Animesh Basnet	Bachelors in ...	1998-01-01	animesh.b@g...	+977-984127...	Satdobato, Lal...	3 years	profile_pic/doctor_pr...	
45	Deeplab Acha...	Masters in De...	1993-01-04	deeplove123...	+977-981827...	Chabahil, Kat...	4 years	profile_pic/doctor_pr...	
46	Subikhyat Ghi...	Bachelors in ...	1995-05-30	subi.gh@gmai...	+977-9851117...	Gairidhara, Ka...	2 years	profile_pic/doctor_pr...	
47	Rabina Lama	Masters in De...	1991-01-17	rabina.lama@...	+977-986628...	Lamatar, Lalitpur	6 years	profile_pic/doctor_pr...	
48	Suresh Magar	Bachelors in ...	1995-11-20	sureshbdm@...	+977-981528...	Kalanki, Kath...	1 year	profile_pic/doctor_pr...	
49	Arun Baniya	Bachelors in ...	1996-06-11	arun.baniya@...	+977-981329...	Samakhusi, K...	2 years	profile_pic/doctor_pr...	

Figure 145: Integration Test 8 - Database After

Test No.9	
Objective	To update the doctor's details by the admin.
Action	The 'Update' button was clicked, then the necessary information were updated and the 'submit' button was clicked.
Expected Output	The doctor would be updated.
Actual Output	The doctor was updated.
Result	The test was successful.

Table 37: Integration Test 9

Name	Profile Picture	Qualification	Experience	Date of Birth	Email	Mobile	Address	Update	Delete
Arun Baniya		Bachelors in Dental Science	2 years	June 11, 1996	arun.baniya@gmail.com	+977-9813293848	Samakhusi, Kathmandu		
Suresh Magar		Bachelors in Dental Surgery	1 year	Nov. 20, 1995	sureshbdom@gmail.com	+977-9815283756	Kalanki, Kathmandu		
Deeplab Acharya		Masters in Dental Hygiene	4 years	Jan. 4, 1993	deeplove123@gmail.com	+977-9818273946	Chabahil, Kathmandu		
Animesh		Bachelors in	3 years	Jan. 1,	animesh.b@gmail.com	+977-	Satdobato,		

Figure 146: Integration Test 9 - 1

Add Doctors | Phulchowki Dental | 127.0.0.1:8000/update_doctor/49/

Phulchowki Dental Clinic Logout

Add Doctors

Full name: Arun Baniya

Qualification: Bachelors in Dental Science

Date of birth: 06/11/1996

Email: arun.baniya@gmail.com

Phone number: +977-9813293848

Address: Samakhosi, Kathmandu

Profile picture: Choose File No file chosen

Experience: 2 years

Submit

Copyright © 2022 Phulchowki Dental Clinic

Figure 147: Integration Test 9 – 2

Add Doctors | Phulchowki Dental | 127.0.0.1:8000/update_doctor/49/

Phulchowki Dental Clinic Logout

Add Doctors

Full name: Arun Baniya

Qualification: Bachelors in Dental Science

Date of birth: 06/11/1996

Email: arun.baniya@gmail.com

Phone number: +977-9813293848

Address: Gongabu, Kathmandu

Profile picture: Choose File No file chosen

Experience: 3 years

Submit

Copyright © 2022 Phulchowki Dental Clinic

Figure 148: Integration Test 9 - 3

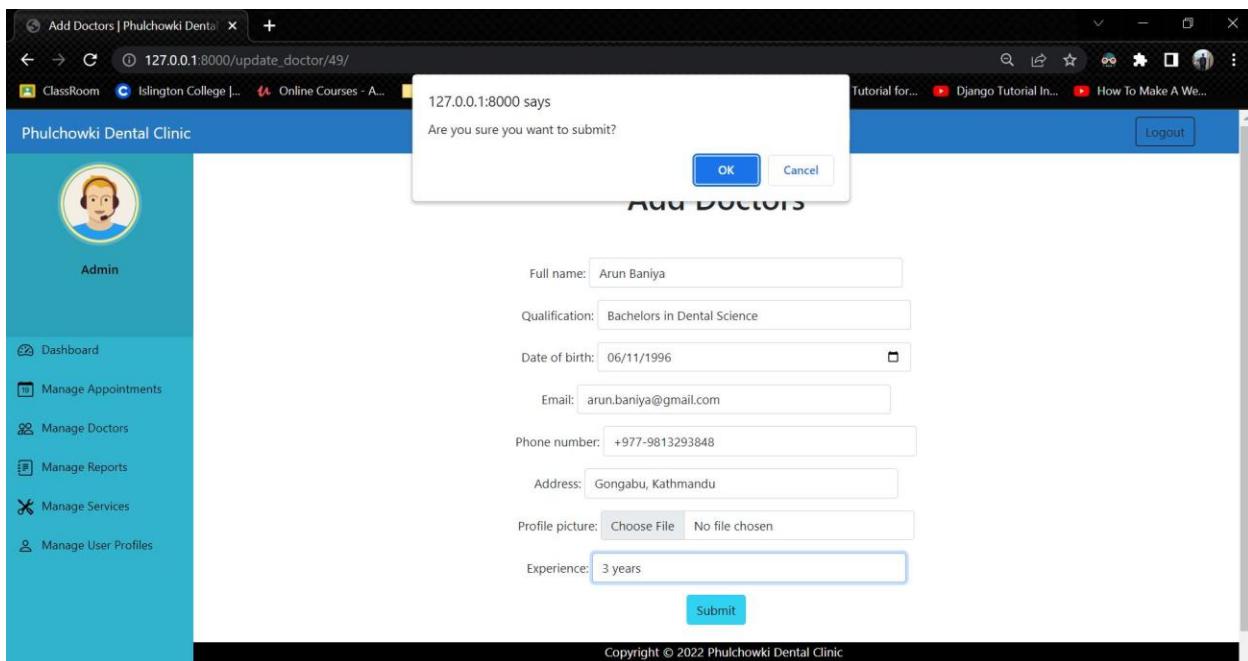


Figure 149: Integration Test 9 - 4

All Doctors									
Name	Profile Picture	Qualification	Experience	Date of Birth	Email	Mobile	Address	Update	Delete
Arun Baniya		Bachelors in Dental Science	3 years	June 11, 1996	arun.baniya@gmail.com	+977-9813293848	Gongabu, Kathmandu		
Suresh Magar		Bachelors in Dental Surgery	1 year	Nov. 20, 1995	sureshbdom@gmail.com	+977-9815283756	Kalanki, Kathmandu		
Deeplab Acharya		Masters in Dental Hygiene	4 years	Jan. 4, 1993	deeplove123@gmail.com	+977-9818273946	Chabahil, Kathmandu		

Figure 150: Integration Test 9 – 5

#	id	full_name	qualification	date_of_birth	email	phone_num...	address	experience	profile_picture
43		Prasamhsa K...	Masters in De...	1992-05-05	prsma12@gm...	+977-986123...	Thaiba, Lalitpur	5 years	profile_pic/doctor_pr...
44		Animesh Basnet	Bachelors in ...	1998-01-01	animesh.b@g...	+977-984127...	Satdobato, Lal...	3 years	profile_pic/doctor_pr...
45		Deeplab Acha...	Masters in De...	1993-01-04	deeplove123...	+977-981827...	Chabahil, Kat...	4 years	profile_pic/doctor_pr...
46		Subikhyat Ghi...	Bachelors in ...	1995-05-30	subi.gh@gmai...	+977-9851117...	Gairidhara, Ka...	2 years	profile_pic/doctor_pr...
47		Rabina Lama	Masters in De...	1991-01-17	rabina.lama@...	+977-986628...	Lamatar, Lalitpur	6 years	profile_pic/doctor_pr...
48		Suresh Magar	Bachelors in ...	1995-11-20	sureshbdm@...	+977-981528...	Kalanki, Kath...	1 year	profile_pic/doctor_pr...
49		Arun Baniya	Bachelors in ...	1996-06-11	arun.baniya@...	+977-981329...	Samakhusi, K...	2 years	profile_pic/doctor_pr...

Figure 151: Integration Test 9 - Database Before

#	id	full_name	qualification	date_of_birth	email	phone_num...	address	experience	profile_picture
43		Prasamhsa K...	Masters in De...	1992-05-05	prsma12@gm...	+977-986123...	Thaiba, Lalitpur	5 years	profile_pic/doctor_pr...
44		Animesh Basnet	Bachelors in ...	1998-01-01	animesh.b@g...	+977-984127...	Satdobato, Lal...	3 years	profile_pic/doctor_pr...
45		Deeplab Acha...	Masters in De...	1993-01-04	deeplove123...	+977-981827...	Chabahil, Kat...	4 years	profile_pic/doctor_pr...
46		Subikhyat Ghi...	Bachelors in ...	1995-05-30	subi.gh@gmai...	+977-9851117...	Gairidhara, Ka...	2 years	profile_pic/doctor_pr...
47		Rabina Lama	Masters in De...	1991-01-17	rabina.lama@...	+977-986628...	Lamatar, Lalitpur	6 years	profile_pic/doctor_pr...
48		Suresh Magar	Bachelors in ...	1995-11-20	sureshbdm@...	+977-981528...	Kalanki, Kath...	1 year	profile_pic/doctor_pr...
49		Arun Baniya	Bachelors in ...	1996-06-11	arun.baniya@...	+977-981329...	Gongabu, Kat...	3 years	profile_pic/doctor_pr...

Figure 152: Integration Test 9 - Database After

Test No.10	
Objective	To delete the doctor by the admin.
Action	The 'Delete' button was clicked, then the confirm button was clicked.
Expected Output	The doctor would be deleted.
Actual Output	The doctor was deleted.
Result	The test was successful.

Table 38: Integration Test 10

Name	Profile Picture	Qualification	Experience	Date of Birth	Email	Mobile	Address	Update	Delete
Arun Baniya		Bachelors in Dental Science	3 years	June 11, 1996	arun.baniya@gmail.com	+977-9813293848	Gongabu, Kathmandu		
Suresh Magar		Bachelors in Dental Surgery	1 year	Nov. 20, 1995	sureshbdom@gmail.com	+977-9815283756	Kalanki, Kathmandu		
Deeplab Acharya		Masters in Dental Hygiene	4 years	Jan. 4, 1993	deeplove123@gmail.com	+977-9818273946	Chabahil, Kathmandu		
Animesh		Bachelors in	3 years	Jan. 1,	animesh.b@gmail.com	+977-	Satdobato,		

Figure 153: Integration Test 10 - 1

The screenshot shows a web-based administration interface for a dental clinic. On the left is a sidebar with a blue header "Admin" and several menu items: Dashboard, Manage Appointments, Manage Doctors, Manage Reports, Manage Services, and Manage User Profiles. The main content area has a title "All Doctors" and a table with the following data:

Name	Profile Picture	Qualification	Experience	Date of Birth	Email	Mobile	Address	Update	Delete
Arun Baniya		Bachelors in Dental Science	3 years	June 11, 1996	arun.baniya@gmail.com	+977-9813293848	Gongabu, Kathmandu		
Suresh Magar		Bachelors in Dental Surgery	1 year	Nov. 20, 1995	sureshbdm@gmail.com	+977-9815283756	Kalanki, Kathmandu		
Deeplab Acharya		Masters in Dental Hygiene	4 years	Jan. 4, 1993	deeplove123@gmail.com	+977-9818273946	Chabahil, Kathmandu		
Animesh		Bachelors in	3 years	Jan. 1,	animesh.b@gmail.com	+977-	Satdobato,		

A modal dialog box is centered over the table, displaying the message "127.0.0.1:8000 says Are you sure you want to delete this Doctor?". It contains two buttons: "OK" and "Cancel".

Figure 154: Integration Test 10 – 2

This screenshot shows the same web application after a doctor has been deleted. The yellow banner at the top center displays the message "Doctor removed.". The main table now lists three doctors: Suresh Magar, Deeplab Acharya, and Animesh Basnet.

Name	Profile Picture	Qualification	Experience	Date of Birth	Email	Mobile	Address	Update	Delete
Suresh Magar		Bachelors in Dental Surgery	1 year	Nov. 20, 1995	sureshbdm@gmail.com	+977-9815283756	Kalanki, Kathmandu		
Deeplab Acharya		Masters in Dental Hygiene	4 years	Jan. 4, 1993	deeplove123@gmail.com	+977-9818273946	Chabahil, Kathmandu		
Animesh Basnet		Bachelors in Dental Science	3 years	Jan. 1, 1998	animesh.b@gmail.com	+977-9841273098	Lalitpur		

Figure 155: Integration Test 10 - 3

#	id	full_name	qualification	date_of_birth	email	phone_num...	address	experience	profile_picture
43	Prasamhsa K...	Masters in De...	1992-05-05	prasma12@gm...	+977-986123...	Thaiba, Lalitpur	5 years	profile_pic/doctor_pr...	
44	Animesh Basnet	Bachelors in ...	1998-01-01	animesh.b@g...	+977-984127...	Satdobato, Lal...	3 years	profile_pic/doctor_pr...	
45	Deeplab Acha...	Masters in De...	1993-01-04	deeplove123...	+977-981827...	Chabahil, Kat...	4 years	profile_pic/doctor_pr...	
46	Subikhyat Ghi...	Bachelors in ...	1995-05-30	subi.gh@gmai...	+977-9851117...	Gairidhara, Ka...	2 years	profile_pic/doctor_pr...	
47	Rabina Lama	Masters in De...	1991-01-17	rabina.lama@...	+977-986628...	Lamatar, Lalitpur	6 years	profile_pic/doctor_pr...	
48	Suresh Magar	Bachelors in ...	1995-11-20	sureshbdm@...	+977-981528...	Kalanki, Kath...	1 year	profile_pic/doctor_pr...	
49	Arun Baniya	Bachelors in ...	1996-06-11	arun.baniya@...	+977-981329...	Gongabu, Kat...	3 years	profile_pic/doctor_pr...	

Figure 156: Integration Test 10 - Database Before

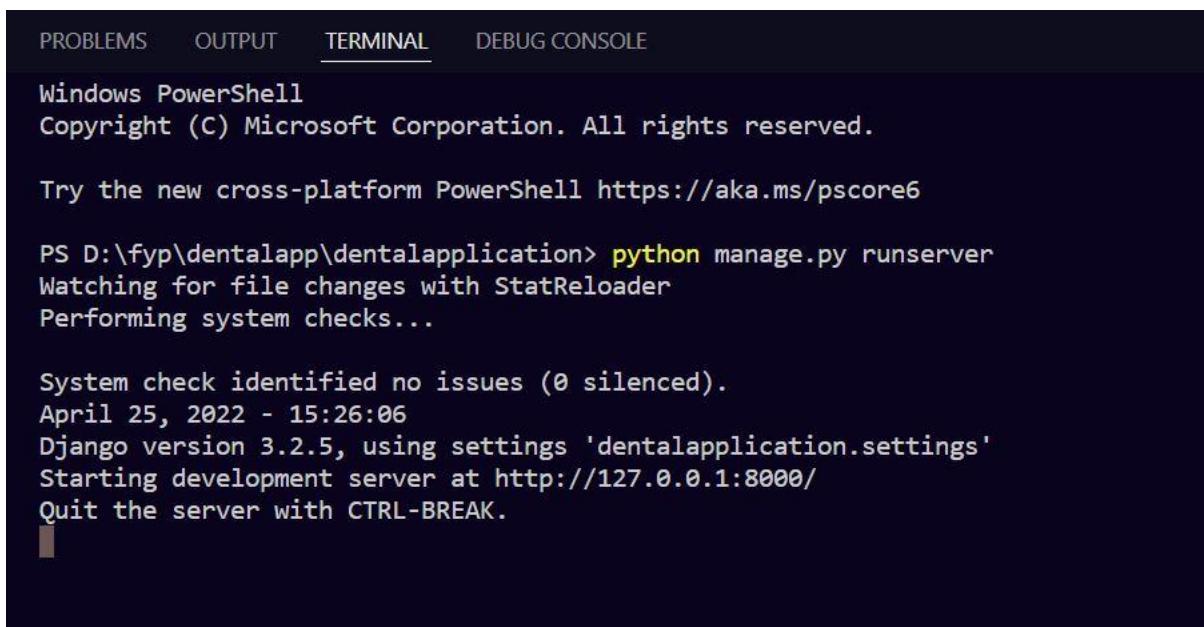
#	id	full_name	qualification	date_of_birth	email	phone_num...	address	experience	profile_picture
43	Prasamhsa K...	Masters in De...	1992-05-05	prasma12@gm...	+977-986123...	Thaiba, Lalitpur	5 years	profile_pic/doctor_pr...	
44	Animesh Basnet	Bachelors in ...	1998-01-01	animesh.b@g...	+977-984127...	Satdobato, Lal...	3 years	profile_pic/doctor_pr...	
45	Deeplab Acha...	Masters in De...	1993-01-04	deeplove123...	+977-981827...	Chabahil, Kat...	4 years	profile_pic/doctor_pr...	
46	Subikhyat Ghi...	Bachelors in ...	1995-05-30	subi.gh@gmai...	+977-9851117...	Gairidhara, Ka...	2 years	profile_pic/doctor_pr...	
47	Rabina Lama	Masters in De...	1991-01-17	rabina.lama@...	+977-986628...	Lamatar, Lalitpur	6 years	profile_pic/doctor_pr...	
48	Suresh Magar	Bachelors in ...	1995-11-20	sureshbdm@...	+977-981528...	Kalanki, Kath...	1 year	profile_pic/doctor_pr...	

Figure 157: Integration Test 10 - Database After

4.5 System Testing

Test No.1	
Objective	To run the system from VS Code.
Action	The command 'python manage.py runserver' was entered in the terminal.
Expected Output	The system would run.
Actual Output	The system ran successfully.
Result	The test was successful.

Table 39: System Testing 1



```

PROBLEMS    OUTPUT    TERMINAL    DEBUG CONSOLE

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\fyp\dentalapp\dentalapplication> python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
April 25, 2022 - 15:26:06
Django version 3.2.5, using settings 'dentalapplication.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.

```

Figure 158: System Testing 1

4.6 Client Evaluation Testing

Full Name

1 response

Prashamsa Khadka

Figure 159: Client Evaluation Testing – 1

Email

1 response

prashamsa.khadka@gmail.com

Figure 160: Client Evaluation Testing – 2

Rate the User Interface of the application

Copy

1 response

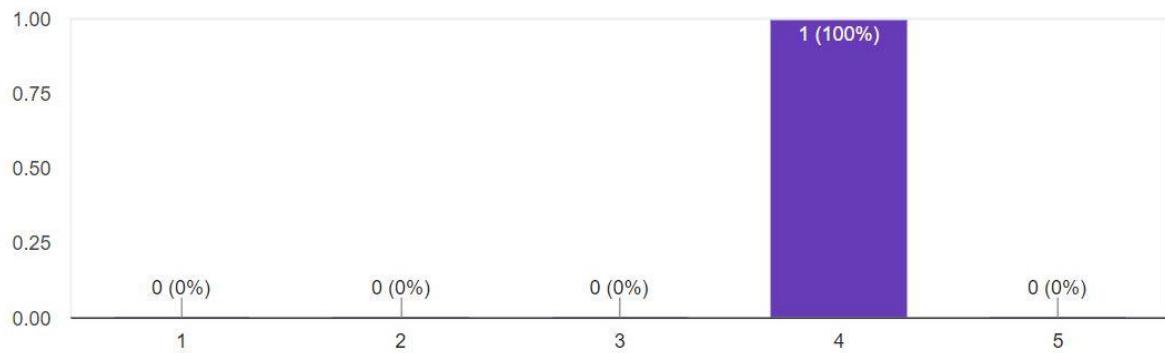


Figure 161: Client Evaluation Testing – 3

How much are you satisfied with the color combinations?

Copy

1 response

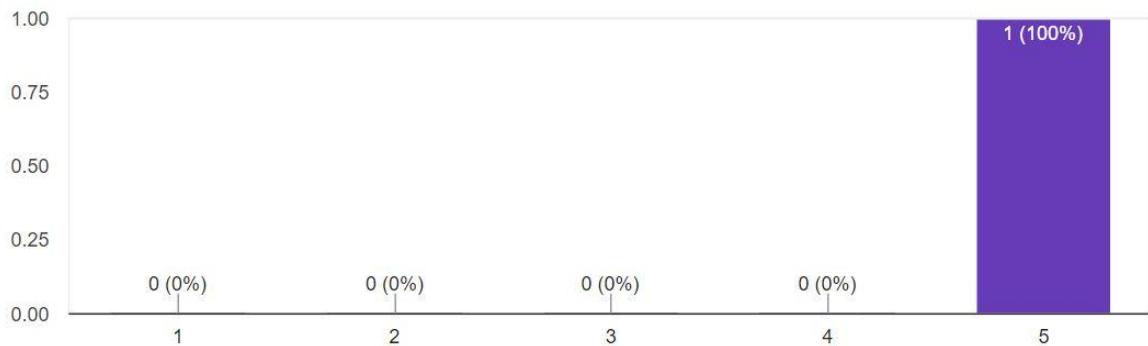


Figure 162: Client Evaluation Testing – 4

How much are you satisfied with the contents present in the application?

Copy

1 response

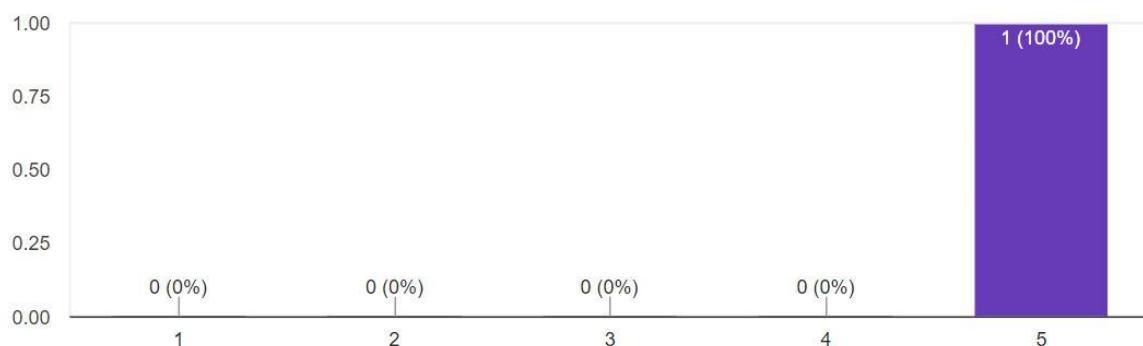


Figure 163: Client Evaluation Testing – 5

Rate the functionality of the application

Copy

1 response

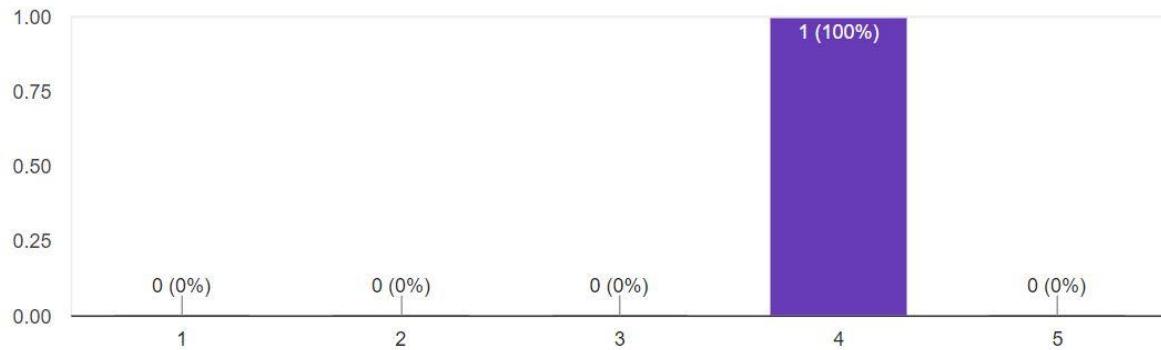


Figure 164: Client Evaluation Testing – 6

How much are you satisfied with the booking feature?

Copy

1 response

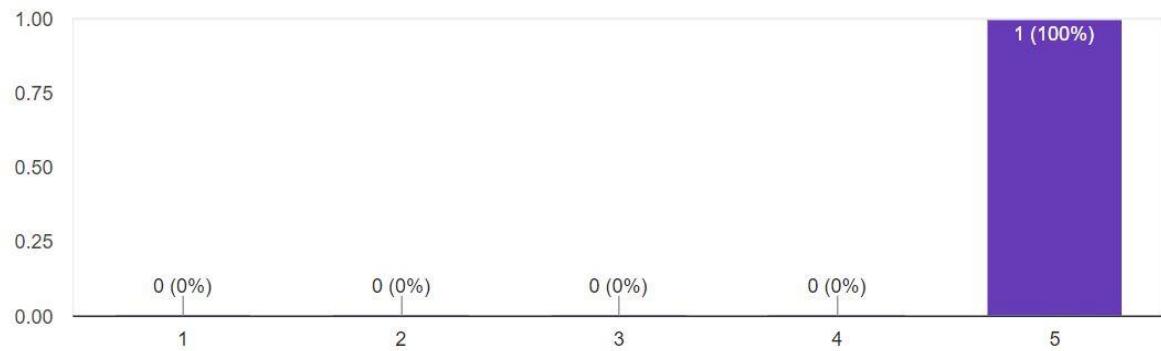


Figure 165: Client Evaluation Testing – 7

How much are you satisfied with the report feature?

Copy

1 response

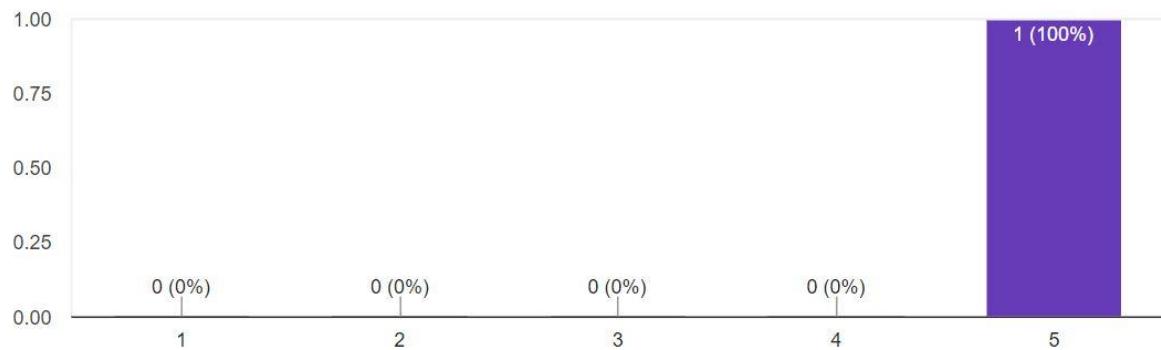


Figure 166: Client Evaluation Testing – 8

How much are you satisfied with the search feature?

Copy

1 response

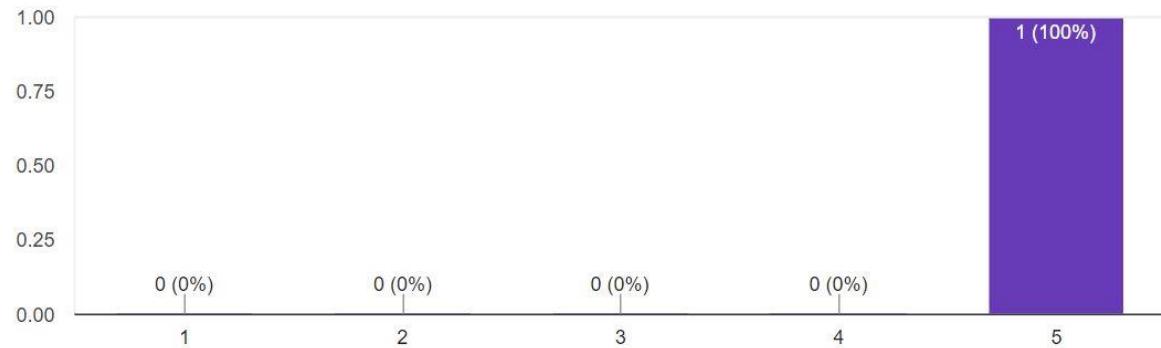


Figure 167: Client Evaluation Testing – 9

Rate the user experience while using the application

Copy

1 response

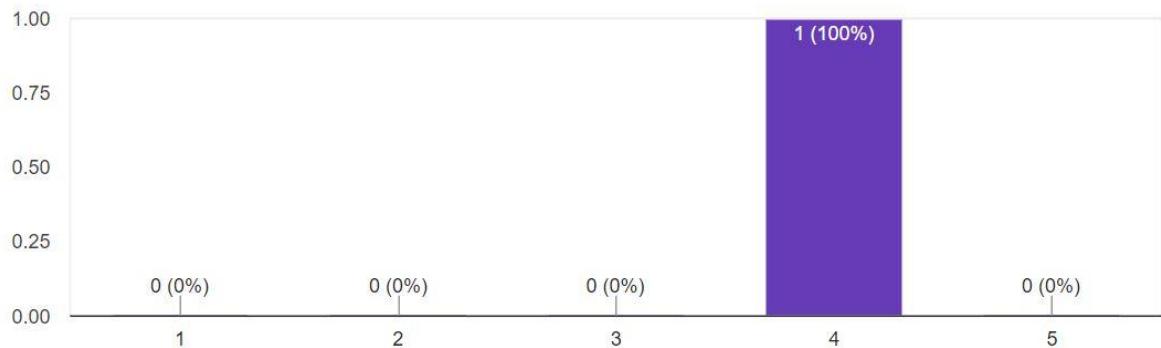


Figure 168: Client Evaluation Testing – 10

4.7 Critical Analysis

Following the completion of the black box, white box, integration, system, and client evaluation testing, it was discovered that all of the tests were successful. All viable testing was carried out at this level of the testing process, with each phase of testing being illustrated using code samples, graphical representations, and process elaboration. The system's performance was good, and the application was simple to use. For erroneous and inaccurate inputs, a nice user experience was provided by notifying users with an appropriate message. The black box testing took longer since the problems had to be fixed. The white box testing revealed that the project's overall features were working appropriately. The integration testing showed that the application was perfectly working with the database. Similarly, the system testing showed that all pages load successfully. Moreover, the client was overall satisfied with the UI and features of the application. The integration testing revealed that the application and database were in perfect sync. In the same way, system testing revealed that the system ran effectively. Furthermore, the client was also pleased with the application's UI and functionalities.

5. Conclusion

The following project included a variety of tasks and obstacles that involved relentless determination and a large number of errors to be dealt with when working. Both our internal and external supervisors offered invaluable guidance and assistance, and the provided instructions made the task simple to complete. However, in order to achieve correct output with good presentation, the rational solution of the problems, as well as the project documents, had to be carried out in a well-mannered manner.

The following project is web-based dental appointment system, which required knowledge of a variety of Front-end, Backend and Database concepts. This project helped me gain knowledge on variety of topics like HTML, Bootstrap, Python, Django, MVT Architecture, Database, Incremental Methodology and many more. There were a number of obstacles while doing this coursework. However, with the help of our supervisors, all of the doubts and misunderstandings were cleared leading to the completion of this project. This project helped in gaining knowledge of how real life applications are made. The project also had a client for whom the project was made. The interaction with the client made me one step closer to the real life software developer. Finally, a detailed report was made covering all of the topic from scratch to testing the application.

To summarize, this project was created after extensive research, hard work and dedication. Since the project had to be done alone, it was quiet challenging but with the help of resources available in the internet and the guidance of the supervisors, the final year project was finally completed.

5.1 Legal, Social and Ethical Issues

5.1.1 Legal Issues

Legal issues are those that are connected to a country's legislation, and breaching such laws leads to legal action, the decision of which is made by the court of justice itself. This project as a whole does not violate any laws that might result in legal consequence. The documentation lists all of the materials and resources utilized in the creation of this project, and they are all available online. The Dental Appointment System application is a straightforward online appointment system for user's dental issues. The data contained in this application, which includes all of the users' personal information, should not be used for personal purposes. This application ensures the confidentiality and privacy of the user's information. As a consequence, no legal difficulties arose throughout the creation of this application.

5.1.2 Social Issues

Social issues are those that influence individuals or groups of individuals living in a society. This project as a whole does not raise any kind of social issues in the society. This web application does not reflect or appear to represent any type of discrimination because it is accessible to persons of all caste, religion and ethnicity. Since every people can face dental issues and might need a dental check-up, anyone can use this application. As a result, this project has no negative consequences for society and is accessible to everyone.

5.1.3 Ethical Issues

Ethical issues are those that involve deciding whether a certain task is proper or incorrect. There are no such ethical issues that are being raised in this project. The entire project was created by me, and the code was not taken from the internet. Because this is a server-based application with sensitive information kept in a database, there are ethical concerns regarding the appointment system leaking and exploiting the user's data. The privacy and information of users will be maintained, and the data will not be made public or exploited.

5.2 Advantages

There are several advantages that the dental appointment system provides. Some of them are listed below:

- This application helps user to easily book their dental appointment.
- The user can update their profile accordingly as well.
- Users can view the details and the qualification of the doctors available and avoid any kind of uncertainty.
- User can select the doctor they want to for their appointment.
- User can view the list of services available in the clinic along with their prices and description. This makes it easy for user to manage their costs accordingly.
- The user can cancel their appointment any time before their appointment if the need arises.
- People can avoid the crowd and avoid staying in lines which saves a significant amount of time and resource. Especially in times of Covid, it helps avoid crowd and reduces the possibility of transmissions of viruses.
- User can view their report of their visit online as well. This avoids user to ever lose their report again which makes it easier for their next dental check-up anywhere in the world.
- User can search their appointment and reports and sort them accordingly as well.

5.3 Limitations

Every applications have their advantages and limitations. Similarly, this application also has its limitations in addition to the advantages above. The limitations of the application are as follows:

- The application only works when connected to the internet.
- There is no payment system integrated in to the application.
- Email verification is not implemented in the system.
- The application is only limited to one clinic i.e. Phulchowki Dental Clinic.

5.4 Future Work

The features and functions in the application is done as per the requirements of the client. However, this application can be scaled further and new features can be added to it. Some of the features that can be added to make this application even better is listed as follows:

- Payment system to confirm the booking.
- Chat-bot to address simple issues of the users.
- Chat system with the doctor so that user can avoid simple confusions.
- Email verification system.
- Notification system to remind user of their appointment status.

6. Bibliography and References

- AskPython, 2022. *AskPython*. [Online]
 Available at: <https://www.askpython.com/django/django-mvt-architecture>
 [Accessed 12 4 2022].
- Awati, R., 2022. *TechTarget*. [Online]
 Available at: <https://www.techtarget.com/searchsoftwarequality/definition/integration-testing>
 [Accessed 13 4 2022].
- Gantt.com, 2022. *Gantt.com*. [Online]
 Available at: <https://www.gantt.com/>
 [Accessed 12 4 2022].
- GeeksForGeeks, 2019. *GeeksForGeeks*. [Online]
 Available at: <https://www.geeksforgeeks.org/software-engineering-incremental-process-model/>
 [Accessed 10 12 2021].
- Ghahrai, A., 2016. *DevQA*. [Online]
 Available at: <https://devqa.io/software-development-methodologies/>
 [Accessed 11 12 2021].
- Guilizzoni, P., 2022. *Balsamiq*. [Online]
 Available at: <https://balsamiq.com/learn/articles/what-are-wireframes/>
 [Accessed 12 4 2022].
- Hamilton, T., 2022. *guru99*. [Online]
 Available at: <https://www.guru99.com/software-testing-introduction-importance.html>
 [Accessed 13 4 2022].
- Hamilton, T., 2022. *Guru99*. [Online]
 Available at: <https://www.guru99.com/black-box-testing.html>
 [Accessed 14 4 2022].
- Imperva, 2022. *Imperva*. [Online]
 Available at: <https://www.imperva.com/learn/application-security/white-box-testing/>
 [Accessed 13 4 2022].

- Lewis, S., 2019. *TechTarget*. [Online]
Available at: <https://www.techtarget.com/searchsoftwarequality/definition/collaboration-diagram>
[Accessed 11 4 2022].
- Lucid, 2022. *LucidChart*. [Online]
Available at: <https://www.lucidchart.com/blog/types-of-UML-diagrams>
[Accessed 11 4 2022].
- Martin, M., 2021. *Guru99*. [Online]
Available at: <https://www.guru99.com/software-engineering-prototyping-model.html>
[Accessed 10 12 2021].
- Mukherjee, P., 2021. *GeeksForGeeks*. [Online]
Available at: <https://www.geeksforgeeks.org/what-is-a-flowchart-and-its-types/>
[Accessed 13 4 2022].
- Organ, C., 2022. *Forbes*. [Online]
Available at: <https://www.forbes.com/advisor/business/what-is-work-breakdown-structure/#:~:text=A%20work%20breakdown%20structure%2C%20or,deliverables%20into%20a%20single%20tool.>
[Accessed 12 4 2022].
- Pankaj, P., 2019. *GeeksForGeeks*. [Online]
Available at: <https://www.geeksforgeeks.org/system-testing/>
[Accessed 14 4 2022].
- Paradigm, V., 2022. *Visual Paradigm*. [Online]
Available at: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-activity-diagram/>
[Accessed 11 4 2022].
- Paradigm, V., 2022. *Visual Paradigm*. [Online]
Available at: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-class-diagram/>
[Accessed 12 4 2022].
- Paradigm, V., 2022. *Visual Paradigm*. [Online]
Available at: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-object-diagram/>

language/what-is-sequence-diagram/

[Accessed 12 4 2022].

Peterson, R., 2022. *guru99*. [Online]

Available at: <https://www.guru99.com/er-diagram-tutorial-dbms.html>

[Accessed 12 4 2022].

Programiz, 2022. *Programiz*. [Online]

Available at: <https://www.programiz.com/dsa/algorithm>

[Accessed 13 4 2022].

Raman, 2019. *GeeksForGeeks*. [Online]

Available at: <https://www.geeksforgeeks.org/software-engineering-work-breakdown-structure/>

[Accessed 12 4 2022].

TechTarget, 2020. *TechTarget*. [Online]

Available at: <https://www.techtarget.com/whatis/definition/use-case-diagram>

[Accessed 11 4 2022].

TestBytes, 2019. *TestBytes*. [Online]

Available at: <https://www.testbytes.net/blog/rational-unified-process/>

[Accessed 10 12 2021].

TutorialsPoint, 2022. *TutorialsPoint*. [Online]

Available at:

https://www.tutorialspoint.com/data_structures_algorithms/bubble_sort_algorithm.htm

[Accessed 13 4 2022].

7. Appendix

7.1 Appendix A: Background

7.1.2 Discussion with client

Phulchowki Dental Clinic is a dental clinic located at Thaiba, Lalitpur. The founder and Managing Director of the clinic, Dr. Prashamsa Khadka has agreed to be the client for this project as she, including other staff, found this project very useful, feasible and effective.

According to Dr. Prashamsa Khadka, Founder and Managing Director of Phulchowki Dental Clinic, the following issues were identified during the discussion and meeting:

Problem:

According to her, there wasn't any system to book an appointment. People either had to call or wait in line. The lines were very long and sometimes very crowded as well. Especially in time of covid, it was very hard to even maintain social distance which increased the risk of transmission. She also stated that the records were written manually in a paper which made it very hard to access the records. Patients used to lose their reports frequently which made them difficult to have access to their previous history. So she had been looking for such system to implement.

Solution:

This system will allow users to book the appointment. Users can view the list of available doctors and select their desired doctors for their desired services. The system allows them to book, cancel and check the booking status. The report can also be uploaded which makes it easier for both patient and doctors.

Client's Requirements:

- Users should have their profile.
- Users should be able to book their appointment easily through the system.

- The system should also have features to cancel the appointment as well as check the appointment booking status.
- List of available doctors along with their description.
- Admin should be able to cancel user's appointment if the need arises.
- User should be able to view their report online as well.
- The Admin will manage the doctor's list and well as the user's list.
- The Admin will upload the doctor's approved report to the system for the user's to see.

[Click here to see the client's letter.](#)

7.2 Appendix B: Development

7.2.1 Considered Methodologies

1. RUP Methodology

The Rational Unified Process (RUP) is an approach for developing agile software. The project life cycle is divided into four phases by RUP. All six main development disciplines are practiced at each phase: business modelling, requirements, analysis and design, implementation, testing, and deployment. RUP provides a systematic way to build this type of system, focusing on the production of an executable architecture in the early stages of the project, that is, before committing resources on a large scale.

(TestBytes, 2019)

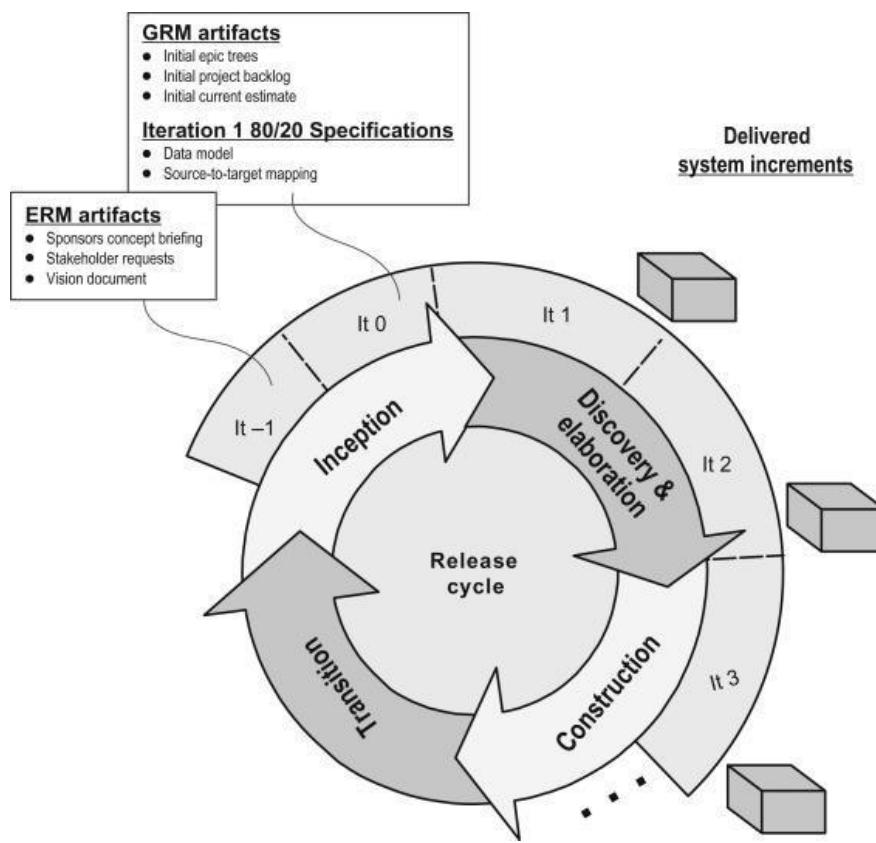


Figure 169: RUP Methodology (TestBytes, 2019)

[Click here to view the justification and reason for not selecting this methodology.](#)

2. Prototype Methodology

Prototype Model is a software development model in which prototype is built, tested, and reworked until an acceptable prototype is achieved. (Martin, 2021) It also serves as a foundation for the creation of the final system or software. It's best used in situations where the project's requirements aren't fully understood. It is an iterative, trial-and-error process that occurs between the developer and the client.

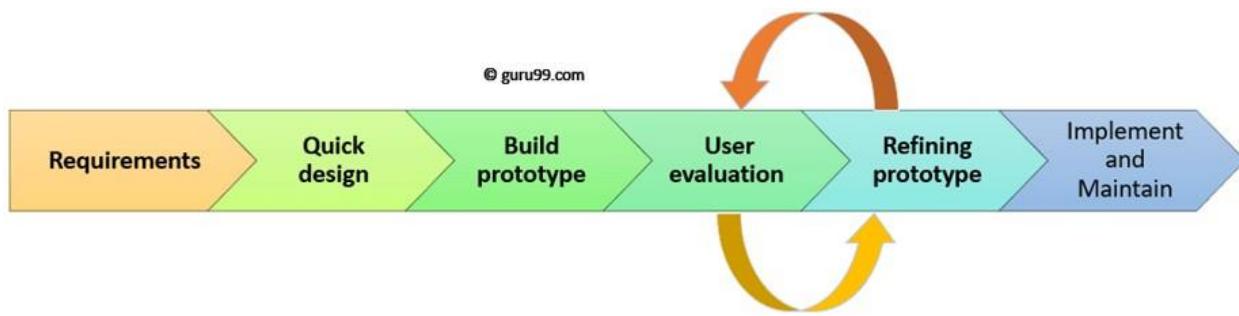


Figure 170: Prototype Methodology (Martin, 2021)

[Click here to view the justification and reason for not selecting this methodology.](#)

3. Incremental Methodology

In Incremental Methodology, first a simple working system is constructed and supplied to the customer, with only a few basic features. Over many successive iterations successive versions are implemented and delivered to the customer until the desired system is realised. (Mall, 2014) The following diagram shows the incremental development model:

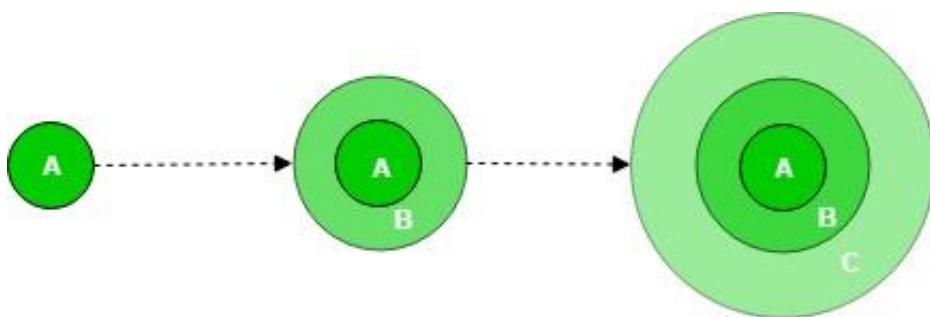


Figure 171: Incremental Methodology (GeeksForGeeks, 2019)

The product is broken down into multiple parts, each of which is designed and constructed independently (termed as builds). When all of the components have been completed, they are delivered to the client. This allows for partial product usage while avoiding a lengthy development process. It also necessitates a significant upfront financial investment, with the long wait time eliminated. This model of development also helps ease the traumatic effect of introducing completely new system all at once. (Ghahrai, 2016)

[Click here to view a brief reason to select this methodology.](#)

7.2.2 Selected Methodology

[Continued from here.](#)

Incremental Methodology

There are various types of software development methodology. Some are designed for large, high-investment projects, while others are designed for smaller ones. While developing a software, the research on a methodology should be done properly and chosen in a very conscious way. I chose the Incremental Model methodology for my project. The incremental model is a software development technique in which the model is defined, implemented, and tested one step at a time until the product is complete. It consists of both development and maintenance. When a product meets all of its specifications, it is said to be done. This model combines the elements of the waterfall model with the iterative philosophy of prototyping. (Ghahrai, 2016)

Reason to select this methodology over prototype and RUP methodology

The system is created in prototypes, which are utilized as a model for inspection, feedback collection, and improvement for the next prototype, according to the prototype approach. During the early stages, these prototypes aren't given much attention and are built quickly with the purpose of being discarded. Because this is a tiny project, prototype technique necessitates a high level of client/end-user participation and contact, which is not physically practical. It's tough to create prototypes rapidly, because each module may take different expertise to construct, with some modules reliant on others. The prior prototypes, on the other hand, are discarded and unusable. Making prototypes for each iteration takes a long time.

In RUP Methodology, the development process is way too complex and disorganized. Since this project is to be done alone, it was better to avoid such development project. The team members also are needed to be expert to develop the software. This was not possible in the given scenario. While adding new features, there might only be more

confusion and cause lots of issues during the testing phase so I chose incremental methodology over this methodology.

All things considered, we can conclude that increments was the preferable choice based on the approaches discussed above. The incremental method starts early in the development process with a first release of software and gradually enhances it. It's similar to prototyping, except instead of resuming development, it improves on what's already been done. This strategy breaks the project into minor iterations that can be handled as sub-projects and is suitable for developing a web application. Several iterations can be made to improve the web application's functions and features, as well as additions to functions in future iterations with updated versioning. It entails both development and maintenance at the same time. It's perfect since each module or component may be created independently, resulting in a better product at the end of each iteration. It is easier to estimate the project's risk because testing and debugging the program is easier in each iteration. Testing is also done in a separate step for each iteration in this manner.

7.3 Appendix C: Pre-Survey

Survey Link: <https://forms.gle/kWVfupLMvfYjMeEs5>

Pre-Survey for Dental Appointment System

Project Name: Dental Appointment System

Description of Project: Dental Appointment System is an online appointment system that helps people to save their time by scheduling an appointment for their dental check-up using a web application. This project is all about reducing the problem that people face when they have to get a dental appointment.

7.3.1 Pre-Survey Form

The screenshot shows a Google Forms survey titled "Pre-Survey for Dental Appointment System". The form features a decorative header with three cartoon teeth. The main content area contains the title and a project name field. Below that is a "Description of Project" section with a detailed description. The footer includes account information and a note indicating required fields.

Pre-Survey for Dental Appointment System

Project Name: Dental Appointment System

Description of Project:
Dental Appointment System is an online appointment system that helps people to save their time by scheduling an appointment for their dental check-up using a web application. This project is all about reducing the problem that people face when they have to get an dental appointment.

abhishekroka3@gmail.com [Switch account](#)

* Required

Figure 172: Pre-Survey Form - 1

The screenshot shows a Google Forms survey titled "Pre-Survey for Dental Appointment". The survey begins with a brief introduction: "Dental Appointment System is an online appointment system that helps people to save their time by scheduling an appointment for their dental check-up using a web application. This project is all about reducing the problem that people face when they have to get an dental appointment." Below this, there is a section for the user's email address, which is currently set to "abhishekroka3@gmail.com". A note indicates that this field is required. The next section asks for the user's name, with the placeholder "Your answer".

Figure 173: Pre-Survey Form - 2

This screenshot shows two questions from the same Google Forms survey. The first question is "How often do you have a dental checkup? *". It offers four options: "3 months", "6 months", "1 year", and "Over 1 year", each represented by a radio button. The second question is "Do you have to wait in line to get an appointment with the doctor? *". It offers three options: "Yes", "No", and "Sometimes", also represented by radio buttons.

Figure 174: Pre-Survey Form - 3

The screenshot shows a Google Forms survey titled "Pre-Survey for Dental Appointment". The first question asks if the user has ever used any online appointment booking web application, with options "Sometimes", "Yes", and "No". The second question asks if the user would like to use a web application to schedule an appointment with the doctor, with options "Yes", "No", and "Depends upon the application".

Figure 175: Pre-Survey Form - 4

The screenshot shows a Google Forms survey titled "Pre-Survey for Dental Appointment". It contains three questions: 1) "Do you think this web application will be helpful?" with options "Yes", "No", and "Maybe". 2) "Do you think having an online report of your checkup is useful?" with options "Yes", "No", and "Maybe". 3) "What features would you like to see if you'd use an application like this?" which is an open-ended text field.

Figure 176: Pre-Survey Form - 5

The screenshot shows a Google Forms survey titled "Pre-Survey for Dental Appointment". The survey consists of two questions. The first question asks, "What features would you like to see if you'd use an application like this?", with a "Your answer" text input field below it. The second question asks, "Any suggestions or issues that you encounter when scheduling a dental checkup appointment?", also with a "Your answer" text input field below it. At the bottom left is a "Submit" button, at the bottom center is a progress bar indicating "Page 1 of 1", and at the bottom right are "Clear form" and "Report Abuse" links. The Google Forms logo is visible at the bottom center.

Figure 177: Pre-Survey Form - 6

7.3.2 Sample of filled pre-survey forms

Sample 1

Dental Appointment System is an online appointment system that helps people to save their time by scheduling an appointment for their dental check-up using a web application. This project is all about reducing the problem that people face when they have to get an dental appointment.

abhishekroka3@gmail.com Switch account

* Required

Email *

abhishekroka3@gmail.com

Your Name *

Abhishek Rokaya

Figure 178: Sample 1 of pre-survey form - 1

How often do you have a dental checkup? *

- 3 months
- 6 months
- 1 year
- Over 1 year

Do you have to wait in line to get an appointment with the doctor? *

- Yes
- No
- Sometimes

Figure 179: Sample 1 of pre-survey form - 2

The screenshot shows a Google Forms survey titled "Pre-Survey for Dental Appointment". The first question asks if the user has ever used any online appointment booking web application, with options "Sometimes", "Yes", and "No". The "No" option is selected. The second question asks if the user would like to use a web application to schedule an appointment with the doctor, with options "Yes", "No", and "Depends upon the application". The "Yes" option is selected.

Figure 180: Sample 1 of pre-survey form - 3

The screenshot shows a Google Forms survey titled "Pre-Survey for Dental Appointment". The first question asks if the user thinks the web application will be helpful, with options "Yes", "No", and "Maybe". The "Yes" option is selected. The second question asks if the user thinks having an online report of their checkup is useful, with options "Yes", "No", and "Maybe". The "Yes" option is selected.

Figure 181: Sample 1 of pre-survey form - 4

The screenshot shows a Google Forms survey titled "Pre-Survey for Dental Appointment". The survey consists of two questions and a footer.

Question 1: What features would you like to see if you'd use an application like this?

I would like to see the services that are available with the cost included.

Question 2: Any suggestions or issues that you encounter when scheduling a dental checkup appointment?

Yes. I had to wait for 3 hours in line just for a 2 minute consultation in a very sunny day.

Footer:

- Submit
- Page 1 of 1
- Clear form

Never submit passwords through Google Forms.

This form was created inside of Islington College. [Report Abuse](#)

Google Forms

Figure 182: Sample 1 of pre-survey form - 5

Sample 2

The screenshot shows a Google Forms survey titled "System". The first question asks for the "Project Name" and the answer is "Dental Appointment System". The second question asks for "Email *" and the answer is "rahaul.stha12@gmail.com". The third question asks for "Your Name *" and the answer is "Rahul Shrestha". A draft save icon is visible at the top right.

Figure 183: Sample 2 of pre-survey form - 1

The screenshot shows a Google Forms survey with three questions. The first question is "How often do you have a dental checkup? *". The options are "3 months", "6 months", "1 year" (which is selected), and "Over 1 year". The second question is "Do you have to wait in line to get an appointment with the doctor? *". The options are "Yes", "No", and "Sometimes" (which is selected). The third question is "Have you ever used any online appointment booking web application? *". The options are "Yes", "No", and "Sometimes". A green edit icon is visible at the bottom right.

Figure 184: Sample 2 of pre-survey form - 2

The screenshot shows a Google Forms survey titled "Pre-Survey for Dental Appointment". The survey consists of four questions:

- "Have you ever used any online appointment booking web application? *": Options are "Yes" (selected), "No", and "Sometimes".
- "Would you like to use a web application to schedule an appointment with the doctor? *": Options are "Yes", "No", and "Depends upon the application" (selected).
- "Do you think this web application will be helpful? *": Options are "Yes", "No", and "Maybe" (selected).
- "Do you think having an online report of your checkup is useful? *": Options are "Yes" (selected), "No", and "Maybe".

A comment section at the bottom asks, "What features would you like to see if you'd use an application like this?", with the response "I would like to be able to see my report of the appointment." A green edit icon is visible in the top right corner.

Figure 185: Sample 2 of pre-survey form - 3

The screenshot shows a Google Forms survey titled "Pre-Survey for Dental Appointment". The survey consists of three questions:

- "Do you think this web application will be helpful? *": Options are "Yes", "No", and "Maybe" (selected).
- "Do you think having an online report of your checkup is useful? *": Options are "Yes" (selected), "No", and "Maybe".
- "What features would you like to see if you'd use an application like this?": A text input field contains the response "I would like to be able to see my report of the appointment." A green edit icon is visible in the top right corner.

Figure 186: Sample 2 of pre-survey form - 4

The screenshot shows a Google Forms survey titled "Pre-Survey for Dental Appointment". The form consists of several questions:

- A multiple-choice question with two options: "No" and "Maybe".
- An open-ended text area asking for features desired in a dental application. The response is: "I would like to be able to see my report of the appointment."
- An open-ended text area asking for suggestions or issues encountered during scheduling. The response is: "I lose my report frequently which causes me to do the same treatment multiple times."

At the bottom, there are buttons for "Submit", "Clear form", and a progress bar indicating "Page 1 of 1". There are also links for "Never submit passwords through Google Forms.", "Report Abuse", and the "Google Forms" logo.

Figure 187: Sample 2 of pre-survey form - 5

Sample 3

The screenshot shows a Google Forms survey titled "Pre-Survey for Dental Appointment". The form consists of several questions:

- Email ***: The input field contains "anishkhatri.k12@gmail.com".
- Your Name ***: The input field contains "Anish Khatri".
- How often do you have a dental checkup? ***: This is a radio button question with four options:
 - 3 months
 - 6 months
 - 1 year
 - Over 1 year

Figure 188: Sample 3 of pre-survey form - 1

The screenshot shows the continuation of the Google Forms survey:

- Over 1 year**: A radio button option selected.
- Do you have to wait in line to get an appointment with the doctor? ***: This is a radio button question with three options:
 - Yes
 - No
 - Sometimes
- Have you ever used any online appointment booking web application? ***: This is a radio button question with two options:
 - Yes
 - No

Figure 189: Sample 3 of pre-survey form - 2

Would you like to use a web application to schedule an appointment with the doctor? *

Yes

No

Depends upon the application

Do you think this web application will be helpful? *

Yes

No

Maybe

Figure 190: Sample 3 of pre-survey form - 3

Do you think having an online report of your checkup is useful? *

Yes

No

Maybe

What features would you like to see if you'd use an application like this?

I would like to see the details and qualification of the doctors that I will be visiting. I need to know if the doctor is qualified for my issues or not.

Any suggestions or issues that you encounter when scheduling a dental checkup appointment?

Figure 191: Sample 3 of pre-survey form - 4

The screenshot shows a Google Forms survey titled "Pre-Survey for Dental Appointment". The survey consists of two questions and a footer section.

Question 1: What features would you like to see if you'd use an application like this?

I would like to see the details and qualification of the doctors that I will be visiting. I need to know if the doctor is qualified for my issues or not.

Question 2: Any suggestions or issues that you encounter when scheduling a dental checkup appointment?

My braces were ruined because the doctor was not qualified.

Footer:

- Submit
- Page 1 of 1
- Clear form

Never submit passwords through Google Forms.

This form was created inside of Islington College. [Report Abuse](#)

Google Forms

Figure 192: Sample 3 of pre-survey form - 5

7.3.3 Pre-Survey Result

How often do you have a dental checkup?



30 responses

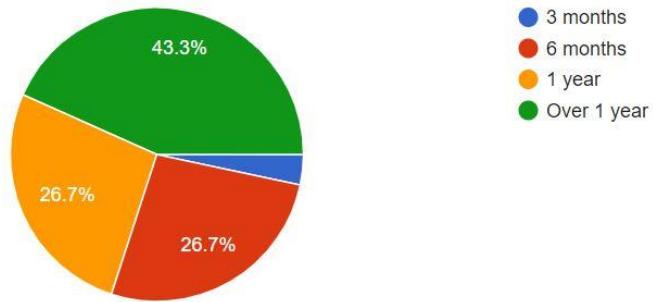


Figure 193: Pre-Survey Result – 1

Do you have to wait in line to get an appointment with the doctor?



30 responses

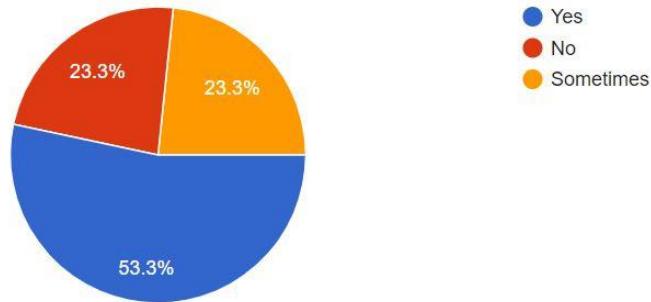


Figure 194: Pre-Survey Result - 2



Have you ever used any online appointment booking web application?

30 responses

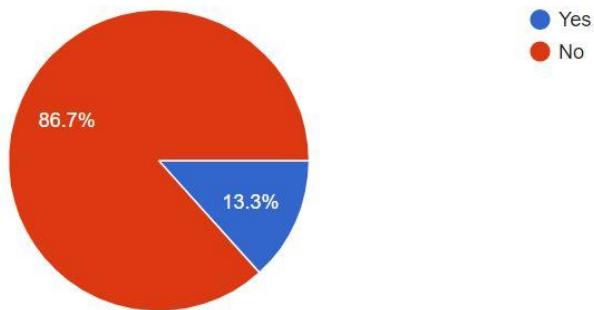


Figure 195: Pre-Survey Result – 3

Would you like to use a web application to schedule an appointment with the doctor?

30 responses

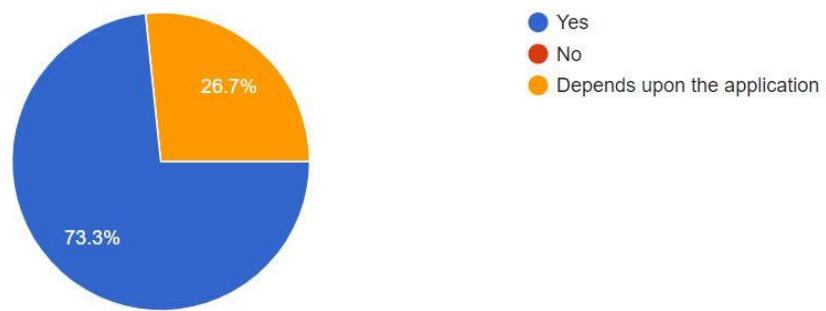


Figure 196: Pre-Survey Result - 4

Do you think this web application will be helpful?

30 responses

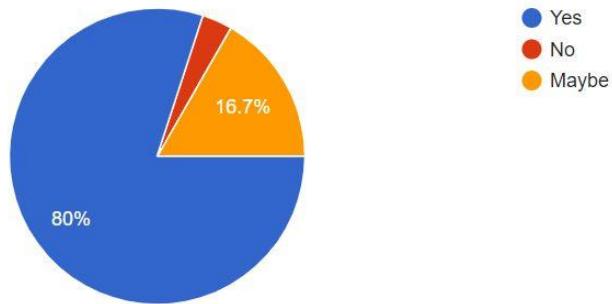


Figure 197: Pre-Survey Result - 5

Do you think having an online report of your checkup is useful?

30 responses

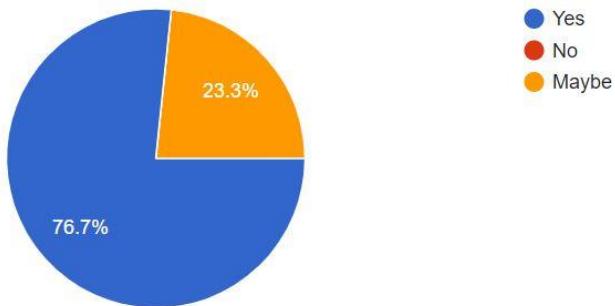


Figure 198: Pre-Survey Result - 6

What features would you like to see if you'd use an application like this?

19 responses

Online report.

An application that provides quick and immediate response while trying to book an appointment would be very helpful.

Booking appointment, seeing report

Appointments, reports, and minor prescription too.

Routine check up notification or reminders

Something that would make appointment easier and faster

-Check up history

-Next regular check up appointment reminder.

Not sure

Figure 199: Pre-Survey Result - 7

Any suggestions or issues that you encounter when scheduling a dental checkup appointment?

14 responses

The lines are too much crowded. In times of covid, it is risky to even get your teeth checked.

none

Confirmation issues, fate and time issues

Not yet

Have to wait for more than hours so it would be better if they can provide you the exact time when they can start your treatment so you can reach there accordingly.

I haven't had been to a dentist since a long so I don't remember

Nothing.

Yes, like my appointment got canceled without any information 🤦

no

Figure 200: Pre-Survey Result - 8

7.4 Appendix D: Post-Survey

Survey Link: <https://forms.gle/jkz5EsW85t98Qohf9>

Post-Survey for Dental Appointment System

Project Name: Dental Appointment System

Description of Project: Dental Appointment System is an online appointment system that helps people to save their time by scheduling an appointment for their dental check-up using a web application. This project is all about reducing the problem that people face when they have to get a dental appointment.

7.4.1 Post-Survey Form

The screenshot shows a Google Forms survey titled "Post-Survey for Dental Appointment System". The title is displayed prominently at the top of the form. Below the title, there is a brief project description. At the bottom of the form, there is a section for account information, including an email address and a note indicating that certain fields are required.

Figure 201: Post-Survey Form - 1

The screenshot shows a Google Forms survey titled "Post-Survey for Dental Appointment". It contains three questions:

- Email ***: A text input field labeled "Your email".
- Your Name ***: A text input field labeled "Your answer".
- Rate the UI of the application ***: A rating scale from 1 to 5. The scale is labeled "Very Bad" at 1 and "Very Good" at 5. The options are radio buttons.

Figure 202: Post-Survey Form - 2

The screenshot shows a Google Forms survey titled "Post-Survey for Dental Appointment". It contains two questions:

- Rate the functionality of the application ***: A rating scale from 1 to 5. The scale is labeled "Very Poor" at 1 and "Very Good" at 5. The options are radio buttons.
- Which feature did you like the most? ***: A list of options with radio buttons and a text input field for "Other". The options are:
 - Booking of appointment
 - Viewing Report
 - Browsing Doctor
 - Other: _____

Figure 203: Post-Survey Form - 3

The screenshot shows a Google Forms survey titled "Post-Survey for Dental Appointment". The survey consists of two questions. The first question asks, "Do you see yourself using this application? *", with three options: "Yes", "No", and "Maybe". The second question asks, "Would you recommend this application to others? *", also with three options: "Yes", "No", and "Maybe". Below the questions is a feedback section with a placeholder "Your answer". At the bottom right are "Submit" and "Clear form" buttons.

Figure 204: Post-Survey Form - 4

This screenshot shows the same Google Forms survey as Figure 204, but it includes two additional feedback sections. The first feedback section asks, "Any feedback regarding the application?", with a placeholder "Your answer". The second feedback section asks, "Any other features you would like to see in this application?", also with a placeholder "Your answer". At the bottom right, there are "Submit" and "Clear form" buttons, and at the bottom center, it says "Page 1 of 1".

Figure 205: Post-Survey Form - 5

7.4.2 Sample of Filled Post-Survey Forms

Sample 1

The screenshot shows a Google Forms survey titled "Post-Survey for Dental Appointment". The first question is "Email *", with the response "abhishekroka3@gmail.com". The second question is "Your Name *", with the response "Abhishek Rokaya". The third question is "Rate the UI of the application *", with a rating of 5 (Very Good) selected from a scale of 1 to 5.

Rating	Value
1	Very Bad
2	
3	
4	
5	Very Good

Figure 206: Sample 1 of post-survey form - 1

The screenshot shows a Google Forms survey titled "Post-Survey for Dental Appointment". The first question is "Rate the functionality of the application *", with a rating of 5 (Very Good) selected from a scale of 1 to 5. The second question is "Which feature did you like the most? *", with "Booking of appointment" selected as the answer.

Rating	Value
1	Very Poor
2	
3	
4	
5	Very Good

Figure 207: Sample 1 of post-survey form - 2

The screenshot shows a Google Forms survey titled "Post-Survey for Dental Appointment". The survey consists of two main sections. The first section contains a question "Do you see yourself using this application? *". Below it are three radio button options: "Yes" (selected), "No", and "Maybe". The second section contains a question "Would you recommend this application to others? *". Below it are three radio button options: "Yes" (selected), "No", and "Maybe". The background of the form is light blue.

Figure 208: Sample 1 of post-survey form - 3

The screenshot shows a completed Google Forms survey titled "Post-Survey for Dental Appointment". The survey has two sections. The first section asks for feedback on the application, with the response "It is a good application.". The second section asks for other features desired, with the response "I would like to see the integration of payment system.". At the bottom, there are "Submit" and "Clear form" buttons, and a note stating "Page 1 of 1". The background of the form is light blue.

Figure 209: Sample 1 of post-survey form - 4

Sample 2

The screenshot shows a Google Forms survey titled "Post-Survey for Dental Appointment". The survey consists of three questions:

- Email ***: The input field contains "swapniltheshadows@gmail.com".
- Your Name ***: The input field contains "Swapnil Sharma".
- Rate the UI of the application ***: A horizontal rating scale from 1 to 5. The scale is labeled "Very Bad" on the left and "Very Good" on the right. The value "4" is selected, indicated by a green circle.

Figure 210: Sample 2 of post-survey form - 1

The screenshot shows a Google Forms survey titled "Post-Survey for Dental Appointment". The survey consists of two questions:

- Rate the functionality of the application ***: A horizontal rating scale from 1 to 5. The scale is labeled "Very Poor" on the left and "Very Good" on the right. The value "5" is selected, indicated by a green circle.
- Which feature did you like the most? ***: A list of options with one selected. The options are:
 - Booking of appointment
 - Viewing Report
 - Browsing Doctor
 - Other: _____

Figure 211: Sample 2 of post-survey form - 2

The screenshot shows a Google Forms survey titled "Post-Survey for Dental Appointment". It contains three questions:

- A text input field labeled "Other:" with a placeholder.
- A question "Do you see yourself using this application? *". The "Yes" option is selected with a green radio button.
- A question "Would you recommend this application to others? *". The "Yes" option is selected with a green radio button.

Figure 212: Sample 2 of post-survey form - 3

The screenshot shows a Google Forms survey titled "Post-Survey for Dental Appointment". It contains two feedback sections:

- A text input field asking for "Any feedback regarding the application?" containing the response "The application is very easy to use."
- A text input field asking for "Any other features you would like to see in this application?" containing the response "I would like to see an online consultation feature rather than visiting the clinic."

At the bottom, there is a "Submit" button, a progress bar indicating "Page 1 of 1", and a "Clear form" link. A note at the bottom states: "Never submit passwords through Google Forms." and "This form was created inside of Islington College. [Report Abuse](#)".

Figure 213: Sample 2 of post-survey form - 4

Sample 3

The screenshot shows a Google Forms survey titled "Post-Survey for Dental Appointment". The survey consists of three questions:

- Email ***: The user has entered "ushaan2000@gmail.com".
- Your Name ***: The user has entered "Ushaan Shrestha".
- Rate the UI of the application ***: The user has rated the application with a score of 4, indicating "Very Good".

Figure 214: Sample 3 of post-survey form - 1

The screenshot shows a Google Forms survey titled "Post-Survey for Dental Appointment". The survey consists of two questions:

- Rate the functionality of the application ***: The user has rated the application with a score of 4, indicating "Very Good".
- Which feature did you like the most? ***: The user selected "Other" and provided the answer "The search and download feature in report section".

Figure 215: Sample 3 of post-survey form - 2

The screenshot shows a Google Forms survey titled "Post-Survey for Dental Appointment". The survey consists of three questions:

- Other:** The search and download feature in report section.
- Do you see yourself using this application? ***
 - Yes
 - No
 - Maybe
- Would you recommend this application to others? ***
 - Yes
 - No
 - Maybe

Figure 216: Sample 3 of post-survey form - 3

The screenshot shows a Google Forms survey titled "Post-Survey for Dental Appointment". The survey includes a feedback section and a summary of responses:

Any feedback regarding the application?
The search and download button in the report section was very helpful.

Any other features you would like to see in this application?
I would like to see a chat system with the doctor and e-sewa integration.

At the bottom, there are buttons for **Submit**, **Page 1 of 1**, and **Clear form**. A note at the bottom left says "Never submit passwords through Google Forms." and a link "This form was created inside of Islington College. Report Abuse". The "Google Forms" logo is at the bottom right.

Figure 217: Sample 3 of post-survey form - 4

7.4.3 Post-Survey Result

Rate the UI of the application

 Copy

55 responses

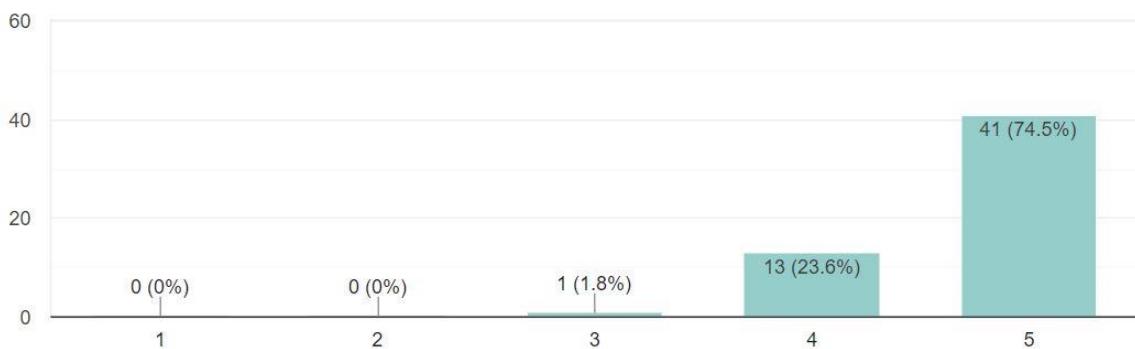


Figure 218: Post-Survey Result – 1

Rate the functionality of the application

 Copy

55 responses

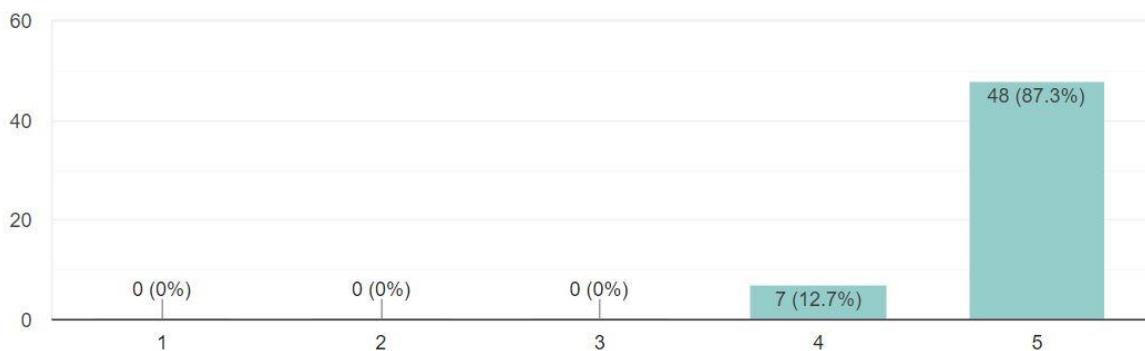


Figure 219: Post-Survey Result - 2

Which feature did you like the most?

55 responses

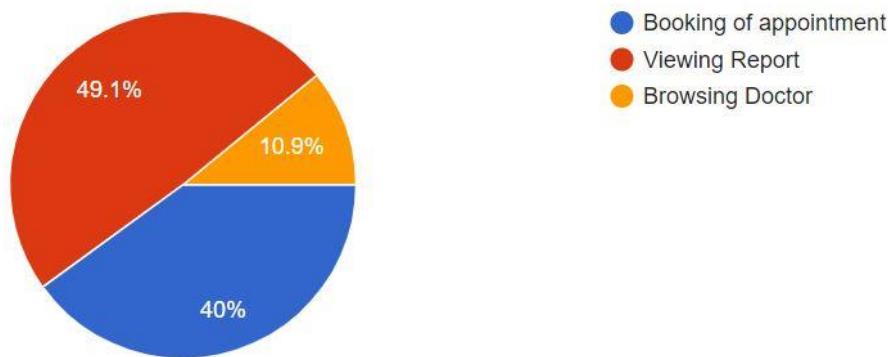


Figure 220: Post-Survey Result - 3

Do you see yourself using this application?

55 responses



Figure 221: Post-Survey Result - 4

Would you recommend this application to others?

55 responses

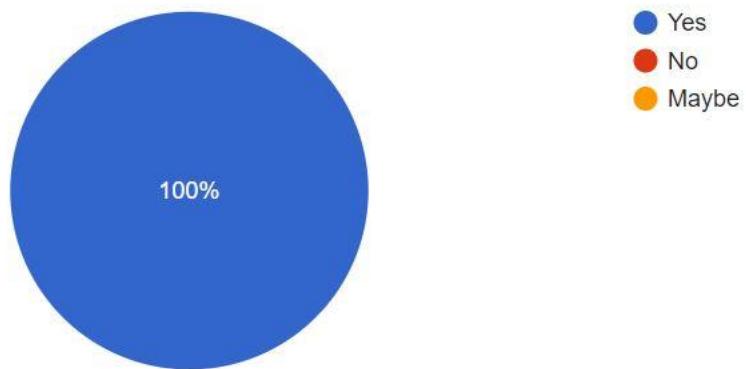


Figure 222: Post-Survey Result - 5

Any feedback regarding the application?

24 responses

Easy to use

No

A very much needed application for the modern life.

Its good

Very nice

Very nice app

Its ok

it is a good app

easy and convenient

Figure 223: Post-Survey Result - 6

Any other features you would like to see in this application?

21 responses

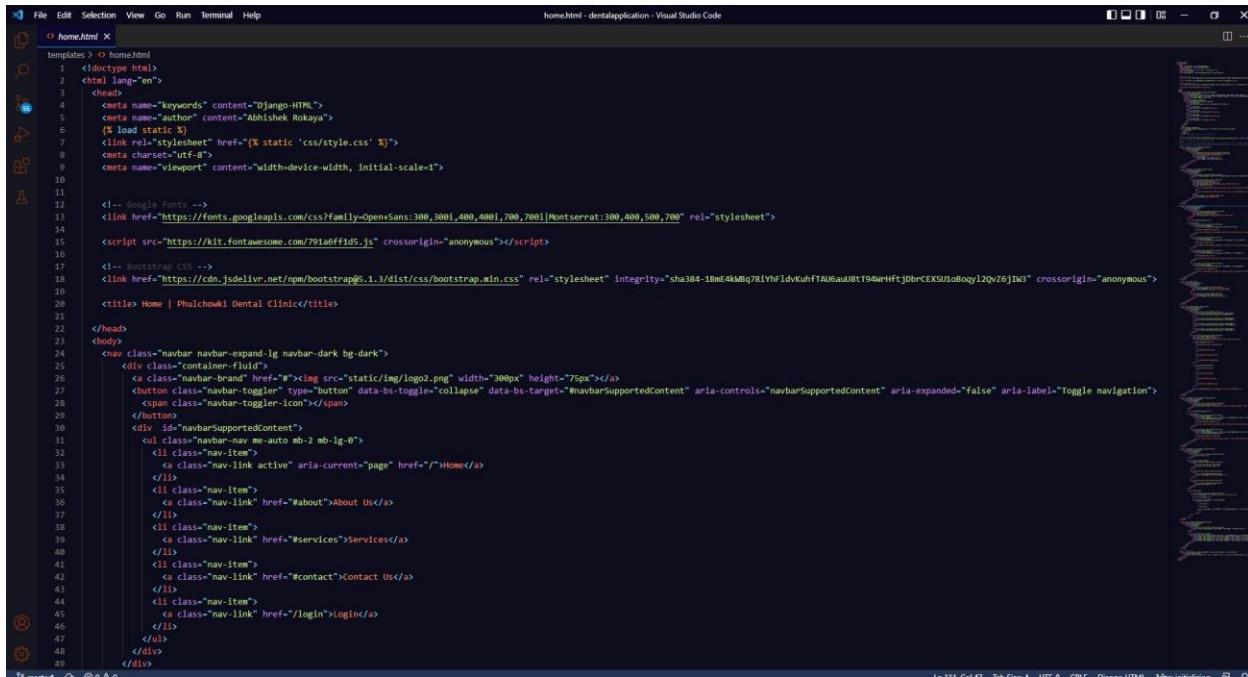


Figure 224: Post-Survey Result - 7

7.5 Appendix E: Sample Codes

7.5.1 Sample code of the UI

Home.html



The screenshot shows the Visual Studio Code interface with the 'home.html' file open. The code editor displays the following HTML and CSS code:

```
File Edit Selection View Go Run Terminal Help
home.html - dentalapplication - Visual Studio Code
templates > home.html
1 <!doctype html>
2 <html lang="en">
3   <head>
4     <meta name="keywords" content="Django-HTML">
5     <meta name="author" content="Abhishek Rokaya">
6     {% load static %}
7     <link rel="stylesheet" href="{% static 'css/style.css' %}">
8     <meta charset="utf-8">
9     <meta name="viewport" content="width=device-width, initial-scale=1">
10
11
12   <!-- Google Fonts -->
13   <link href="https://fonts.googleapis.com/css?family=Open+Sans:300,300i,400,400i,700,700i|Montserrat:300,400,500,700" rel="stylesheet">
14
15   <script src="https://kit.fontawesome.com/791a6ffids.js" crossorigin="anonymous"></script>
16
17   <!-- Bootstrap CSS -->
18   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-1BmE4kBgQ78LYldvkuhFTdau8tT94mHftJ0brCEXSUlo8qy12QvZ6J1nO" crossorigin="anonymous">
19
20   <title> Home | Phulchowki Dental Clinic</title>
21
22 </head>
23 <body>
24   <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
25     <div class="container-fluid">
26       <a class="navbar-brand" href="#"></a>
27       <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="toggle navigation">
28         <span class="navbar-toggler-icon"></span>
29       </button>
30       <div id="navbarSupportedContent">
31         <ul class="navbar-nav me-auto mb-2 mb-lg-0">
32           <li class="nav-item">
33             <a class="nav-link active" aria-current="page" href="/">Home</a>
34           </li>
35           <li class="nav-item">
36             <a class="nav-link" href="#about">About Us</a>
37           </li>
38           <li class="nav-item">
39             <a class="nav-link" href="#services">Services</a>
40           </li>
41           <li class="nav-item">
42             <a class="nav-link" href="#contact">Contact Us</a>
43           </li>
44           <li class="nav-item">
45             <a class="nav-link" href="/login">Login</a>
46           </li>
47         </ul>
48       </div>
49     </div>

```

Figure 225: Sample Code of UI - 1

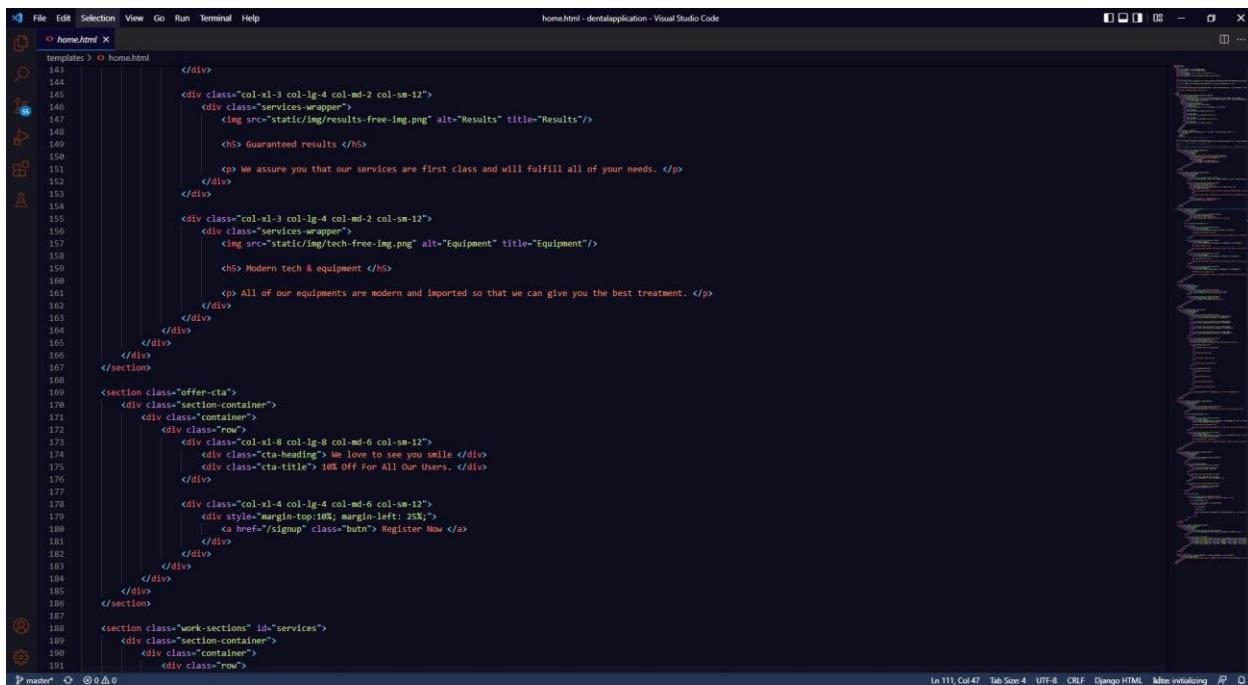
Figure 226: Sample Code of UI - 2

```
File Edit Selection View Go Run Terminal Help
homehtml - dentalapplication - Visual Studio Code

templates > home.html

96      <p>
97          We, at Phulchowki Dental Clinic, assure you to give you the best dental treatment as possible. We believe in serving our patients.
98      </p>
99      <p>
100         With our expert team, we believe we can serve you at a very affordable rate with a guaranteed success.
101     </p>
102   </div>
103   <h4 class="dr-name">Dr. Prashansa Khadka </h4>
104   <p class="dr-prof">Dental Surgeon </p>
105 </div>
106 </div>
107 </div>
108 </div>
109 </section>
110
111 <section class="clearfix service-section">
112   <div class="section-container">
113     <div class="container">
114       <div class="row">
115         <div class="col-xl-12 col-lg-12 col-md-12 col-sm-12">
116           <div class="content-top-space"></div>
117           <h2>Complete Care On Your Schedule </h2>
118           <div class="service-title"> Bringing life to your smile </div>
119         </div>
120       </div>
121
122       <div class="content-top-space"></div>
123
124       <div class="row">
125         <div class="col-xl-3 col-lg-4 col-md-2 col-sm-12">
126           <div class="services-wrapper">
127             
128
129             <h5> Highly trained dental team </h5>
130
131             <p> All of our doctors are professional and very well trained with lots of experience. </p>
132           </div>
133         </div>
134
135         <div class="col-xl-3 col-lg-4 col-md-2 col-sm-12">
136           <div class="services-wrapper">
137             
138
139             <h5> Advance treatment </h5>
140
141             <p> We perform highly advanced treatment at a regular rate so that we can see you smile. </p>
142           </div>
143         </div>
144       </div>
145     </div>
146   </div>
147 </section>
```

Figure 227: Sample Code of UI - 3



A screenshot of Visual Studio Code displaying the `home.html` file. The code is a Django template for a dental application. It includes sections for services, a call-to-action button, and a work section. The code uses Bootstrap's grid system with classes like `col-sm-12`, `col-md-2`, and `col-lg-4`. It also includes static image URLs and some CSS inline styles.

```

<div class="col-xl-3 col-lg-4 col-md-2 col-sm-12">
    <div class="services-wrapper">
        
        <h5> Guaranteed results </h5>
        <p> We assure you that our services are first class and will fulfill all of your needs. </p>
    </div>
</div>

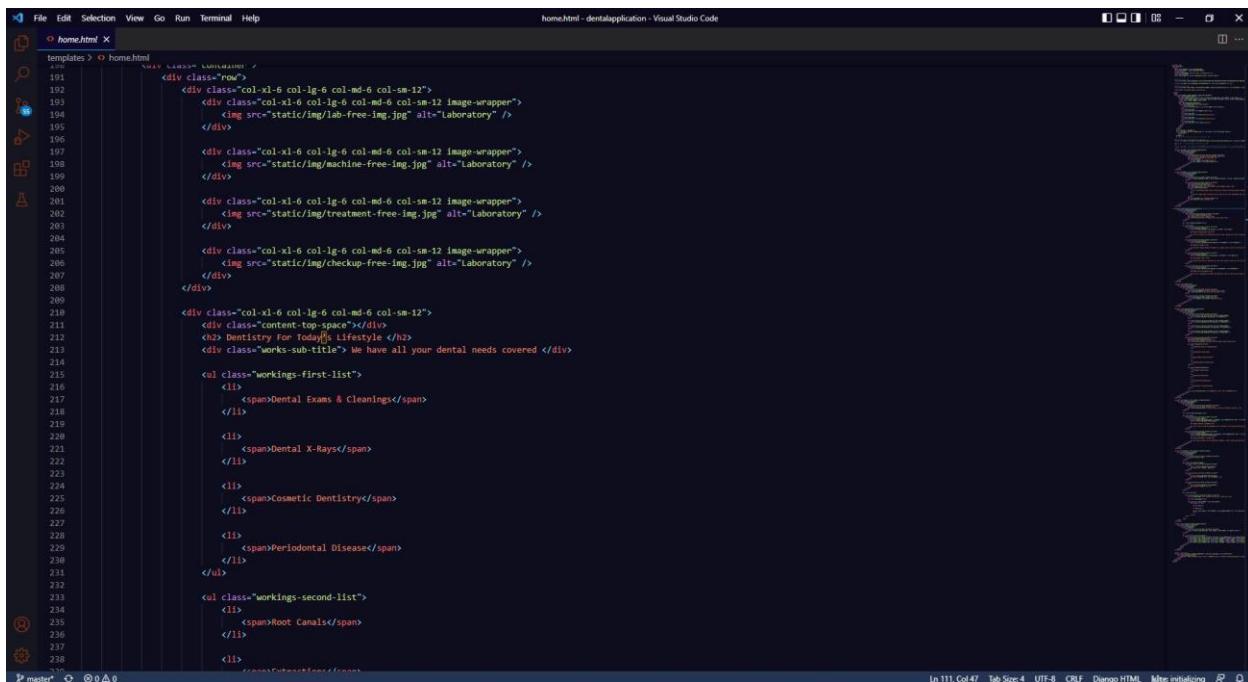
<div class="col-xl-3 col-lg-4 col-md-2 col-sm-12">
    <div class="services-wrapper">
        
        <h5> Modern tech & equipment </h5>
        <p> All of our equipments are modern and imported so that we can give you the best treatment. </p>
    </div>
</div>

<div class="offer-cta">
    <div class="section-container">
        <div class="container">
            <div class="row">
                <div class="col-xl-8 col-lg-8 col-md-6 col-sm-12">
                    <div class="cta-heading"> We love to see you smile </div>
                    <div class="cta-title"> 10% Off For All Our Users. </div>
                </div>
                <div class="col-xl-4 col-lg-4 col-md-6 col-sm-12">
                    <div style="margin-top:10%; margin-left: 25%; ">
                        <a href="/signup" class="button"> Register Now </a>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>

<div class="work-sections" id="services">
    <div class="section-container">
        <div class="container">
            <div class="row">

```

Figure 228: Sample Code of UI - 4



A screenshot of Visual Studio Code displaying the `home.html` file. This version of the template focuses on dental services. It features four columns of images with captions: 'Lab Free', 'Machine Free', 'Treatment Free', and 'Checkup Free'. Below these images is a list of services under the heading 'Dentistry for Today's Lifestyle'. The services listed include Dental Exams & Cleanings, Dental X-Rays, Cosmetic Dentistry, Periodontal Disease, Root Canals, and Implants.

```

<div class="container" >
    <div class="row">
        <div class="col-xl-6 col-lg-6 col-md-6 col-sm-12">
            <div class="image-wrapper">
                
            </div>
        </div>

        <div class="col-xl-6 col-lg-6 col-md-6 col-sm-12" image-wrapper">
            
        </div>

        <div class="col-xl-6 col-lg-6 col-md-6 col-sm-12" image-wrapper">
            
        </div>

        <div class="col-xl-6 col-lg-6 col-md-6 col-sm-12" image-wrapper">
            
        </div>
    </div>

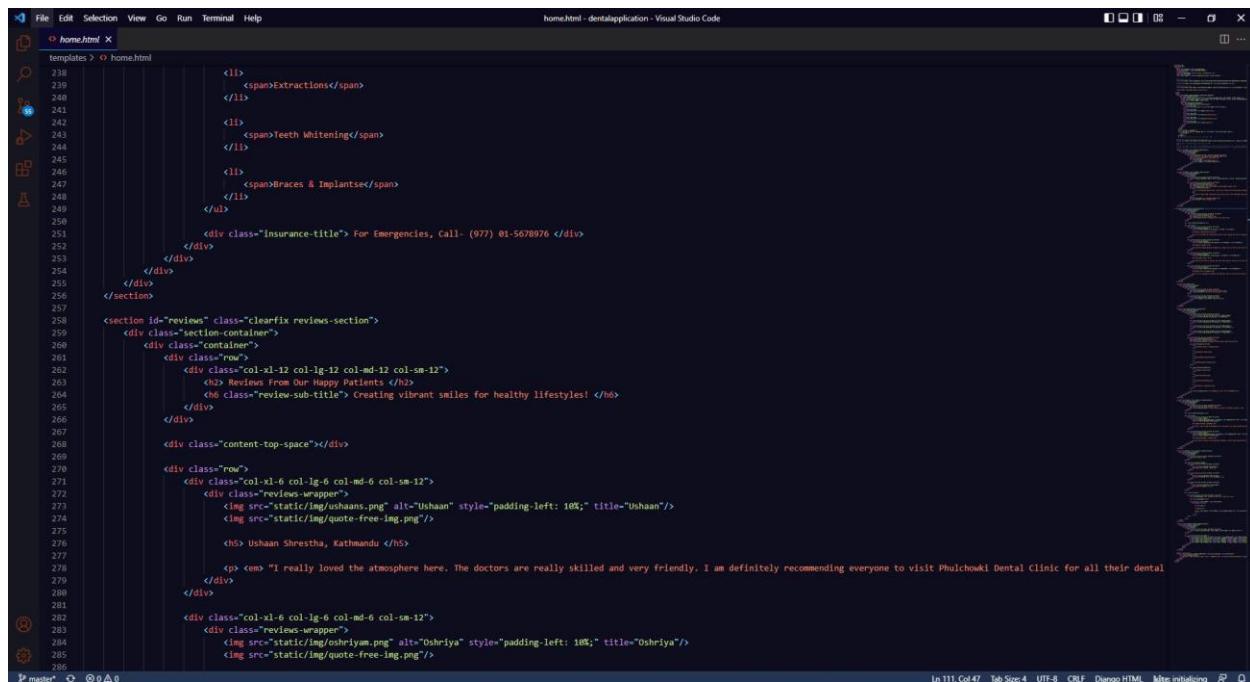
    <div class="col-xl-6 col-lg-6 col-md-6 col-sm-12">
        <div class="content-top-space"></div>
        <h2> Dentistry for Today's Lifestyle </h2>
        <div class="works-sub-title"> We have all your dental needs covered </div>
    </div>

    <ul class="workings-first-list">
        <li>
            <span> Dental Exams & Cleanings </span>
        </li>
        <li>
            <span> Dental X-Rays </span>
        </li>
        <li>
            <span> Cosmetic Dentistry </span>
        </li>
        <li>
            <span> Periodontal Disease </span>
        </li>
    </ul>

    <ul class="workings-second-list">
        <li>
            <span> Root Canals </span>
        </li>
        <li>
            <span> Implants </span>
        </li>
    </ul>

```

Figure 229: Sample Code of UI - 5

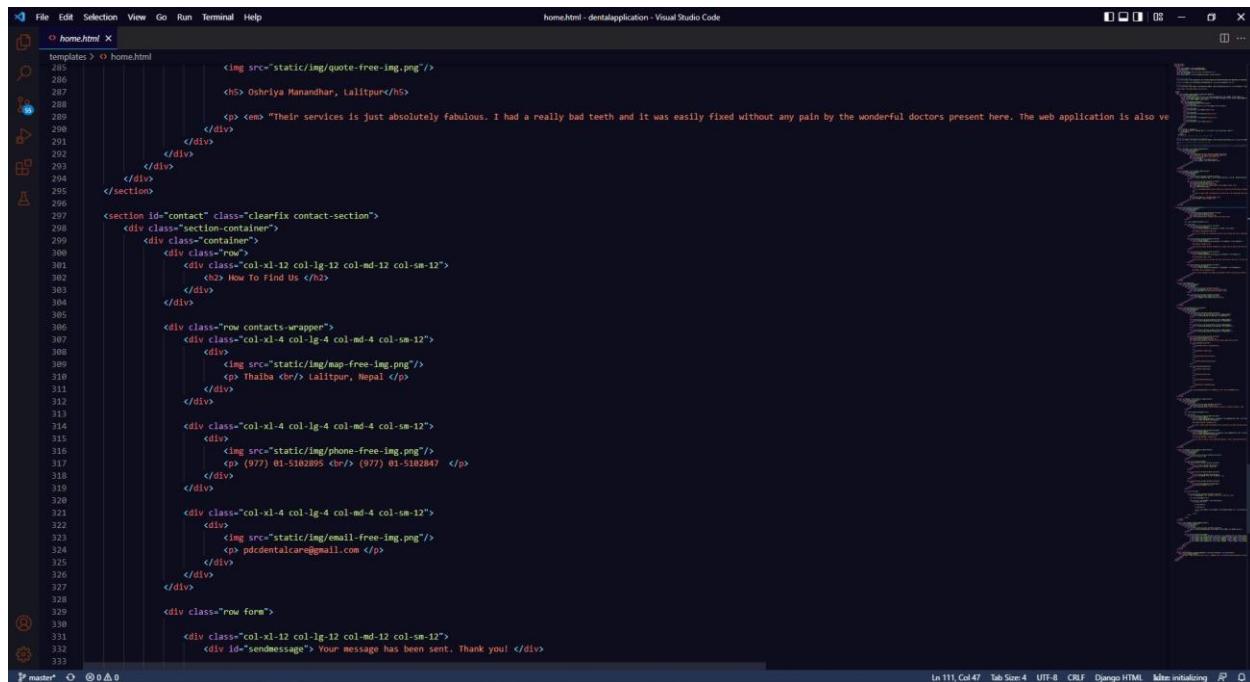


```

238         <li>
239             <span>Extractions</span>
240         </li>
241
242         <li>
243             <span>Teeth Whitening</span>
244         </li>
245
246         <li>
247             <span>Braces & Implants</span>
248         </li>
249     </ul>
250
251     <div class="insurance-title"> For Emergencies, Call- (977) 01-5678976 </div>
252   </div>
253 </div>
254 </div>
255 </section>
256
257 <section id="reviews" class="clearfix reviews-section">
258   <div class="section-container">
259     <div class="container">
260       <div class="row">
261         <div class="col-xl-12 col-lg-12 col-md-12 col-sm-12">
262           <h2> Reviews From Our Happy Patients </h2>
263           <h3 class="review-sub-title"> Creating vibrant smiles for healthy lifestyles! </h3>
264         </div>
265       </div>
266     </div>
267   </div>
268   <div class="content-top-space"></div>
269
270   <div class="row">
271     <div class="col-xl-6 col-lg-6 col-md-6 col-sm-12">
272       <div class="quote-wrapper">
273         
274         
275
276         <h5> Ushaan Shrestha, Kathmandu </h5>
277
278         <p> <br> "I really loved the atmosphere here. The doctors are really skilled and very friendly. I am definitely recommending everyone to visit Phulchowki Dental Clinic for all their dental
279
280
281
282       </div>
283     </div>
284
285     <div class="col-xl-6 col-lg-6 col-md-6 col-sm-12">
286       <div class="review-wrapper">
287         
288         
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333

```

Figure 230: Sample Code of UI - 6



```

245         
246
247         <h5> Oshriya Manandhar, Lalitpur </h5>
248
249         <p> <br> "Their services is just absolutely fabulous. I had a really bad teeth and it was easily fixed without any pain by the wonderful doctors present here. The web application is also ve
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333

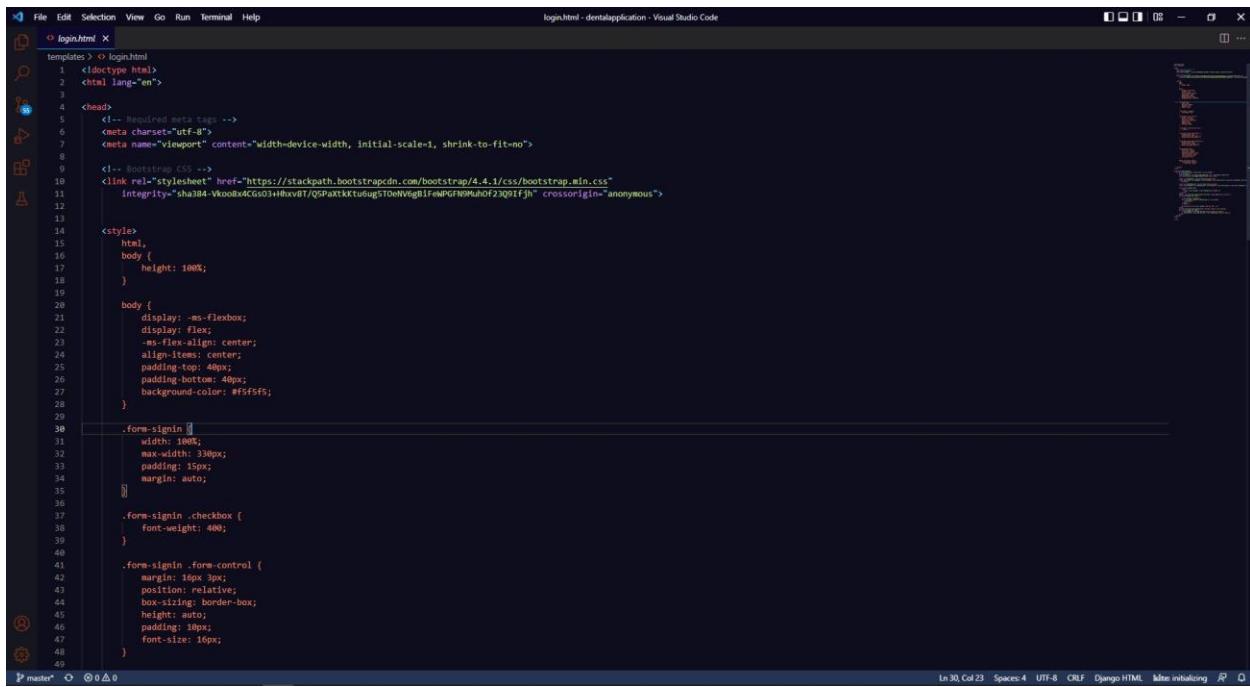
```

Figure 231: Sample Code of UI - 7

```
<File Edit Selection View Go Run Terminal Help>
home.html - dentalapplication - Visual Studio Code
templates > home.html
336     <form action="" method="POST" class="text-center">
337         <h3> Contact Us </h3>
338
339         (% csrf_token %)
340
341         {{ form.as_p }}
342
343         <button type="submit" value="Submit" style="margin-bottom: 3%; class="btn btn-info" onclick="return confirm('Are you sure you want to submit?')>submit</button>
344
345     </form>
346
347     </div>
348
349     </div>
350 </div>
351 </section>
352 <section class="clearfix footer-widgets-section">
353     <div class="section-container">
354         <div class="container">
355             <div class="row">
356                 <div class="col-xl-12 col-lg-12 col-md-12 col-sm-12">
357                     
358                 </div>
359             </div>
360         </div>
361     </div>
362     <div class="row social-icon-wrapper">
363         <div class="col-xl-12 col-lg-12 col-md-12 col-sm-12" style="padding-left: 5%; margin-bottom: -5%;>
364             <a href="https://www.instagram.com" style="color: #31b3cd" target="_blank" class="social-icon"><i class="fa fa-instagram">/i></a>
365             <a href="https://www.twitter.com" style="color: #31b3cd" target="_blank" class="social-icon"><i class="fa fa-twitter">/i></a>
366             <a href="https://www.linkedin.com" style="color: #31b3cd" target="_blank" class="social-icon"><i class="fa fa-linkedin">/i></a>
367             <a href="https://www.facebook.com" style="color: #31b3cd" target="_blank" class="social-icon"><i class="fa fa-facebook">/i></a>
368         </div>
369     </div>
370     </div>
371     </div>
372 </div>
373 </section>
374
375 <!-- Start Footer-->
376 <footer itemtype="https://schema.org/WPFooter" itemscope="ItemScope" class="site-footer">
377     <div class="copyright-footer">
378         Copyright © 2022 Phulchowki Dental Clinic | 19030695 <span><a href="#"> Abhishek Rokaya</a></span>
379     </div>
380 </footer>
381 </body>
382 </html>
```

Figure 232: Sample Code of UI - 8

Login.html

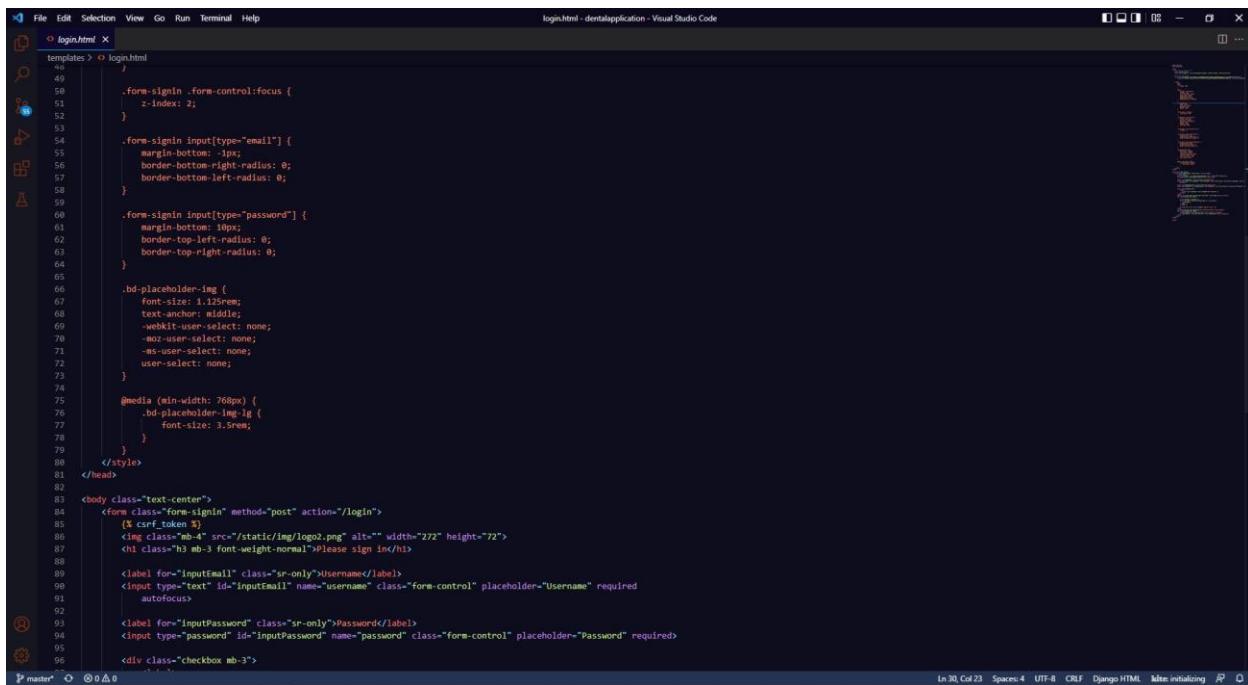


The screenshot shows the Visual Studio Code interface with the file 'login.html' open. The code is a Django template for a login page. It includes Bootstrap CSS imports and styles for the sign-in form. The code is as follows:

```
File Edit Selection View Go Run Terminal Help
templates > O login.html
1 <!DOCTYPE html>
2 <html lang="en">
3
4   <head>
5     <!-- Required meta tags -->
6     <meta charset="utf-8">
7     <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
8
9     <!-- Bootstrap CSS -->
10    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css" integrity="sha384-Vkoo8x4CGsO3IHBxv8T/QPAxkkTuusug510eNv6gBifmAPGR0Mwh0F23Q9Ifjh" crossorigin="anonymous">
11
12
13
14   <style>
15     html,
16     body {
17       height: 100%;
18     }
19
20   body {
21     display: -ms-flexbox;
22     display: flex;
23     -ms-flex-align: center;
24     align-items: center;
25     padding-top: 40px;
26     padding-bottom: 40px;
27     background-color: #f5f5f5;
28   }
29
30   .form-signin {
31     width: 100%;
32     max-width: 330px;
33     padding: 15px;
34     margin: auto;
35   }
36
37   .form-signin .checkbox {
38     font-weight: 400;
39   }
40
41   .form-signin .form-control {
42     margin: 16px 0px;
43     position: relative;
44     box-sizing: border-box;
45     height: auto;
46     padding: 10px;
47     font-size: 16px;
48   }
49
```

Ln 30, Col 23 Spaces: 4 UTF-8 CRLF Django HTML file initiating R

Figure 233: Sample Code of UI – 9



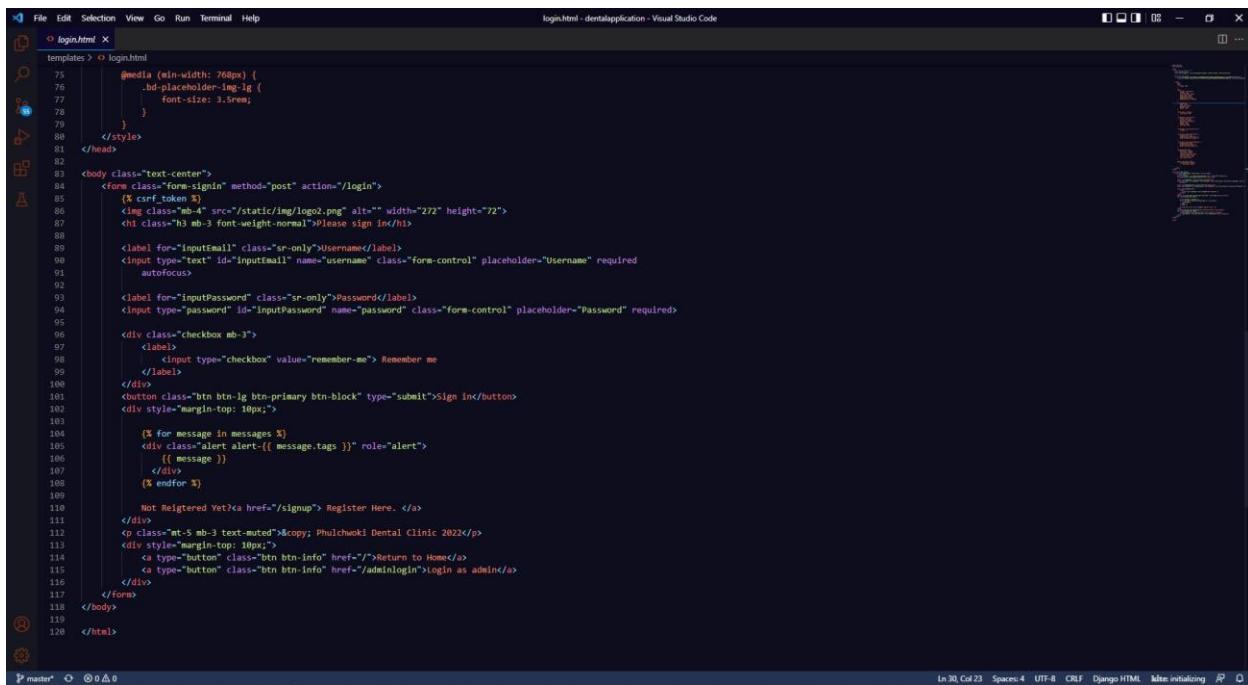
```

File Edit Selection View Go Run Terminal Help
login.html - Visual Studio Code
templates > login.html
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
</head>
</style>
</body>
</html>

```

This screenshot shows the Visual Studio Code interface with the file 'login.html' open. The code is a Django template for a login page. It includes CSS styles for form controls and a placeholder image, and HTML for a logo, user input fields, and a 'Sign In' button.

Figure 234: Sample Code of UI - 10



```

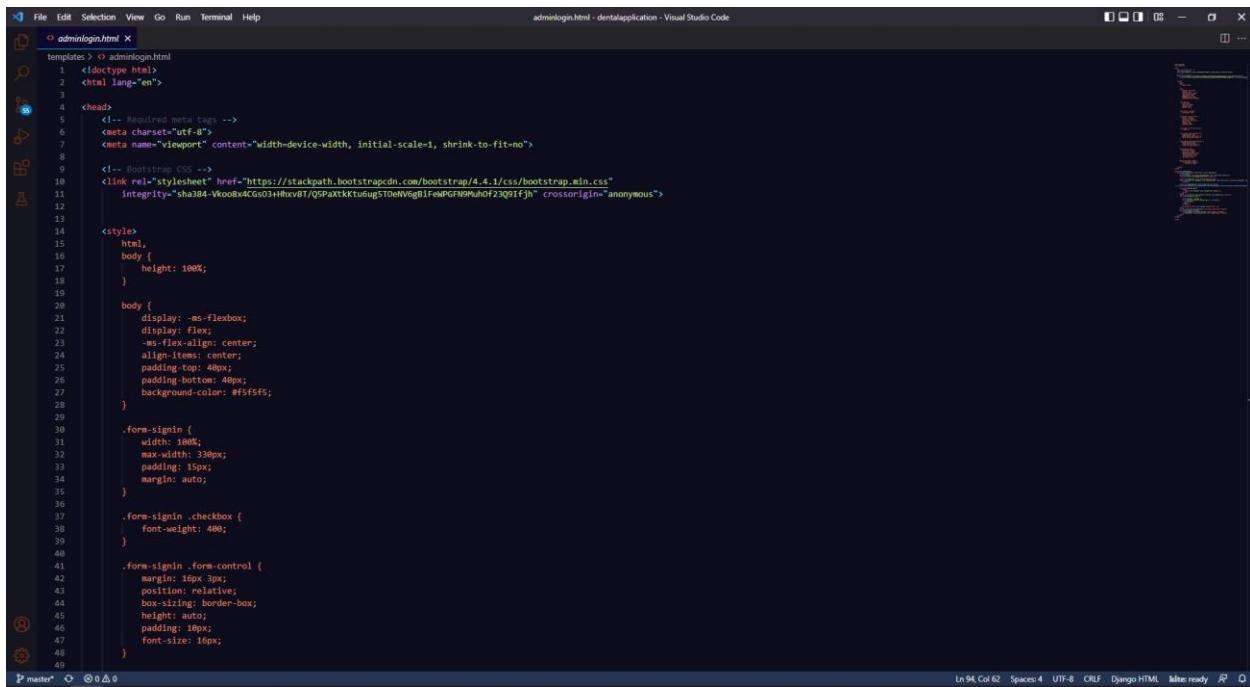
File Edit Selection View Go Run Terminal Help
login.html - Visual Studio Code
templates > login.html
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
</head>
</style>
</body>
</html>

```

This screenshot shows the Visual Studio Code interface with the file 'login.html' open. The code is a Django template for a login page, similar to Figure 234 but with more complex logic. It includes a 'remember me' checkbox and links for registration and admin login.

Figure 235: Sample Code of UI - 11

Adminlogin.html

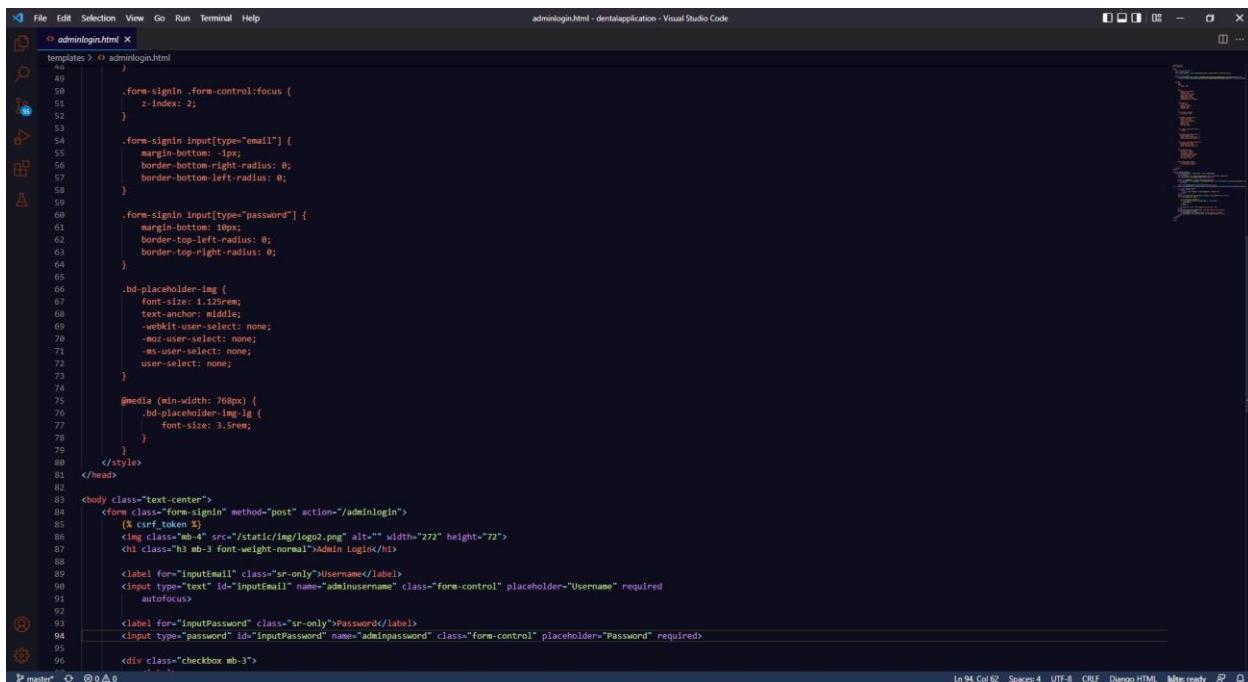


The screenshot shows the Visual Studio Code interface with the file 'adminlogin.html' open. The code is an HTML template for an administrator login page. It includes Bootstrap CSS imports and a body section with a form container and form control elements.

```
File Edit Selection View Go Run Terminal Help
templates > adminlogin.html
1 <!DOCTYPE html>
2 <html lang="en">
3
4   <head>
5     <!-- Required meta tags -->
6     <meta charset="utf-8">
7     <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
8
9     <!-- Bootstrap CSS -->
10    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css" integrity="sha384-Vkoo8xL5QVnYfR8FTyXQoT4ipDsljPIaX437xJw7mZ/iB4=" crossorigin="anonymous">
11
12
13
14   <style>
15     html,
16     body {
17       height: 100%;
18     }
19
20   body {
21     display: -ms-flexbox;
22     display: flex;
23     -ms-flex-align: center;
24     align-items: center;
25     padding-top: 40px;
26     padding-bottom: 40px;
27     background-color: #f5f5f5;
28   }
29
30   .form-signin {
31     width: 100%;
32     max-width: 330px;
33     padding: 15px;
34     margin: auto;
35   }
36
37   .form-signin .checkbox {
38     font-weight: 400;
39   }
40
41   .form-signin .form-control {
42     margin: 10px 0px;
43     position: relative;
44     box-sizing: border-box;
45     height: auto;
46     padding: 10px;
47     font-size: 16px;
48   }
49
```

Ln 94, Col 62 Spaces: 4 UTF-8 CR/LF Django HTML Editor ready J P

Figure 236: Sample Code of UI - 12



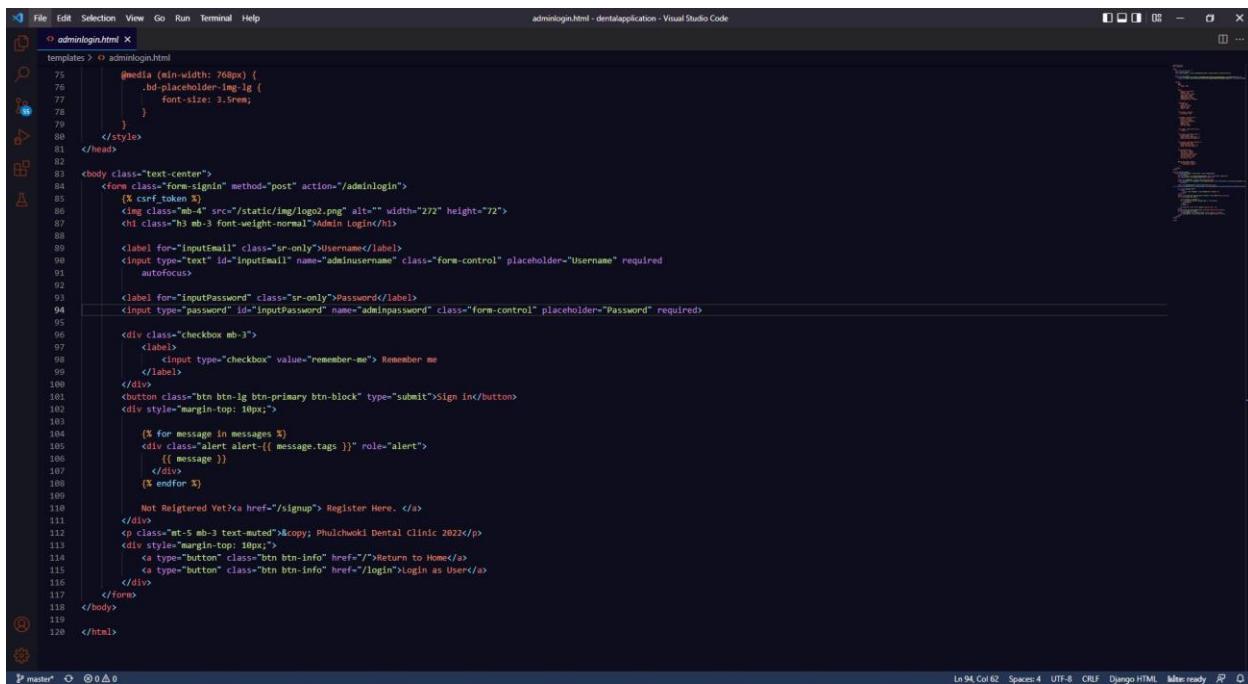
```

File Edit Selection View Go Run Terminal Help
adminlogin.html - dentalapplication - Visual Studio Code
templates > adminlogin.html
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
</style>
</head>
<body class="text-center">
<form class="form-signin" method="post" action="/adminlogin">
    {% csrf_token %}
    
    <h1 class="h3 mb-3 font-weight-normal">Admin Login</h1>
    <label for="inputEmail1" class="sr-only">Username</label>
    <input type="text" id="inputEmail1" name="adminusername" class="form-control" placeholder="Username" required
        autofocus>
    <label for="inputPassword" class="sr-only">Password</label>
    <input type="password" id="inputPassword" name="adminpassword" class="form-control" placeholder="Password" required>
    <div class="checkbox mb-3">
        <label>
            <input type="checkbox" value="remember-me" checked="checked" name="remember"> Remember me
        </label>
    </div>
    <button class="btn btn-lg btn-primary btn-block" type="submit">Sign In</button>
<div style="margin-top: 10px;">
    {% for message in messages %}
        <div class="alert alert-{% if message.tags %}{{ message.tags }}{% endif %}" role="alert">
            {{ message }}
        </div>
    {% endfor %}
    Not Registered Yet?<a href="/signup"> Register Here. </a>
</div>
<p class="mb-5 mb-3 text-align-right">&copy; Phulchowki Dental Clinic 2022</p>
<div style="margin-top: 10px;">
    <a type="button" class="btn btn-info" href="/">Return to Home</a>
    <a type="button" class="btn btn-info" href="/login">Login as User</a>
</div>
</form>
</body>
</html>

```

Line 94, Col 62 · Spaces: 4 · UTF-8 · CRLF · Django HTML · file ready

Figure 237: Sample Code of UI - 13



```

File Edit Selection View Go Run Terminal Help
adminlogin.html - dentalapplication - Visual Studio Code
templates > adminlogin.html
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
</style>
</head>
<body class="text-center">
<form class="form-signin" method="post" action="/adminlogin">
    {% csrf_token %}
    
    <h1 class="h3 mb-3 font-weight-normal">Admin Login</h1>
    <label for="inputEmail1" class="sr-only">Username</label>
    <input type="text" id="inputEmail1" name="adminusername" class="form-control" placeholder="Username" required
        autofocus>
    <label for="inputPassword" class="sr-only">Password</label>
    <input type="password" id="inputPassword" name="adminpassword" class="form-control" placeholder="Password" required>
    <div class="checkbox mb-3">
        <label>
            <input type="checkbox" value="remember-me" checked="checked" name="remember"> Remember me
        </label>
    </div>
    <button class="btn btn-lg btn-primary btn-block" type="submit">Sign In</button>
<div style="margin-top: 10px;">
    {% for message in messages %}
        <div class="alert alert-{% if message.tags %}{{ message.tags }}{% endif %}" role="alert">
            {{ message }}
        </div>
    {% endfor %}
    Not Registered Yet?<a href="/signup"> Register Here. </a>
</div>
<p class="mb-5 mb-3 text-align-right">&copy; Phulchowki Dental Clinic 2022</p>
<div style="margin-top: 10px;">
    <a type="button" class="btn btn-info" href="/">Return to Home</a>
    <a type="button" class="btn btn-info" href="/login">Login as User</a>
</div>
</form>
</body>
</html>

```

Line 94, Col 62 · Spaces: 4 · UTF-8 · CRLF · Django HTML · file ready

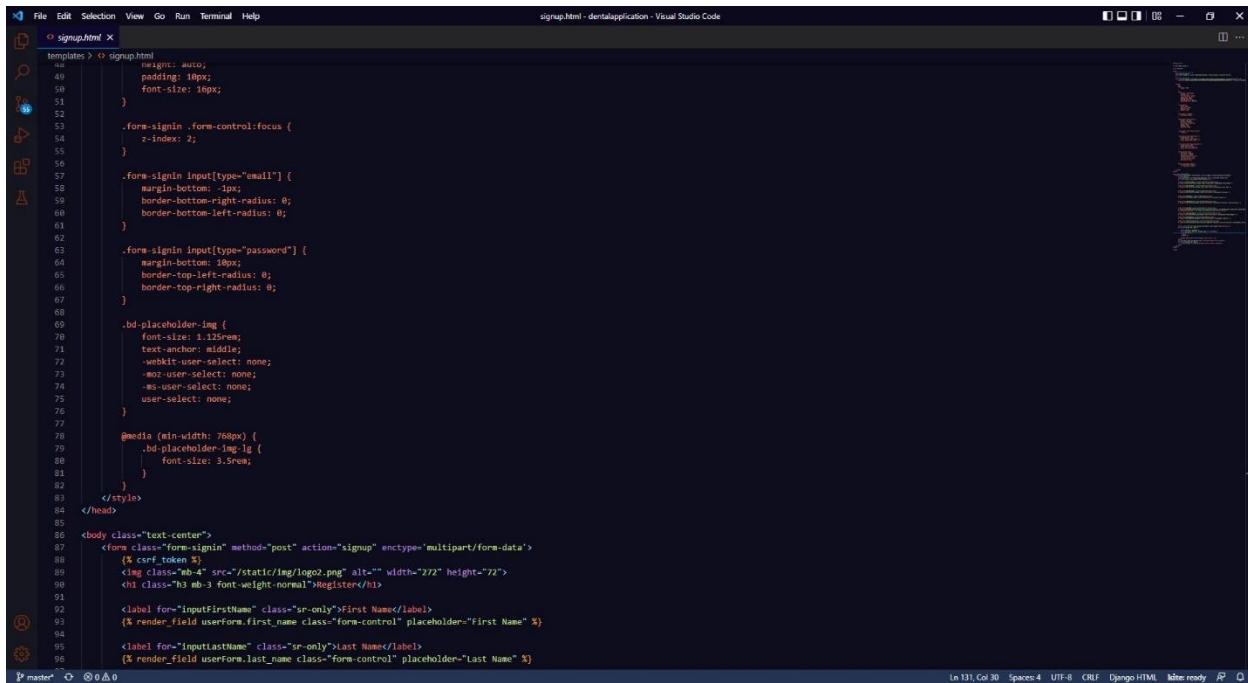
Figure 238: Sample Code of UI - 14

Signup.html

```
File Edit Selection View Go Run Terminal Help
signup.html - signup.html · dentalapplication · Visual Studio Code

templates > signup.html
1  <!DOCTYPE html>
2
3  (% load widget_tweaks %)
4
5  <html lang="en">
6
7  <head>
8      <!-- Required meta tags -->
9      <meta charset="utf-8">
10     <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
11
12     <!-- Bootstrap CSS -->
13     <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css"
14         integrity="sha384-Vkoo8XFl5vEZvJaX0O6QisZo+UpfAlWp96hwNbQ" crossorigin="anonymous">
15
16
17 <style>
18     html,
19     body {
20         height: 100%;
21     }
22
23     body {
24         display: -ms-flexbox;
25         display: flex;
26         -ms-flex-align: center;
27         align-items: center;
28         padding-top: 40px;
29         padding-bottom: 40px;
30         background-color: #f5f5f5;
31     }
32
33 .form-signin {
34     width: 100%;
35     max-width: 330px;
36     padding: 15px;
37     margin: auto;
38 }
39
40 .form-signin .checkbox {
41     font-weight: 400;
42 }
43
44 .form-signin .form-control {
45     margin: 10px 0px;
46     position: relative;
47     box-sizing: border-box;
48     height: auto;
49     padding: 10px;
```

Figure 239: Sample Code of UI - 15



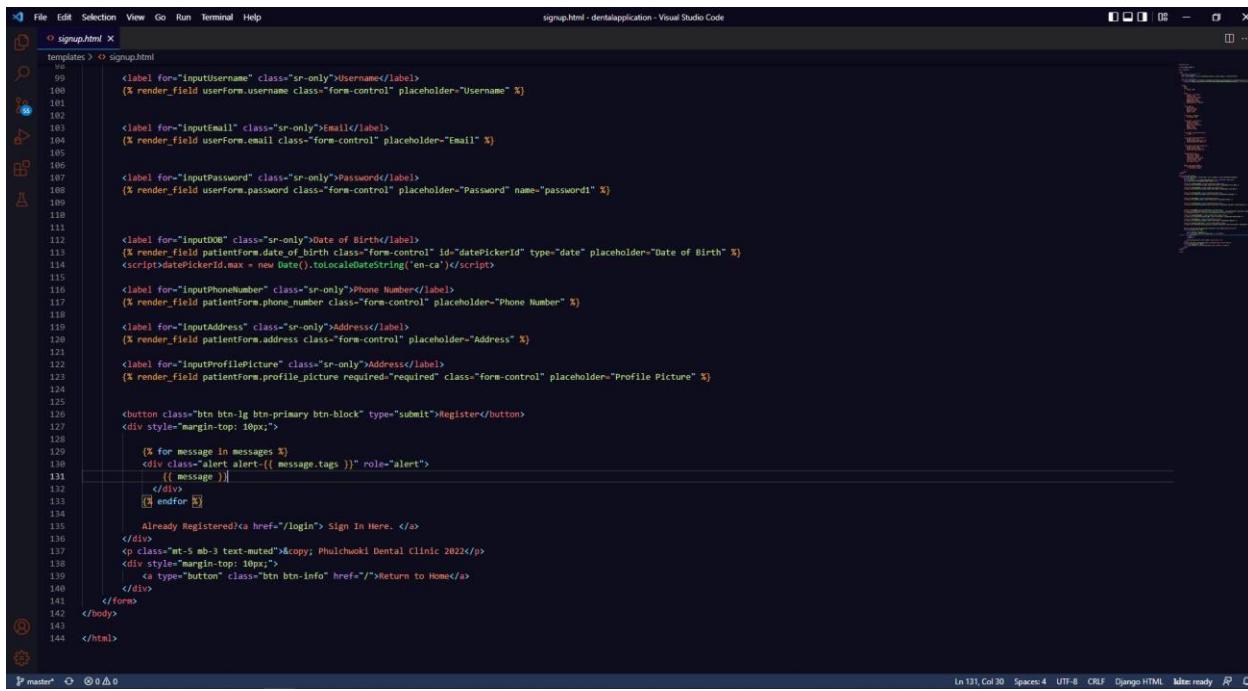
```

File Edit Selection View Go Run Terminal Help
templates > signup.html
  46     height: auto;
  47     padding: 10px;
  48     font-size: 16px;
  49   }
  50   .form-signin .form-control:focus {
  51     z-index: 2;
  52   }
  53   .form-signin input[type="email"] {
  54     margin-bottom: -1px;
  55     border-bottom-right-radius: 0;
  56     border-bottom-left-radius: 0;
  57   }
  58   .form-signin input[type="password"] {
  59     margin-bottom: 10px;
  60     border-top-left-radius: 0;
  61     border-top-right-radius: 0;
  62   }
  63   .bd-placeholder-lg {
  64     font-size: 1.125rem;
  65     text-anchor: middle;
  66     -webkit-user-select: none;
  67     -moz-user-select: none;
  68     -ms-user-select: none;
  69     user-select: none;
  70   }
  71   @media (min-width: 768px) {
  72     .bd-placeholder-lg-lg {
  73       font-size: 1.5rem;
  74     }
  75   }
  76   </style>
  77 </head>
  78 <body class="text-center">
  79   <form class="form-signin" method="post" action="signup" enctype="multipart/form-data">
  80     {% csrf_token %}
  81     
  82     <h1 class="h3 mb-3 font-weight-normal">Register</h1>
  83     <label for="inputFirstName" class="sr-only">First Name</label>
  84     {{ render_field userForm.first_name class="form-control" placeholder="First Name" }}
  85     <label for="inputLastName" class="sr-only">Last Name</label>
  86     {{ render_field userForm.last_name class="form-control" placeholder="Last Name" }}
  87   </form>
  88 </body>
  89 </html>

```

Line 131, Col 30 Spaces: 4 UTF-8 CRLF Django HTML Intel ready

Figure 240: Sample Code of UI - 16



```

File Edit Selection View Go Run Terminal Help
templates > signup.html
  99     <label for="inputUsername" class="sr-only">Username</label>
 100    {{ render_field userForm.username class="form-control" placeholder="Username" }}
 101
 102    <label for="inputEmail" class="sr-only">Email</label>
 103    {{ render_field userForm.email class="form-control" placeholder="Email" }}
 104
 105    <label for="inputPassword" class="sr-only">Password</label>
 106    {{ render_field userForm.password class="form-control" placeholder="Password" name="password" }}
 107
 108
 109    <label for="inputDOB" class="sr-only">Date of Birth</label>
 110    {{ render_field patientForm.date_of_birth class="form-control" id="datePickerId" type="date" placeholder="Date of Birth" }}
 111    <script>datePickerId.max = new Date().toLocaleDateString("en-ca")</script>
 112
 113    <label for="inputPhoneNumber" class="sr-only">Phone Number</label>
 114    {{ render_field patientForm.phone_number class="form-control" placeholder="Phone Number" }}
 115
 116    <label for="inputAddress" class="sr-only">Address</label>
 117    {{ render_field patientForm.address class="form-control" placeholder="Address" }}
 118
 119    <label for="inputProfilePicture" class="sr-only">Address</label>
 120    {{ render_field patientForm.profile_picture required="required" class="form-control" placeholder="Profile Picture" }}
 121
 122
 123    <button class="btn btn-lg btn-primary btn-block" type="submit">Register</button>
 124    <div style="margin-top: 10px;">
 125      <% for message in messages %>
 126        <div class="alert alert-{{ message.tags }}" role="alert">
 127          {{ message }}
 128        </div>
 129      <% endfor %>
 130      Already Registered!<a href="/login"> Sign In Here. </a>
 131    </div>
 132
 133    <p class="mt-5 mb-3 text-muted">&copy; Phulchokli Dental Clinic 2022</p>
 134    <div style="margin-top: 10px;">
 135      <a type="button" class="btn btn-info" href="/">Return to Home</a>
 136    </div>
 137
 138  </form>
 139 </body>
 140 </html>

```

Line 131, Col 30 Spaces: 4 UTF-8 CRLF Django HTML Intel ready

Figure 241: Sample Code of UI - 17

7.5.1.1 User side

Userbase.html

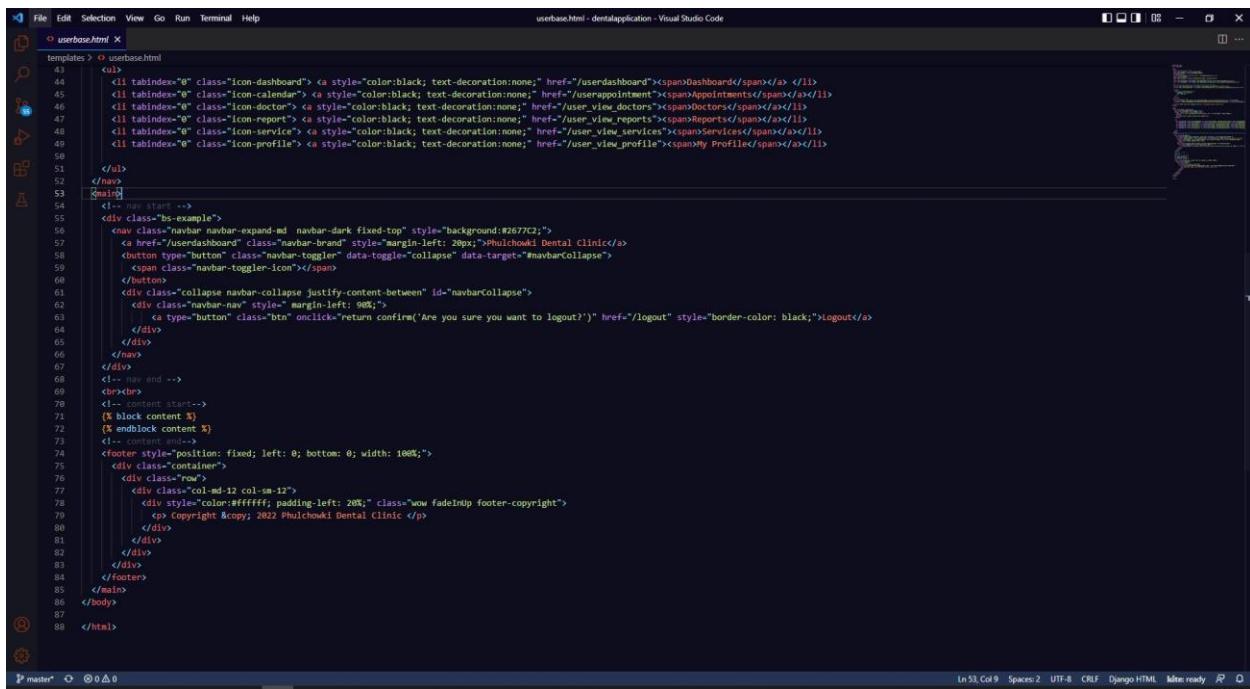
```

File Edit Selection View Go Run Terminal Help
templates > userbase.html
templates > userbase.html
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <!-- Required meta tags -->
6   <meta name="keywords" content="Django-HTML">
7   <meta name="author" content="Abhishek Rokaya">
8   </!> load static in X
9   <link rel="stylesheet" href="{% static 'css/dashboardstyle.css' %}">
10  <meta charset="utf-8">
11  <meta name="viewport" content="width=device-width, initial-scale=1">
12  <!-- Loading CSS for saving the table and searching it -->
13  <link rel="stylesheet" href="https://cdn.datatables.net/1.11.5/css/jquery.dataTables.min.css">
14  <link rel="stylesheet" href="https://cdn.datatables.net/buttons/2.2.2/css/buttons.dataTables.min.css">
15
16  <!-- Loading JavaScript for saving the table and searching it-->
17  <script type="text/javascript" src="https://code.jquery.com/jquery-3.5.1.js"></script>
18  <script type="text/javascript" src="https://cdn.datatables.net/1.11.5/js/jquery.dataTables.min.js"></script>
19
20 <script>
21   $(document).ready(function () {
22     $('#example').DataTable({
23       dom: 'Bfrtip',
24       ...
25     });
26   });
27 <!-- Bootstrap CSS -->
28 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet"
29      integrity="sha384-1BmE4kWBq78wY5EfgOrlu7WZlV2vKU+qev9eXZf/DyjXK2du+MzQqJ4q6+QHh&gt;
30      crossorigin="anonymous">
31 <title>{% block title %}{% endblock title %} | Phulchowki Dental Care</title>
32
33 </head>
34
35 <body>
36   <nav class="menu" tabindex="0">
37     <div class="smartphone-menu-trigger"></div>
38     <header class="avatar" style="height: 35%;>
39       
40       <br><br>
41       <h2 style="color: black;">{{user.first_name}}</h2>
42     </header>
43
44     <ul>
45       <li tabindex="0" class="icon-dashboard"> <a style="color:black; text-decoration:none;" href="/userdashboard"><span>Dashboard</span></a> </li>
46       <li tabindex="0" class="icon-calendar"> <a style="color:black; text-decoration:none;" href="/userappointment"><span>Appointments</span></a></li>
47       <li tabindex="0" class="icon-doctor"> <a style="color:black; text-decoration:none;" href="/user_view_doctors"><span>Doctors</span></a></li>
48       <li tabindex="0" class="icon-report"> <a style="color:black; text-decoration:none;" href="/user_view_reports"><span>Reports</span></a></li>
49       <li tabindex="0" class="icon-service"> <a style="color:black; text-decoration:none;" href="/user_view_services"><span>Services</span></a></li>
50       <li tabindex="0" class="icon-profile"> <a style="color:black; text-decoration:none;" href="/user_view_profile"><span>My Profile</span></a></li>
51     </ul>
52   </nav>
53
54   <div class="content">
55     <h1>User Dashboard</h1>
56     <p>Welcome, {{user.first_name}}!</p>
57     <p>Your dashboard is your central hub for managing your dental care. You can view your appointments, access reports from your previous visits, and explore our services. If you have any questions or need further assistance, feel free to contact us.</p>
58   </div>
59
60 </body>
61
62 </html>

```

Ln 53, Col 9 | Spaces: 2 | UTF-8 | CRLF | Django HTML | hint: ready | 0

Figure 242: Sample Code of UI - 18



A screenshot of the Visual Studio Code interface showing the `userbase.html` file. The code is an HTML template for a user dashboard. It includes a navigation bar with links to Dashboard, Appointment, Doctors, Reports, Services, and Profile. Below the nav bar is a collapseable section containing a logout button. At the bottom is a footer with copyright information. The code uses Bootstrap classes like `navbar-expand-md`, `fixed-top`, and `collapse`. The footer is styled with `position: fixed; left: 0; bottom: 0; width: 100%`.

```
File Edit Selection View Go Run Terminal Help
templates > userbase.html X
43     <ul class="icon-list">
44         <li tabindex="0" class="icon-dashboard"><a style="color:black; text-decoration:none;" href="/userdashboard"><span>Dashboard</span></a></li>
45         <li tabindex="0" class="icon-calendar"><a style="color:black; text-decoration:none;" href="/userappointment"><span>Appointments</span></a></li>
46         <li tabindex="0" class="icon-doctor"><a style="color:black; text-decoration:none;" href="/user_view_doctors"><span>Doctors</span></a></li>
47         <li tabindex="0" class="icon-report"><a style="color:black; text-decoration:none;" href="/user_view_reports"><span>Reports</span></a></li>
48         <li tabindex="0" class="icon-service"><a style="color:black; text-decoration:none;" href="/user_view_services"><span>Services</span></a></li>
49         <li tabindex="0" class="icon-profile"><a style="color:black; text-decoration:none;" href="/user_view_profile"><span>My Profile</span></a></li>
50     </ul>
51 
```

```
</nav>
52 <!-- new start -->
53 <div class="bs-example">
54     <div class="navbar navbar-expand-md navbar-dark fixed-top" style="background:#2677C2;">
55         <a href="/userdashboard" class="navbar-brand" style="margin-left: 20px;">Phulchowki Dental Clinic</a>
56         <button type="button" class="navbar-toggler" data-toggle="collapse" data-target="#navbarCollapse">
57             <span>Nav</span> <span>Toggle</span>
58         </button>
59         <div class="collapse navbar-collapse justify-content-between" id="navbarCollapse">
60             <div class="navbar-nav" style="margin-left: 90px;">
61                 <a type="button" class="btn" onclick="return confirm('Are you sure you want to logout?')" href="/logout" style="border-color: black;">Logout</a>
62             </div>
63         </div>
64     </div>
65 
```

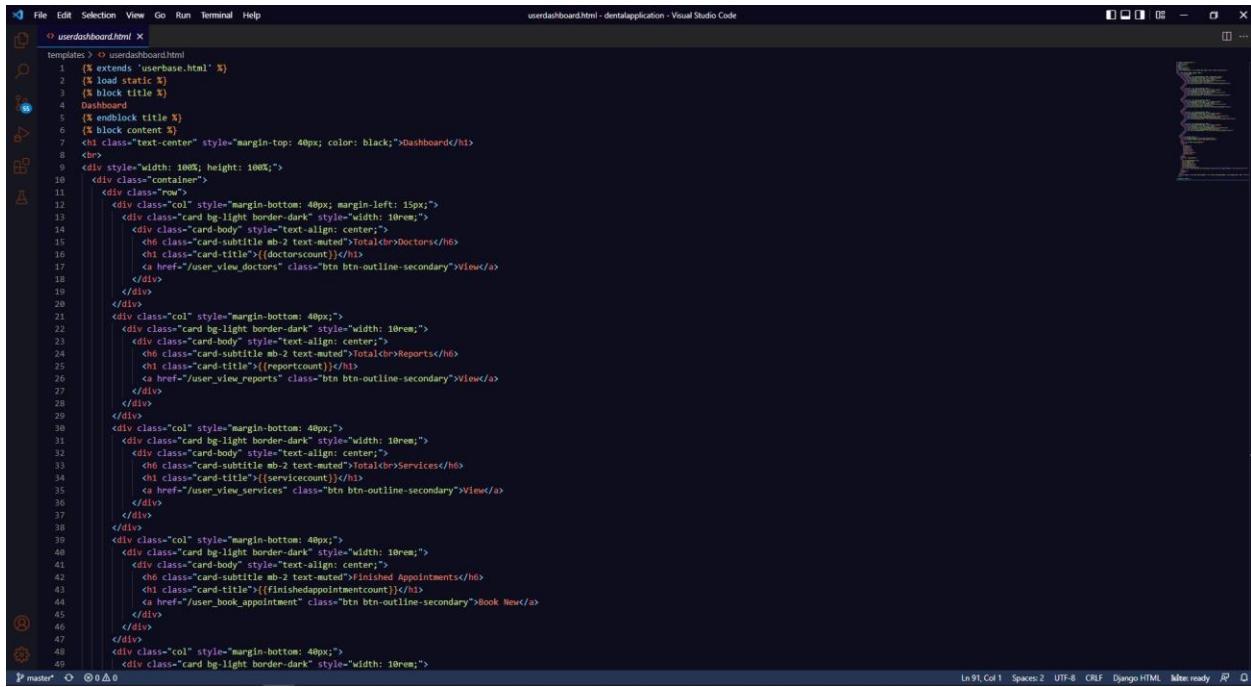
```
</div>
66 <!-- new end -->
67 <br>
68 <!-- content start -->
69 <!-- content end -->
70 <!-- content start -->
71 <!-- content end -->
72 <!-- content start -->
73 <!-- content end -->
74 <footer style="position: fixed; left: 0; bottom: 0; width: 100%">
75     <div style="text-align: center; padding: 10px; background-color: #f0f0f0; border-top: 1px solid #ccc; border-bottom: 1px solid #ccc;">
76         <div class="row">
77             <div class="col-md-12 col-sm-12">
78                 <div style="color:#ffffff; padding-left: 20px;" class="wow fadeInUp footer-copyright">
79                     <p>Copyright &copy; 2022 Phulchowki Dental Clinic </p>
80                 </div>
81             </div>
82         </div>
83     </div>
84 
```

```
</div>
85 </main>
86 </body>
87 
```

```
</html>
```

Figure 243: Sample Code of UI - 19

Userdashboard.html



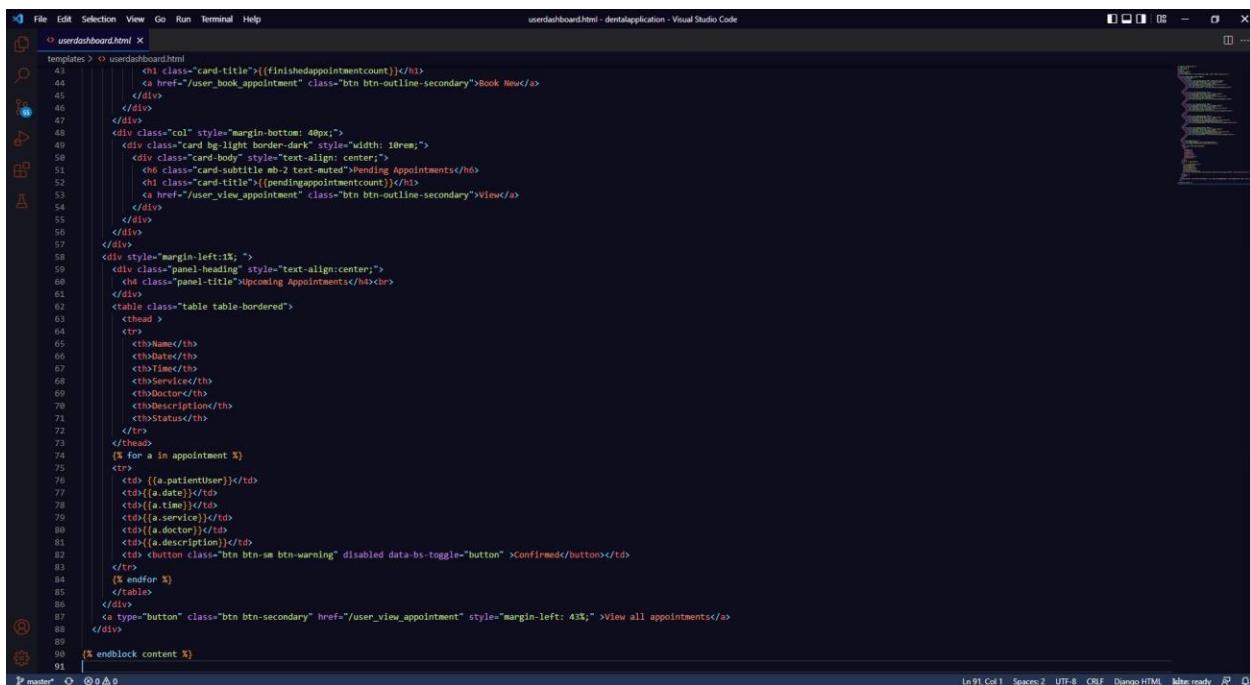
```

File Edit Selection View Go Run Terminal Help
templates > userdashboard.html
1  {% extends 'userbase.html' %} 
2  {% load static %} 
3  {% block title %} 
4  Dashboard
5  {% endblock title %} 
6  {% block content %} 
7  <h1 class="text-center" style="margin-top: 40px; color: black;">Dashboard 
9  <div class="container">
10 <div class="row">
11   <div class="col" style="margin-bottom: 40px; margin-left: 15px;">
12     <div class="card bg-light border-dark" style="width: 10rem;">
13       <div class="card-body" style="text-align: center;">
14         <h6 class="card-subtitle mb-2 text-muted">Total Doctors</h6>
15         <h3 class="card-title">{{doctorscount}}</h3>
16         <a href="/user_view_doctors" class="btn btn-outline-secondary">View</a>
17       </div>
18     </div>
19   </div>
20   <div class="col" style="margin-bottom: 40px;">
21     <div class="card bg-light border-dark" style="width: 10rem;">
22       <div class="card-body" style="text-align: center;">
23         <h6 class="card-subtitle mb-2 text-muted">Total Reports</h6>
24         <h3 class="card-title">{{reportcount}}</h3>
25         <a href="/user_view_reports" class="btn btn-outline-secondary">View</a>
26       </div>
27     </div>
28   </div>
29   <div class="col" style="margin-bottom: 40px;">
30     <div class="card bg-light border-dark" style="width: 10rem;">
31       <div class="card-body" style="text-align: center;">
32         <h6 class="card-subtitle mb-2 text-muted">Total Services</h6>
33         <h3 class="card-title">{{servicecount}}</h3>
34         <a href="/user_view_services" class="btn btn-outline-secondary">View</a>
35       </div>
36     </div>
37   </div>
38 </div>
39   <div class="col" style="margin-bottom: 40px;">
40     <div class="card bg-light border-dark" style="width: 10rem;">
41       <div class="card-body" style="text-align: center;">
42         <h6 class="card-subtitle mb-2 text-muted">Total Appointments</h6>
43         <h3 class="card-title">{{finishedappointmentcount}}</h3>
44         <a href="/user_book_appointment" class="btn btn-outline-secondary">Book Now</a>
45       </div>
46     </div>
47   </div>
48 </div class="row">
49 </div class="card bg-light border-dark" style="width: 10rem;">

```

Ln 91, Col 1 Spaces: 2 UTF-8 CRLF Django HTML Idle: ready R

Figure 244: Sample Code of UI - 20

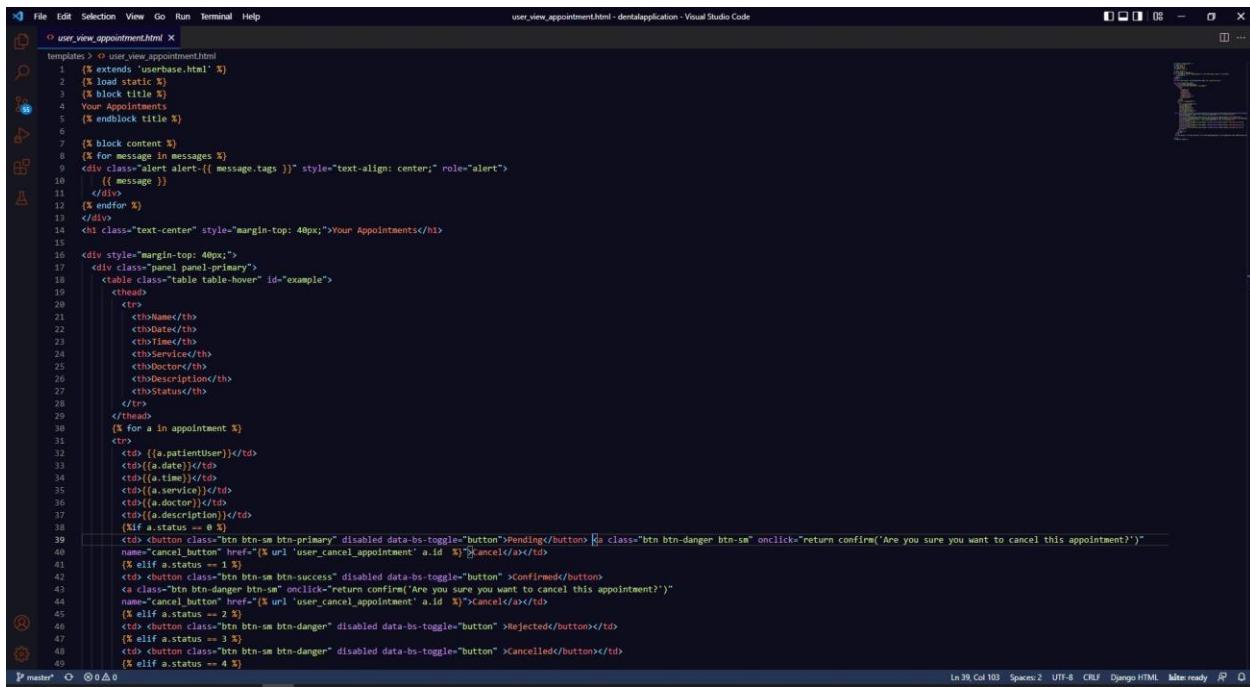


The screenshot shows a Visual Studio Code window with the file "userdashboard.html" open. The code is a Django template for a user dashboard. It includes sections for finished appointments, pending appointments, and upcoming appointments, each with links to "Book New", "View", and "Edit". A table lists all appointments with columns for Name, Date, Time, Doctor, Description, and Status, plus a "Confirmed" button. The code uses Bootstrap classes like card, card-body, card-subtitle, and card-title, and Django template tags like {{}} and {% %}. The status bar at the bottom indicates the file is 91 lines long, in UTF-8 encoding, and is a Django HTML file.

```
File Edit Selection View Go Run Terminal Help
templates > userdashboard.html
1   <div class="card" style="margin-bottom: 40px;">
2     <div class="card-header">
3       <h3 class="card-title">{{finishedappointmentcount}}
4       <a href="/user_book_appointment" class="btn btn-outline-secondary">Book New</a>
5     </div>
6   </div>
7   <div class="col" style="margin-bottom: 40px;">
8     <div class="card bg-light border-dark" style="width: 10rem;">
9       <div class="card-body" style="text-align: center;">
10      <h6 class="card-subtitle mb-2 text-muted">Pending Appointments</h6>
11      <div class="card-title">{{pendingappointmentcount}}View</a>
13    </div>
14  </div>
15 </div>
16 <div style="margin-left:1%; ">
17   <div class="panel-heading" style="text-align:center">
18     <h4 class="panel-title">Upcoming Appointments</h4>
19   </div>
20   <table class="table table-bordered">
21     <thead >
22       <tr>
23         <th>Name</th>
24         <th>Date</th>
25         <th>Time</th>
26         <th>Doctor</th>
27         <th>Description</th>
28         <th>Status</th>
29       </tr>
30     </thead>
31     <% for a in appointment %>
32     <tr>
33       <td>{{a.patientUser}}</td>
34       <td>{{a.date}}</td>
35       <td>{{a.time}}</td>
36       <td>{{a.service}}</td>
37       <td>{{a.doctor}}</td>
38       <td>{{a.description}}</td>
39       <td><button class="btn btn-sm btn-warning" disabled data-bs-toggle="button" >confirmed</button></td>
40     </tr>
41     <% endfor %>
42   </table>
43 </div>
44 </div>
45 <% endblock content %>
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
```

Figure 245: Sample Code of UI - 21

User_view_appointment.html



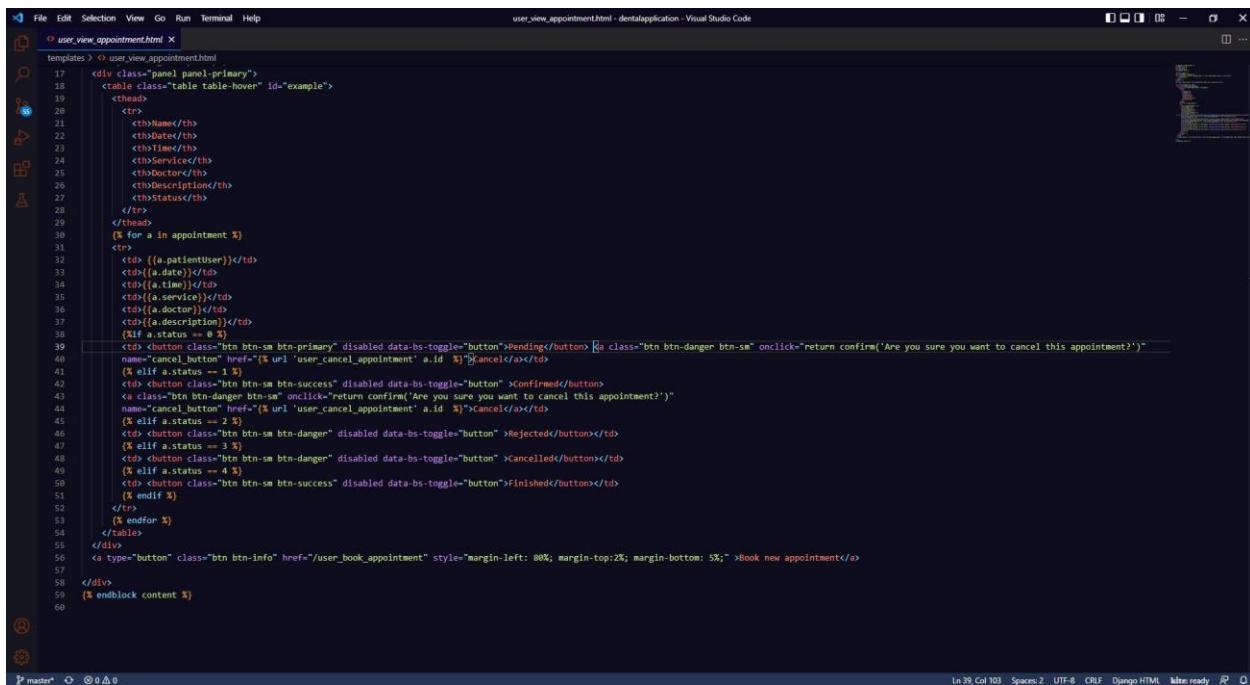
```

File Edit Selection View Go Run Terminal Help
templates > user_view_appointment.html - dentalapplication - Visual Studio Code
1  {% extends 'userbase.html' %}
2  {% load static %}
3  {% block title %}
4  Your Appointments
5  {% endblock title %}
6
7  {% block content %}
8  {% for message in messages %}
9  <div class="alert alert-{{ message.tags }} style="text-align: center;" role="alert">
10   {{ message }}
11  </div>
12  {% endif %}
13 </div>
14 <h1 class="text-center" style="margin-top: 40px;">Your Appointments</h1>
15 <div style="margin-top: 40px;">
16  <div class="panel panel-primary">
17   <table class="table table-hover" id="example">
18    <thead>
19     <tr>
20      <th>Name</th>
21      <th>Date</th>
22      <th>Time</th>
23      <th>Service</th>
24      <th>Doctor</th>
25      <th>Description</th>
26      <th>Status</th>
27     </tr>
28    </thead>
29    <tbody>
30      {% for a in appointment %}
31      <tr>
32        <td>{{a.patientUser}}</td>
33        <td>{{a.date}}</td>
34        <td>{{a.time}}</td>
35        <td>{{a.service}}</td>
36        <td>{{a.doctor}}</td>
37        <td>{{a.description}}</td>
38        {% if a.status == 0 %}
39          <td><button class="btn btn-sm btn-primary" disabled data-bs-toggle="button">Pending</button> <a class="btn btn-danger btn-sm" onclick="return confirm('Are you sure you want to cancel this appointment?') name='cancel_button' href='{% url 'user_cancel_appointment' a.id %}'>Cancel</a></td>
40        {% elif a.status == 1 %}
41          <td><button class="btn btn-sm btn-success" disabled data-bs-toggle="button">Confirmed</button> <a class="btn btn-danger btn-sm" onclick="return confirm('Are you sure you want to cancel this appointment?') name='cancel_button' href='{% url 'user_cancel_appointment' a.id %}'>Cancel</a></td>
42        {% elif a.status == 2 %}
43          <td><button class="btn btn-sm btn-danger" disabled data-bs-toggle="button">Rejected</button></td>
44        {% elif a.status == 3 %}
45          <td><button class="btn btn-sm btn-danger" disabled data-bs-toggle="button">Cancelled</button></td>
46        {% elif a.status == 4 %}
47
48
49

```

In 39 Col 103 Spaces: 2 UTF-8 CR LF Django HTML later ready

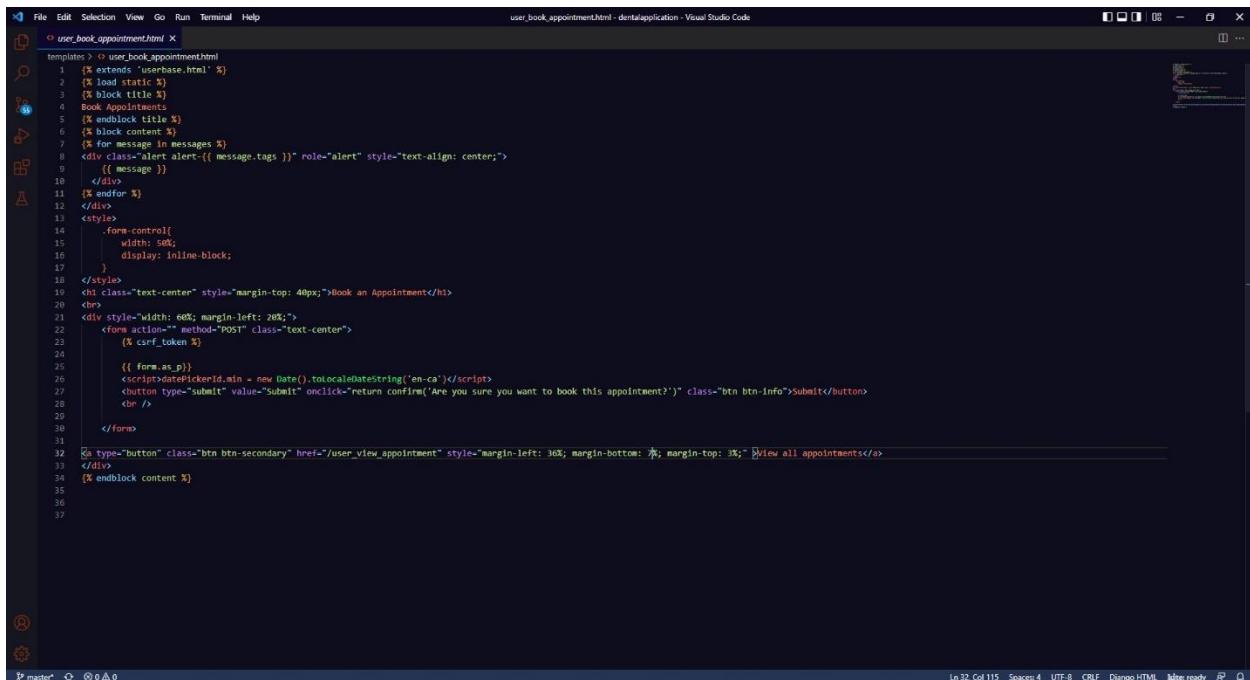
Figure 246: Sample Code of UI - 22



The screenshot shows a Visual Studio Code window with the file "user_view_appointment.html" open. The code is a Django template for displaying a list of appointments. It uses a table with columns for Name, Date, Time, Services, Doctor, Description, and Status. For each appointment, there is a "Pending" button (disabled) and a "Cancel" button (danger class, onclick="return confirm('Are you sure you want to cancel this appointment?')"). If the status is 1, there is also a "Confirmed" button (success class, disabled). If the status is 2, there is a "Rejected" button (danger class, disabled). If the status is 3, there is a "Cancelled" button (danger class, disabled). If the status is 4, there is a "Finished" button (success class, disabled). A "Book new appointment" button is at the bottom.

```
File Edit Selection View Go Run Terminal Help
templates > user_view_appointment.html
17  <div class="panel panel-primary">
18    <table class="table table-hover" id="example">
19      <thead>
20        <tr>
21          <th>Name</th>
22          <th>Date</th>
23          <th>Time</th>
24          <th>Services</th>
25          <th>Doctor</th>
26          <th>Description</th>
27          <th>Status</th>
28        </tr>
29      </thead>
30      <% for a in appointment %>
31        <tr>
32          <td>{{a.patientUser}}</td>
33          <td>{{a.date}}</td>
34          <td>{{a.time}}</td>
35          <td>{{a.services}}</td>
36          <td>{{a.doctor}}</td>
37          <td>{{a.description}}</td>
38          <%if a.status == 0 %>
39            <td><button class="btn btn-sm btn-primary" disabled data-bs-toggle="button">Pending</button> <a class="btn btn-danger btn-sm" onclick="return confirm('Are you sure you want to cancel this appointment?')" name="cancel_button" href="{% url 'user_cancel_appointment' a.id %}">Cancel</a></td>
40          <% elif a.status == 1 %>
41            <td><button class="btn btn-sm btn-success" disabled data-bs-toggle="button">Confirmed</button>
42            <a class="btn btn-danger btn-sm" onclick="return confirm('Are you sure you want to cancel this appointment?')" name="cancel_button" href="{% url 'user_cancel_appointment' a.id %}">Cancel</a></td>
43          <% elif a.status == 2 %>
44            <td><button class="btn btn-sm btn-danger" disabled data-bs-toggle="button">Rejected</button></td>
45          <% elif a.status == 3 %>
46            <td><button class="btn btn-sm btn-danger" disabled data-bs-toggle="button">Cancelled</button></td>
47          <% elif a.status == 4 %>
48            <td><button class="btn btn-sm btn-success" disabled data-bs-toggle="button">Finished</button></td>
49          <% endif %>
50        </td>
51      <% endfor %>
52    </table>
53  </div>
54  <a type="button" class="btn btn-info" href="/user_book_appointment" style="margin-left: 80px; margin-top: 2%; margin-bottom: 5%;">Book new appointment</a>
55
56 <% endblock content %>
57
58
59
```

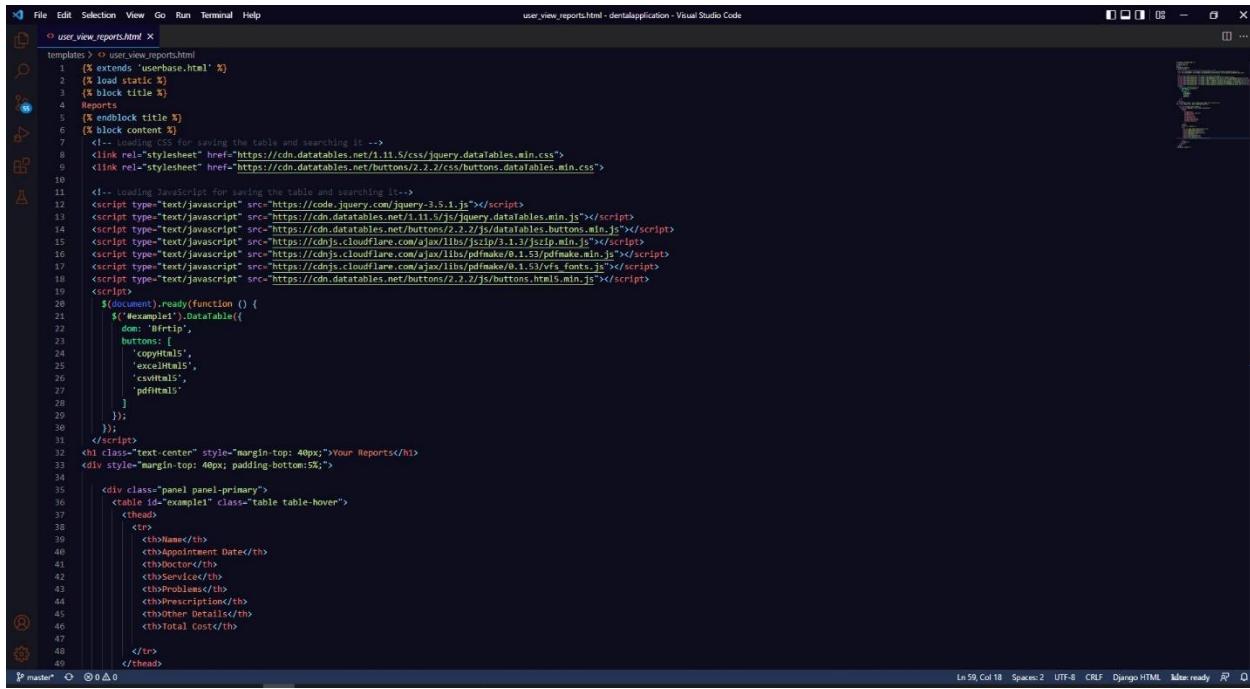
Figure 247: Sample Code of UI - 23

User_book_appointment.html

The screenshot shows a Visual Studio Code window with the file "user_book_appointment.html" open. The code is a Django template for booking an appointment. It includes a header section with a title and a message block for displaying alerts. Below this is a form for booking an appointment, which includes a date picker, a submit button with a confirmation message, and a link to view all appointments. The code uses Bootstrap classes like .form-control and .text-center.

```
File Edit Selection View Go Run Terminal Help
templates > user_book_appointment.html
1  {% extends 'userbase.html' %} 
2  {% load static %} 
3  {% block title %} 
4  Book Appointments 
5  {% endblock title %} 
6  {% block content %} 
7  {% for message in messages %} 
8  <div class="alert alert-{{ message.tags }}" role="alert" style="text-align: center;">> 
9  {{ message }} 
10 </div> 
11 {% endfor %} 
12 </div> 
13 <style> 
14   .form-control{ 
15     width: 50%; 
16     display: inline-block; 
17   } 
18 </style> 
19 <h1 class="text-center" style="margin-top: 40px;">Book an Appointment</h1> 
20 <br> 
21 <div style="width: 60%; margin-left: 20%;> 
22   <form action="" method="POST" class="text-center"> 
23     {% csrf_token %} 
24     {{ form.as_p }} 
25     <script>datepickrId.min = new Date().toLocaleDateString('en-ca')</script> 
26     <button type="submit" value="Submit" onclick="return confirm('Are you sure you want to book this appointment?')" class="btn btn-info">Submit</button> 
27     <br /> 
28   </form> 
29   <a href="/user_view_appointment" style="margin-left: 30%; margin-bottom: 10px; margin-top: 10px;">View all appointments</a> 
30 </div> 
31 {% endblock content %} 
32 
33 
34 
35 
36 
37
```

Figure 248: Sample Code of UI - 24

User_view_report.html


```

File Edit Selection View Go Run Terminal Help
templates > user_view_reports.html
1  {% extends 'userbase.html' %} 
2  {% load static %} 
3  {% block title %} 
4  Reports 
5  {% endblock title %} 
6  {% block content %} 
7  <!-- Loading CSS for saving the table and searching it-->
8  <link rel="stylesheet" href="https://cdn.datatables.net/1.11.5/css/jquery.dataTables.min.css">
9  <link rel="stylesheet" href="https://cdn.datatables.net/buttons/2.2.2/css/buttons.dataTables.min.css">
10 <!-- Loading JavaScript for saving the table and searching it-->
11 <script type="text/javascript" src="https://code.jquery.com/jquery-3.5.1.js"></script>
12 <script type="text/javascript" src="https://cdn.datatables.net/1.11.5/js/jquery.dataTables.min.js"></script>
13 <script type="text/javascript" src="https://cdn.datatables.net/buttons/2.2.2/js/dataTables.buttons.min.js"></script>
14 <script type="text/javascript" src="https://cdnjs.cloudflare.com/ajax/libs/jszip/3.1.3/jszip.min.js"></script>
15 <script type="text/javascript" src="https://cdnjs.cloudflare.com/ajax/libs/pdfmake/0.1.53/pdfmake.min.js"></script>
16 <script type="text/javascript" src="https://cdnjs.cloudflare.com/ajax/libs/html2pdf.js/0.1.5/html2pdf.min.js"></script>
17 <script type="text/javascript" src="https://cdn.datatables.net/buttons/2.2.2/js/buttons.html5.min.js"></script>
18 <script type="text/javascript" src="https://cdn.datatables.net/buttons/2.2.2/js/buttons.html5.min.js"></script>
19 <script type="text/javascript" src="https://cdn.datatables.net/buttons/2.2.2/js/buttons.html5.min.js"></script>
20 <script type="text/javascript" src="https://cdn.datatables.net/buttons/2.2.2/js/buttons.html5.min.js"></script>
21 $(document).ready(function () {
22   $('#example').DataTable({
23     dom: 'Bfrtip',
24     buttons: [
25       'copy',
26       'excel',
27       'pdf',
28       'print',
29     ]
30   });
31 });
32 <h1 class="text-center" style="margin-top: 40px;">Your Reports
33 <div style="margin-top: 40px; padding-bottom: 5px;">
34 
35   <div class="panel panel-primary">
36     <table id="example1" class="table table-hover">
37       <thead>
38         <tr>
39           <th>Name</th>
40           <th>Appointment Date</th>
41           <th>Doctor ID</th>
42           <th>Gender</th>
43           <th>Problems</th>
44           <th>Prescription</th>
45           <th>Other Details</th>
46           <th>Total Cost</th>
47         </tr>
48       </thead>
49     </table>
50   </div>
51 
```

Figure 249: Sample Code of UI – 25

```
File Edit Selection View Go Run Terminal Help
templates > user_view_reports.html
  1 <button type="button" onclick="location.href='{% url 'user_view_reports' %}';">Home</button>
  2 <button type="button" onclick="location.href='{% url 'user_view_reports_csv' %}';">CSV</button>
  3 <button type="button" onclick="location.href='{% url 'user_view_reports_excel' %}';">Excel</button>
  4 <button type="button" onclick="location.href='{% url 'user_view_reports_pdf' %}';">PDF</button>
  5 </div>
  6 </div>
  7 </div>
  8 </div>
  9 </div>
 10 </div>
 11 </div>
 12 <div style="text-align: center; margin-top: 40px;">Your Reports</div>
 13 <div style="margin-top: 40px; padding-bottom: 5px; border-bottom: 1px solid black; position: relative; height: 10px; width: 100%;">
 14 <div class="panel panel-primary">
 15   <div class="panel-heading">View Reports</div>
 16   <div class="panel-body">
 17     <table id="example1" class="table table-hover">
 18       <thead>
 19         <tr>
 20           <th>Patient Name</th>
 21           <th>Appointment Date</th>
 22           <th>Doctor</th>
 23           <th>Services</th>
 24           <th>Problems</th>
 25           <th>Prescription</th>
 26           <th>Other Details</th>
 27           <th>Total Cost</th>
 28         </tr>
 29       </thead>
 30       <tbody>
 31         <% for r in reports %>
 32           <tr>
 33             <td>{{r.appointment.patientuser}}</td>
 34             <td>{{r.appointment.date}}</td>
 35             <td>{{r.appointment.doctor}}</td>
 36             <td>{{r.appointment.service}}</td>
 37             <td>{{r.problems}}</td>
 38             <td>{{r.prescription}}</td>
 39             <td>{{r.other_details}}</td>
 40             <td>Rs.{{r.appointment.service.cost}}</td>
 41           </tr>
 42         <% endfor %>
 43       </tbody>
 44     </table>
 45   </div>
 46 <% endblock content %>
 47 </div>
 48 </div>
 49 </div>
 50 </div>
 51 </div>
 52 </div>
 53 </div>
 54 </div>
 55 </div>
 56 </div>
 57 </div>
 58 </div>
 59 </div>
 60 </div>
 61 </div>
 62 </div>
 63 </div>
 64 </div>
 65 </div>
 66 </div>
 67 </div>
 68 </div>
```

Figure 250: Sample Code of UI - 26

User_view_services.html

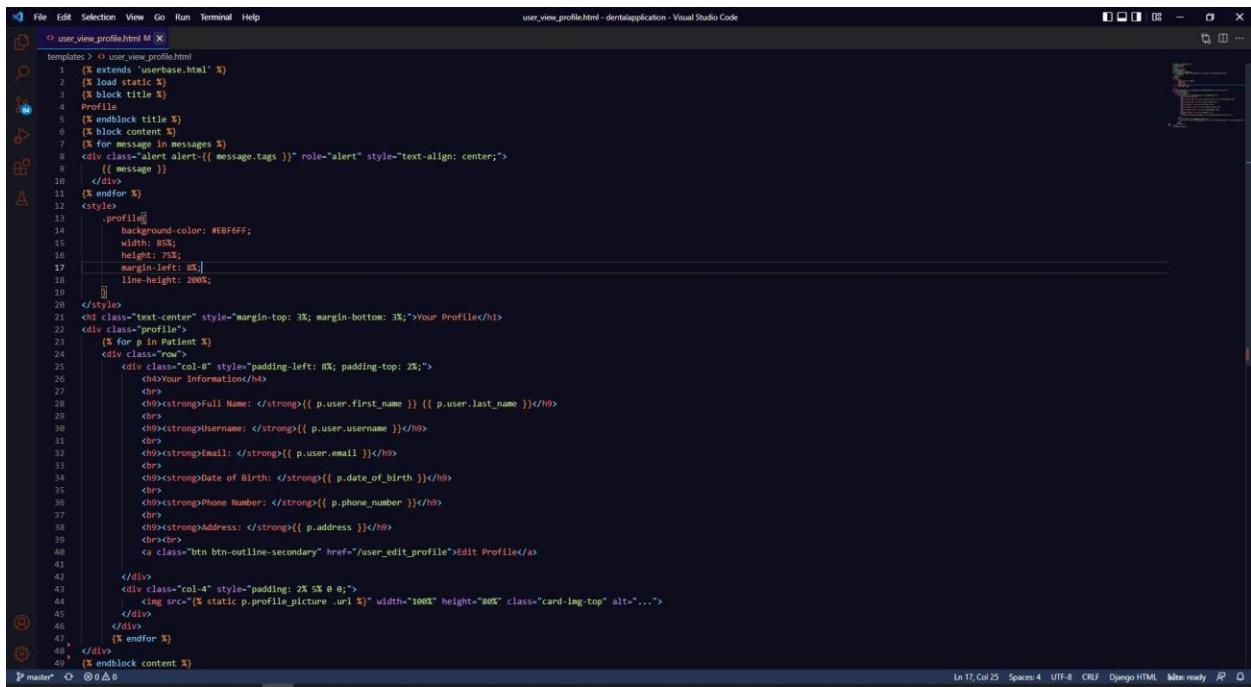
```
File Edit Selection View Go Run Terminal Help
templates > user_view_services.html
1  {% extends 'userbase.html' %} 
2  {% load static %} 
3  {% block title %} 
4  Services
5  {% endblock content %} 
6  <h1 class="text-center" style="margin-top: 40px;">Our Services</h1>
7  <div style="margin-top: 40px;">
8  <div>
9    <table id="example" class="table table-hover">
10      <thead>
11        <tr>
12          <th>Name</th>
13          <th>Cost</th>
14          <th>Description</th>
15        </tr>
16      </thead>
17      <tbody>
18        {% for s in services %}
19          <tr>
20            <td>{{s.service_name}}</td>
21            <td>{{s.cost}}</td>
22            <td>{{s.description}}</td>
23          </tr>
24        {% endfor %}
25      </tbody>
26    </table>
27  </div>
28  </div>
29  {% endblock content %}
```

Figure 251: Sample Code of UI - 27

User_view_doctors.html

```
File Edit Selection View Go Run Terminal Help
templates > user_view_doctors.html
1  {% extends 'userbase.html' %} 
2  {% load static %} 
3  {% block title %} 
4  Doctors 
5  {% endblock title %} 
6  {% block content %} 
7  <h1 class="text-center" style="margin-top: 40px; margin-bottom: 40px;">Our Doctors</h1>
8  <div class="panel panel-primary">
9
10   <div class="container">
11    <div class="row">
12      {% for d in doctors %}<br>
13      <div class="col" style="margin-bottom: 40px; margin-left: 20px;">
14
15        <div class="card text-dark bg-light mb-3" style="width: 18rem;">
16
17          
18          <div class="card-body">
19
20            <h3 class="card-title">Dr. {{d.full_name}}</h3>
21          </div>
22          <ul class="list-group list-group-flush">
23            <li class="list-group-item">Qualification: {{d.qualification}}</li>
24            <li class="list-group-item">Experience: {{d.experience}}</li>
25            <li class="list-group-item">Phone Number: {{d.phone_number}}</li>
26            <li class="list-group-item">Email: {{d.email}}</li>
27          </ul>
28
29          <div class="card-body" style="text-align: center;">
30            <a href="/user_book_appointment" class="btn btn-outline-secondary">Book an Appointment now.</a>
31          </div>
32
33        </div>
34      {% endfor %}
35
36    </div>
37  </div>
38  {% endblock content %}
```

Figure 252: Sample Code of UI - 28

User_view_profile.html


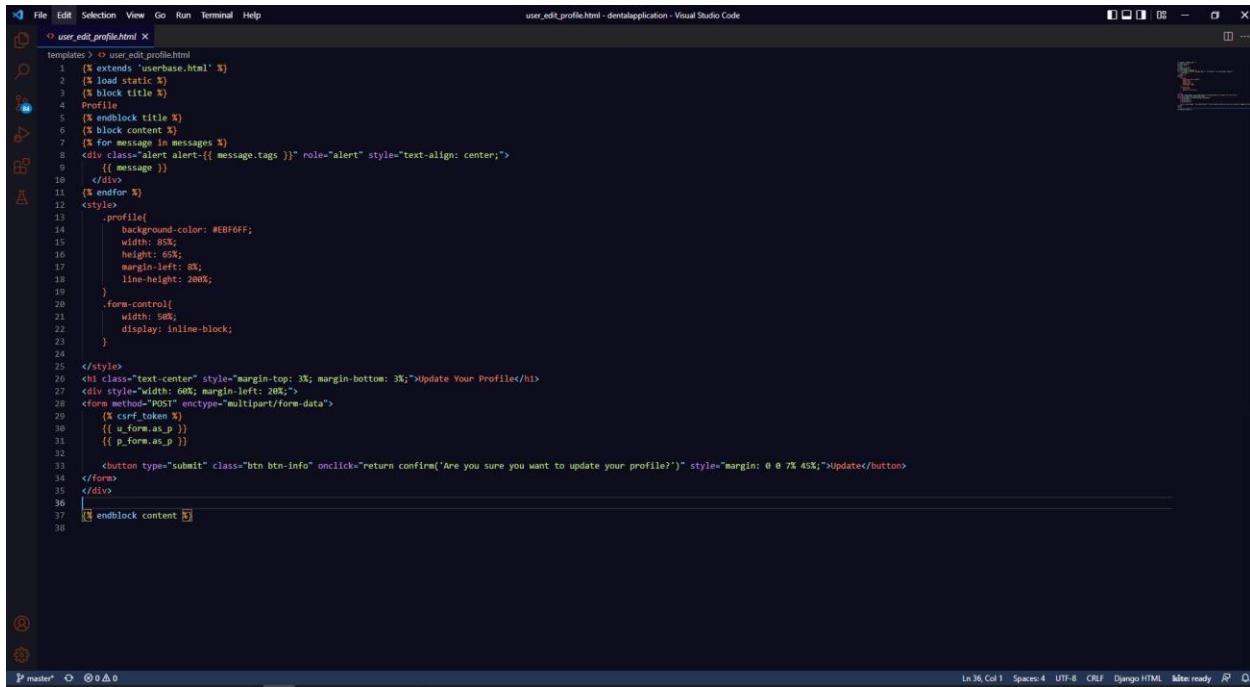
The screenshot shows the Visual Studio Code interface with the file 'user_view_profile.html' open. The code is a Django template for viewing a user's profile. It includes HTML, CSS, and Django templating syntax. The code displays a profile card with patient information and a preview image.

```

File Edit Selection View Go Run Terminal Help
templates > O user_view_profile.html M X
user_view_profile.html - dentalapplication - Visual Studio Code
1  {% extends 'userbase.html' %} 
2  {% load static %} 
3  {% block title %} 
4  Profile 
5  {% endblock title %} 
6  {% block content %} 
7  {% for message in messages %} 
8  <div><div>{{ message }}</div><div>{{ message }}</div>
9  </div> 
10 <% endfor %> 
11 <style> 
12 .profile{ 
13   background-color: #E8F0FF; 
14   width: 85%; 
15   height: 75%; 
16   margin-left: auto; 
17   margin-right: auto; 
18   line-height: 200%; 
19 } 
20 </style> 
21 <h1 class="text-center" style="margin-top: 3%; margin-bottom: 3%;">Your Profile

```

Figure 253: Sample Code of UI - 29

User_edit_profile.html

```
File Edit Selection View Go Run Terminal Help
templates > user_edit_profile.html - dentalapplication - Visual Studio Code
1  {% extends 'base.html' %}
2  {% load static %}
3  {% block title %}
4  Profile
5  {% endblock title %}
6  {% block content %}
7  {% for message in messages %}
8  <div class="alert alert-{% message.tags %}" role="alert" style="text-align: center;">
9      {{ message }}
10 
```

Figure 254: Sample Code of UI - 30

7.5.1.2 Admin Side

Adminbase.html

```

File Edit Selection View Go Run Terminal Help
templates > adminbase.html
adminbase.html
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <!-- Required meta tags -->
6   <meta name="author" content="Django-HTML">
7   <meta name="author" content="Abhishek Rokaya">
8   <% load static %>
9   <link rel="stylesheet" href="{% static 'css/dashboardstyle.css' %}">
10  <meta charset="utf-8">
11  <meta name="viewport" content="width=device-width, initial-scale=1">
12
13  <!-- Bootstrap CSS -->
14  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
15  Integrity="sha384-1Bmc4R4Nq7aYhdIdvKmfTAQduu08T9wHwfJ0nCEXSUloDwq1Qo7GJIM3" crossorigin="anonymous">
16
17  <!-- loading CSS for saving the table and searching it -->
18  <link rel="stylesheet" href="https://cdn.datatables.net/1.11.5/css/jquery.dataTables.min.css">
19  <link rel="stylesheet" href="https://cdn.datatables.net/buttons/2.2.2/css/buttons.dataTables.min.css">
20
21  <!-- loading JavaScript for saving the table and searching it -->
22  <script type="text/javascript" src="https://code.jquery.com/jquery-3.5.1.js"></script>
23  <script type="text/javascript" src="https://cdn.datatables.net/1.11.5/js/jquery.dataTables.min.js"></script>
24  <script type="text/javascript" src="https://cdn.datatables.net/buttons/2.2.2/js/dataTables.buttons.min.js"></script>
25  <script type="text/javascript" src="https://cdnjs.cloudflare.com/ajax/libs/jszip/3.1.3/jszip.min.js"></script>
26  <script type="text/javascript" src="https://cdnjs.cloudflare.com/ajax/libs/pdfmake/0.1.53/pdfmake.min.js"></script>
27  <script type="text/javascript" src="https://cdnjs.cloudflare.com/ajax/libs/pdfmake/0.1.53/vfs_fonts.js"></script>
28  <script type="text/javascript" src="https://cdn.datatables.net/buttons/2.2.2/js/buttons.html5.min.js"></script>
29  <script>
30
31  $(document).ready(function () {
32    $('#example').DataTable({
33      dom: 'Bfrtip',
34      buttons: [
35        'copyHtml5',
36        'excelHtml5',
37        'csvHtml5',
38        'pdfHtml5'
39      ]
40    });
41  });
42
43  <title>[<% block title %>](<% endblock title %>) | Phulchowki Dental Care </title>
44
45
46 </head>
47
48 <body>
49   <nav class="menu" tabindex="0">

```

In 12, Col 1 Spaces: 2 UFT-8 CRLF Django HTML line ready

Figure 255: Sample Code of UI - 31

```

File Edit Selection View Go Run Terminal Help
adminbase.html X
templates > adminbase.html
 41 </script>
 42 
 43 <title>{% block title %}{% endblock title %} | Phulchowki Dental Care </title>
 44 
 45 
 46 </head>
 47 
 48 <body>
 49   <nav class="menu" tabindex="0">
 50     <div class="smartphone-menu-trigger"></div>
 51     <header class="avatar" style="margin-top: -5px; height: 35px;">
 52       
 53     </header>
 54     <div style="color: black; ">Admin</div>
 55   </header>
 56 
 57   <ul style="margin-top: 10px;">
 58     <li tabindex="0" class="icon-dashboard"> <a style="color: black; text-decoration: none;" href="/admindashboard">spanDashboard</span></a></li>
 59     <li tabindex="0" class="icon-appointment"> <a style="color: black; text-decoration: none;" href="/admin_appointment">spanAppointment</span></a></li>
 60     <li tabindex="0" class="icon-doctor"> <a style="color: black; text-decoration: none;" href="/admin_view_doctors">spanManage Doctors</span></a></li>
 61     <li tabindex="0" class="icon-report"> <a style="color: black; text-decoration: none;" href="/admin_view_reports">spanManage Reports</span></a></li>
 62     <li tabindex="0" class="icon-service"> <a style="color: black; text-decoration: none;" href="/admin_view_services">spanManage Services</span></a></li>
 63     <li tabindex="0" class="icon-profile"> <a style="color: black; text-decoration: none;" href="/admin_view_users">spanManage User Profiles</span></a></li>
 64   </ul>
 65 </nav>
 66 <main>
 67   <!-- nav start -->
 68   <div class="bs-example">
 69     <nav class="navbar navbar-expand-md navbar-dark fixed-top" style="background: #2677C2;">
 70       <a href="/admindashboard" class="navbar-brand" style="margin-left: 20px;">Phulchowki Dental Clinic</a>
 71       <button type="button" class="navbar-toggler" data-toggle="collapse" data-target="#navbarCollapse">
 72         <span class="navbar-toggler-icon"></span>
 73       </button>
 74       <div class="collapse navbar-collapse justify-content-between" id="navbarCollapse">
 75         <div class="navbar-nav" style="margin-left: 90%; ">
 76           <a type="button" onclick="return confirm('Are you sure you want to logout?')> class="btn" href="/logout" style="border-color: black; ">Logout</a>
 77         </div>
 78       </div>
 79     </nav>
 80   <!-- nav end -->
 81   <br>
 82 </main>
 83 </div>
 84 <!-- content starts -->
 85 <!-- content ends -->
 86 <!-- footer -->
 87 <footer style="position: fixed; left: 0; bottom: 0; width: 100%;">
 88   <div class="container">
 89     <div class="row">
 90       <div class="col-md-12 col-sm-12">
 91         <div style="color: #ffffff; padding-left: 20px;" class="wow fadeInUp footer-copyright">
 92           <p>Copyright © 2022 Phulchowki Dental Clinic</p>
 93         </div>
 94       </div>
 95     </div>
 96   </div>
 97 </footer>
 98 </body>
 99 </html>

```

Line 12, Col 1 Spaces: 2 UTF-8 CRLF Django HTML file ready

Figure 256: Sample Code of UI - 32

```

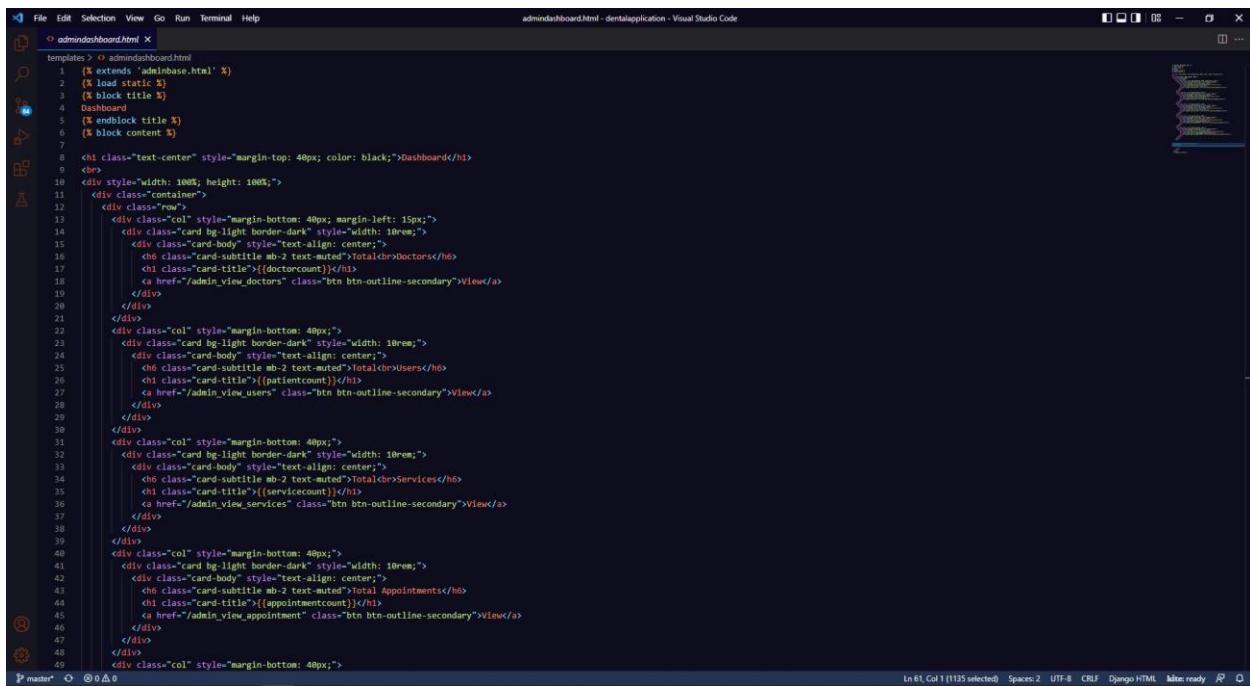
File Edit Selection View Go Run Terminal Help
adminbase.html X
templates > adminbase.html
 41 </script>
 42 
 43 <title>{% block title %}{% endblock title %} | Phulchowki Dental Care </title>
 44 
 45 
 46 </head>
 47 
 48 <body>
 49   <nav class="menu" tabindex="0">
 50     <div class="smartphone-menu-trigger"></div>
 51     <header class="avatar" style="margin-top: -5px; height: 35px;">
 52       
 53     </header>
 54     <div style="color: black; ">Admin</div>
 55   </header>
 56 
 57   <ul style="margin-top: 10px;">
 58     <li tabindex="0" class="icon-dashboard"> <a style="color: black; text-decoration: none;" href="/admindashboard">spanDashboard</span></a></li>
 59     <li tabindex="0" class="icon-appointment"> <a style="color: black; text-decoration: none;" href="/admin_appointment">spanAppointment</span></a></li>
 60     <li tabindex="0" class="icon-doctor"> <a style="color: black; text-decoration: none;" href="/admin_view_doctors">spanManage Doctors</span></a></li>
 61     <li tabindex="0" class="icon-report"> <a style="color: black; text-decoration: none;" href="/admin_view_reports">spanManage Reports</span></a></li>
 62     <li tabindex="0" class="icon-service"> <a style="color: black; text-decoration: none;" href="/admin_view_services">spanManage Services</span></a></li>
 63     <li tabindex="0" class="icon-profile"> <a style="color: black; text-decoration: none;" href="/admin_view_users">spanManage User Profiles</span></a></li>
 64   </ul>
 65 </nav>
 66 <main>
 67   <!-- nav start -->
 68   <div class="bs-example">
 69     <nav class="navbar navbar-expand-md navbar-dark fixed-top" style="background: #2677C2;">
 70       <a href="/admindashboard" class="navbar-brand" style="margin-left: 20px;">Phulchowki Dental Clinic</a>
 71       <button type="button" class="navbar-toggler" data-toggle="collapse" data-target="#navbarCollapse">
 72         <span class="navbar-toggler-icon"></span>
 73       </button>
 74       <div class="collapse navbar-collapse justify-content-between" id="navbarCollapse">
 75         <div class="navbar-nav" style="margin-left: 90%; ">
 76           <a type="button" onclick="return confirm('Are you sure you want to logout?')> class="btn" href="/logout" style="border-color: black; ">Logout</a>
 77         </div>
 78       </div>
 79     </nav>
 80   <!-- nav end -->
 81   <br>
 82 </main>
 83 </div>
 84 <!-- content starts -->
 85 <!-- content ends -->
 86 <!-- footer -->
 87 <footer style="position: fixed; left: 0; bottom: 0; width: 100%;">
 88   <div class="container">
 89     <div class="row">
 90       <div class="col-md-12 col-sm-12">
 91         <div style="color: #ffffff; padding-left: 20px;" class="wow fadeInUp footer-copyright">
 92           <p>Copyright © 2022 Phulchowki Dental Clinic</p>
 93         </div>
 94       </div>
 95     </div>
 96   </div>
 97 </footer>
 98 </body>
 99 </html>

```

Line 12, Col 1 Spaces: 2 UTF-8 CRLF Django HTML file ready

Figure 257: Sample Code of UI - 33

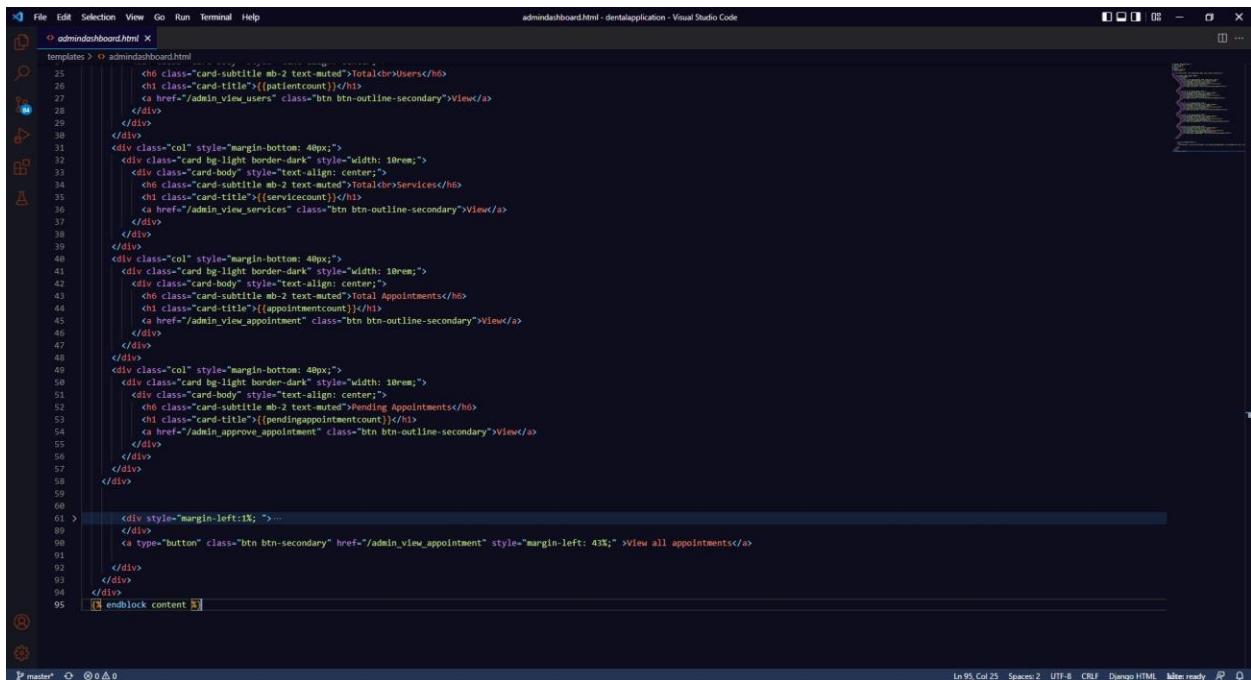
Admindashboard.html



The screenshot shows a Visual Studio Code window with the file "admindashboard.html" open. The code is a Django template for an admin dashboard. It extends "adminbase.html" and contains a main title "Dashboard". The page is divided into four columns using a grid system. Each column contains a card with a title, subtitle, and a "View" button. The first column displays statistics for Doctors (Total: {{doctorcount}}) and Patients (Total: {{patientcount}}). The second column displays statistics for Services (Total: {{servicecount}}). The third column displays statistics for Appointments (Total Appointments: {{appointmentcount}}). The code uses Bootstrap classes like "col", "card", "card-body", and "card-subtitle" along with Django's {{ }} syntax for variable substitution.

```
File Edit Selection View Go Run Terminal Help
templates > admindashboard.html - dentalapplication - Visual Studio Code
admindashboard.html
1  {% extends 'adminbase.html' %} 
2  {% load static %} 
3  {% block title %} 
4  Dashboard 
5  {% endblock title %} 
6  {% block content %} 
7  <h1 class="text-center" style="margin-top: 40px; color: black;">Dashboard</h1>
8  <br>
9  <div style="width: 100%; height: 100%;>
10 <div class="container">
11   <div class="row">
12     <div class="col" style="margin-bottom: 40px; margin-left: 15px;">
13       <div class="card bg-light border-dark" style="width: 10rem;">
14         <div class="card-body" style="text-align: center;">
15           <h6 class="card-subtitle mb-2 text-muted">Total Doctors</h6>
16           <h1 class="card-title">{{doctorcount}}</h1>
17           <a href="/admin_view_doctors" class="btn btn-outline-secondary">View</a>
18         </div>
19       </div>
20     </div>
21     <div class="col" style="margin-bottom: 40px;">
22       <div class="card bg-light border-dark" style="width: 10rem;">
23         <div class="card-body" style="text-align: center;">
24           <h6 class="card-subtitle mb-2 text-muted">Total Users</h6>
25           <h1 class="card-title">{{patientcount}}</h1>
26           <a href="/admin_view_users" class="btn btn-outline-secondary">View</a>
27         </div>
28       </div>
29     </div>
30     <div class="col" style="margin-bottom: 40px;">
31       <div class="card bg-light border-dark" style="width: 10rem;">
32         <div class="card-body" style="text-align: center;">
33           <h6 class="card-subtitle mb-2 text-muted">Total Services</h6>
34           <h1 class="card-title">{{servicecount}}</h1>
35           <a href="/admin_view_services" class="btn btn-outline-secondary">View</a>
36         </div>
37       </div>
38     </div>
39   </div>
40   <div class="col" style="margin-bottom: 40px;">
41     <div class="card bg-light border-dark" style="width: 10rem;">
42       <div class="card-body" style="text-align: center;">
43         <h6 class="card-subtitle mb-2 text-muted">Total Appointments</h6>
44         <h1 class="card-title">{{appointmentcount}}</h1>
45         <a href="/admin_view_appointment" class="btn btn-outline-secondary">View</a>
46       </div>
47     </div>
48   </div>
49 </div class="col" style="margin-bottom: 40px;">
```

Figure 258: Sample Code of UI – 34



```
File Edit Selection View Go Run Terminal Help
admindashboard.html - dentalapplication - Visual Studio Code
templates > admindashboard.html
25     <div class="card-subtitle mb-2 text-muted">Total Users</div>
26     <div class="card-title">{{patientcount}}</div>
27     <a href="/admin_view_users" class="btn btn-outline-secondary">View</a>
28   </div>
29 </div>
30 <div class="col" style="margin-bottom: 40px;">
31   <div class="card bg-light border-dark" style="width: 10rem;">
32     <div class="card-body" style="text-align: center;">
33       <div class="card-subtitle mb-2 text-muted">Total Services</div>
34       <div class="card-title">{{servicecount}}</div>
35       <a href="/admin_view_services" class="btn btn-outline-secondary">View</a>
36     </div>
37   </div>
38 </div>
39 <div class="col" style="margin-bottom: 40px;">
40   <div class="card bg-light border-dark" style="width: 10rem;">
41     <div class="card-body" style="text-align: center;">
42       <div class="card-subtitle mb-2 text-muted">Total Appointments</div>
43       <div class="card-title">{{appointmentcount}}</div>
44       <a href="/admin_view_appointment" class="btn btn-outline-secondary">View</a>
45     </div>
46   </div>
47 </div>
48 <div class="col" style="margin-bottom: 40px;">
49   <div class="card bg-light border-dark" style="width: 10rem;">
50     <div class="card-body" style="text-align: center;">
51       <div class="card-subtitle mb-2 text-muted">Pending Appointments</div>
52       <div class="card-title">{{pendingappointmentcount}}</div>
53       <a href="/admin_approve_appointment" class="btn btn-outline-secondary">View</a>
54     </div>
55   </div>
56 </div>
57 </div>
58 </div>
59 </div>
60 </div>
61 >   <div style="margin-left:15px;">...
62     <a type="button" class="btn btn-secondary" href="/admin_view_appointment" style="margin-left: 45px;">View all appointments</a>
63   </div>
64 </div>
65 </div>
66 <div style="margin-left:15px;">...
67   <a type="button" class="btn btn-secondary" href="/admin_view_appointment" style="margin-left: 45px;">View all appointments</a>
68 </div>
69 </div>
70 </div>
71 </div>
72 </div>
73 </div>
74 </div>
75 </div>
76 </div>
77 </div>
78 </div>
79 </div>
80 </div>
81 </div>
82 </div>
83 </div>
84 </div>
85 </div>
86 </div>
87 </div>
88 </div>
89 </div>
90 </div>
91 </div>
92 </div>
93 </div>
94 </div>
95 </div>
96 <div>
97   <div>
98     <div>
99       <div>
100         <div>
101           <div>
102             <div>
103               <div>
104                 <div>
105                   <div>
106                     <div>
107                       <div>
108                         <div>
109                           <div>
110                             <div>
111                               <div>
112                                 <div>
113                                   <div>
114                                     <div>
115                                       <div>
116                                         <div>
117                                           <div>
118                                             <div>
119                                               <div>
120                                                 <div>
121                                                   <div>
122                                                     <div>
123                                                       <div>
124                                                         <div>
125               </div>
126             </div>
127           </div>
128         </div>
129       </div>
130     </div>
131   </div>
132 </div>
133 </div>
134 </div>
135 </div>
136 </div>
137 </div>
138 </div>
139 </div>
140 </div>
141 </div>
142 </div>
143 </div>
144 </div>
145 </div>
146 </div>
147 </div>
148 </div>
149 </div>
150 </div>
151 </div>
152 </div>
153 </div>
154 </div>
155 </div>
156 </div>
157 </div>
158 </div>
159 </div>
160 </div>
161 </div>
162 </div>
163 </div>
164 </div>
165 </div>
166 </div>
167 </div>
168 </div>
169 </div>
170 </div>
171 </div>
172 </div>
173 </div>
174 </div>
175 </div>
176 </div>
177 </div>
178 </div>
179 </div>
180 </div>
181 </div>
182 </div>
183 </div>
184 </div>
185 </div>
186 </div>
187 </div>
188 </div>
189 </div>
190 </div>
191 </div>
192 </div>
193 </div>
194 </div>
195 </div>
196 </div>
197 </div>
198 </div>
199 </div>
200 </div>
201 </div>
202 </div>
203 </div>
204 </div>
205 </div>
206 </div>
207 </div>
208 </div>
209 </div>
210 </div>
211 </div>
212 </div>
213 </div>
214 </div>
215 </div>
216 </div>
217 </div>
218 </div>
219 </div>
220 </div>
221 </div>
222 </div>
223 </div>
224 </div>
225 </div>
226 </div>
227 </div>
228 </div>
229 </div>
230 </div>
231 </div>
232 </div>
233 </div>
234 </div>
235 </div>
236 </div>
237 </div>
238 </div>
239 </div>
240 </div>
241 </div>
242 </div>
243 </div>
244 </div>
245 </div>
246 </div>
247 </div>
248 </div>
249 </div>
250 </div>
251 </div>
252 </div>
253 </div>
254 </div>
255 </div>
256 </div>
257 </div>
258 </div>
259 </div>
260 </div>
261 </div>
262 </div>
263 </div>
264 </div>
265 </div>
266 </div>
267 </div>
268 </div>
269 </div>
270 </div>
271 </div>
272 </div>
273 </div>
274 </div>
275 </div>
276 </div>
277 </div>
278 </div>
279 </div>
280 </div>
281 </div>
282 </div>
283 </div>
284 </div>
285 </div>
286 </div>
287 </div>
288 </div>
289 </div>
290 </div>
291 </div>
292 </div>
293 </div>
294 </div>
295 </div>
296 </div>
297 </div>
298 </div>
299 </div>
300 </div>
301 </div>
302 </div>
303 </div>
304 </div>
305 </div>
306 </div>
307 </div>
308 </div>
309 </div>
310 </div>
311 </div>
312 </div>
313 </div>
314 </div>
315 </div>
316 </div>
317 </div>
318 </div>
319 </div>
320 </div>
321 </div>
322 </div>
323 </div>
324 </div>
325 </div>
326 </div>
327 </div>
328 </div>
329 </div>
330 </div>
331 </div>
332 </div>
333 </div>
334 </div>
335 </div>
336 </div>
337 </div>
338 </div>
339 </div>
340 </div>
341 </div>
342 </div>
343 </div>
344 </div>
345 </div>
346 </div>
347 </div>
348 </div>
349 </div>
350 </div>
351 </div>
352 </div>
353 </div>
354 </div>
355 </div>
356 </div>
357 </div>
358 </div>
359 </div>
360 </div>
361 </div>
362 </div>
363 </div>
364 </div>
365 </div>
366 </div>
367 </div>
368 </div>
369 </div>
370 </div>
371 </div>
372 </div>
373 </div>
374 </div>
375 </div>
376 </div>
377 </div>
378 </div>
379 </div>
380 </div>
381 </div>
382 </div>
383 </div>
384 </div>
385 </div>
386 </div>
387 </div>
388 </div>
389 </div>
390 </div>
391 </div>
392 </div>
393 </div>
394 </div>
395 </div>
396 </div>
397 </div>
398 </div>
399 </div>
400 </div>
401 </div>
402 </div>
403 </div>
404 </div>
405 </div>
406 </div>
407 </div>
408 </div>
409 </div>
410 </div>
```

Figure 259: Sample Code of UI - 35

Admin_view_appointment.html

```

File Edit Selection View Go Run Terminal Help
templates > admin_view_appointment.html
admin_view_appointment.html
1  {% extends 'adminbase.html' %}
2  {% load static %}
3  {% block title %}
4  All Appointments
5  {% endblock title %}
6  {% block content %}
7  {% if messages in messages %}
8  <div class="alert alert-{{ message.tags }}" style="text-align: center;" role="alert">
9  {{ message }}
10 </div>
11 {% endif %}
12 </div>
13 <h1 class="text-center" style="margin-top: 40px;">All Appointments</h1>
14 <div style="margin-top: 40px;">
15 <div class="panel panel-primary" style="margin-bottom: 10px;">
16   <table class="table table-hover" id="example">
17     <thead>
18       <tr>
19         <th>Name</th>
20         <th>Date</th>
21         <th>Time</th>
22         <th>Service</th>
23         <th>Doctor</th>
24         <th>Description</th>
25         <th>Status</th>
26       </tr>
27     </thead>
28     {% for a in appointment %}
29       <tr>
30         <td>{{a.patientUser}}</td>
31         <td>{{a.date}}</td>
32         <td>{{a.time}}</td>
33         <td>{{a.service}}</td>
34         <td>{{a.doctor}}</td>
35         <td>{{a.description}}</td>
36         {% if a.status == 1 %}
37           <td><button class="btn btn-sm btn-warning" disabled data-bs-toggle="button">Confirmed</button>
38           <td><button class="btn btn-sm" onclick="return confirm('Are you sure this appointment is finished?')">
39             name='finish' href="{% url 'finish_appointment' a.id %}#{% finish %}</a>
40           <td><button class="btn btn-danger btn-sm" onclick="return confirm('Are you sure you want to cancel this appointment?')">
41             name='cancel' href="{% url 'admin_cancel_appointment' a.id %}#{% cancel %}</a>
42           <td><button class="btn btn-sm" href="#">Pending</button>
43           <td><button class="btn btn-sm btn-danger" disabled data-bs-toggle="button">Rejected</button>
44           <td><button class="btn btn-sm btn-danger" disabled data-bs-toggle="button">Cancelled</button>
45           <td><button class="btn btn-sm btn-success" disabled data-bs-toggle="button">Finished</button>
46           <td><button class="btn btn-sm" href="#">Edit</button>
47           <td><button class="btn btn-sm" href="#">Delete</button>
48         {% endif %}
49       </tr>

```

Figure 260: Sample Code of UI - 36

The screenshot shows a Visual Studio Code window with the title "admin_view_appointment.html - dentalapplication - Visual Studio Code". The code editor displays a Django template for viewing appointments. The template uses the Django templating language with blocks like `{% for a in appointment %}` and conditionals like `{% if a.status == 1 %}`. It includes a table with columns for patient name, date, time, service, doctor, description, and status. Buttons for confirming, finishing, canceling, or rejecting appointments are provided. At the bottom, there are links for pending appointments and booking a new appointment.

```
File Edit Selection View Go Run Terminal Help
templates > admin_view_appointment.html X
17     <thead>
18         <tr>
19             <th>Name</th>
20             <th>Date</th>
21             <th>Time</th>
22             <th>Service</th>
23             <th>Doctor</th>
24             <th>Description</th>
25             <th>Status</th>
26         </tr>
27     </thead>
28     {% for a in appointment %}
29     <tr>
30         <td>{{a.patientUser}}</td>
31         <td>{{a.date}}</td>
32         <td>{{a.time}}</td>
33         <td>{{a.service}}</td>
34         <td>{{a.doctor}}</td>
35         <td>{{a.description}}</td>
36         <td>{{a.status|title}}</td>
37         <td> <button class="btn btn-sm btn-warning" disabled data-bs-toggle="button" >Confirmed</button>
38         <a class="finish_button" href="{% url 'finish_appointment' a.id %}">Finish</a>
39         <a class="cancel_button" href="{% url 'admin_cancel_appointment' a.id %}">Cancel</a></td>
40     {% else %}
41         <td> <button class="btn btn-sm btn-danger" disabled data-bs-toggle="button" >Rejected</button></td>
42     {% elif a.status == 3 %}
43         <td> <button class="btn btn-sm btn-danger" disabled data-bs-toggle="button" >Cancelled</button></td>
44     {% elif a.status == 4 %}
45         <td> <button class="btn btn-sm btn-success" disabled data-bs-toggle="button">Finished</button></td>
46     {% endif %}
47     </tr>
48     {% endfor %}
49     </table>
50     </div>
51     <div>
52         <a type="button" class="btn btn-info" href="/admin_approve_appointment" style="margin-left: 60%; width: 10%; margin-bottom: 10%;">Pending Appointments</a>
53         <a type="button" class="btn btn-info" href="/admin_book_appointment" style="width: 10%; margin-bottom: 10%;">Book an Appointment</a>
54     </div>
55     {% endblock content %}
```

Figure 261: Sample Code of UI - 37

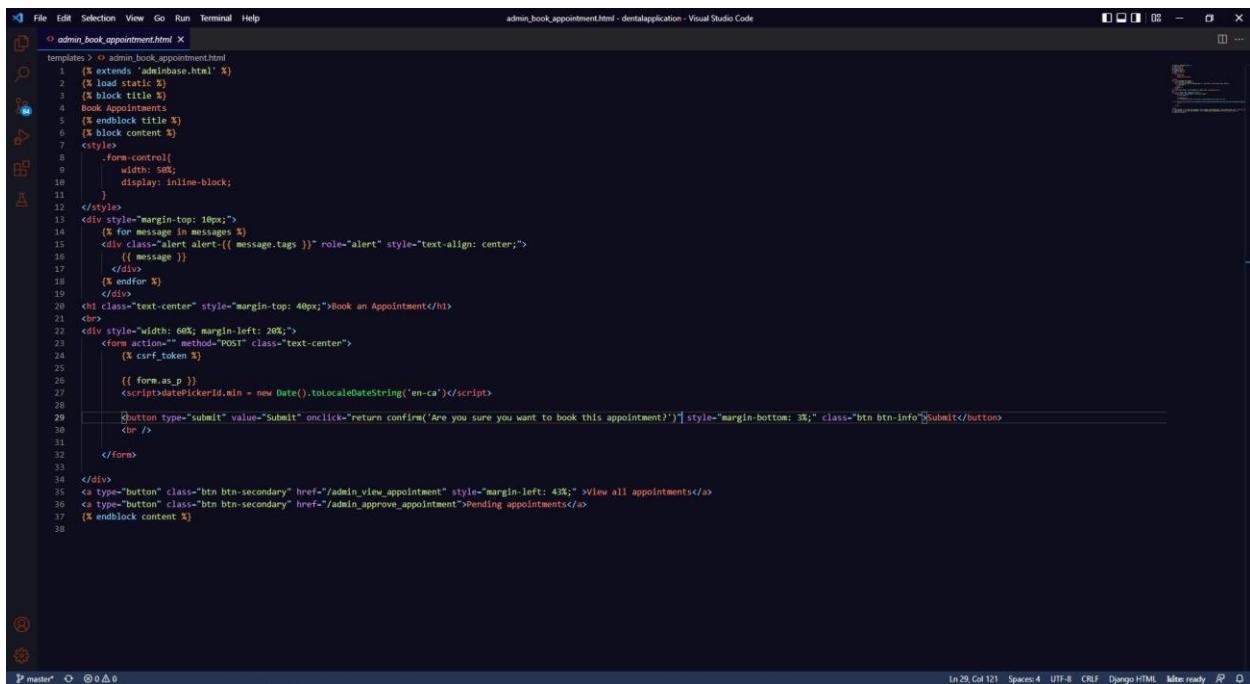
Admin_approve_appointment.html

```
File Edit Selection View Go Run Terminal Help
admin_approve_appointment.html · dentalapplication - Visual Studio Code
templates> O admin/approve_appointment.html
1 <% extends "adminbase.html" %>
2 <% load static %>
3 <% block title %>
4 Pending Appointments
5 <% endblock title %>
6 <% block content %>
7 <% for message in messages %>
8 <div class="alert alert-{{ message.tags }}" role="alert" style="text-align: center;">
9     {{ message }}
10 
```

11 <% endif %>
12 </div>
13 <h1 class="text-center" style="margin-top: 40px;">Pending Appointments</h1>
14 <div style="margin-top: 40px;">
15 <div class="panel panel-primary">
16 <table class="table table-hover" id="example">
17 <thead>
18 <tr>
19 <th>Name</th>
20 <th>Date</th>
21 <th>Time</th>
22 <th>Service</th>
23 <th>Description</th>
24 <th>Status</th>
25 </tr>
26 </thead>
27 <% for a in appointment %>
28 <tr>
29 <td>{{a.patientUser}}</td>
30 <td>{{a.date}}</td>
31 <td>{{a.time}}</td>
32 <td>{{a.service}}</td>
33 <td>{{a.doctor}}</td>
34 <td>{{a.description}}</td>
35 <td><form method="get" action="#">
36 <input type="button" class="btn btn-success btn-sm" onclick="return confirm('Are you sure you want to approve this appointment?')"
37 name="approve_button" href="{% url 'approve_appointment' a.id %}>Approve</input>
38 <input type="button" class="btn btn-danger btn-sm" onclick="return confirm('Are you sure you want to reject this appointment?')"
39 name="reject_button" href="{% url 'reject_appointment' a.id %}>Reject</input>
40 </form>
41 <% endif %>
42 </tr>
43 </table>
44 <% endif %>
45 </div>
46 </div>
47 </div>
48 All Appointments
49 Book an Appointment

Figure 262: Sample Code of UI - 38

Admin_book_appointment.html



The screenshot shows a Visual Studio Code window with the file "admin_book_appointment.html" open. The code is a Django template for booking an appointment. It includes a title block, a message block, and a form block. The form contains fields for date, time, and patient information, along with a csrf token and a submit button with an onclick confirmation. The code uses Bootstrap classes like .form-control, .text-center, and .btn btn-info.

```
File Edit Selection View Go Run Terminal Help
templates > admin_book_appointment.html
admin_book_appointment.html - dentalapplication - Visual Studio Code
1  {% extends 'admin_base.html' %} 
2  {% load static %} 
3  {% block title %} 
4  Book Appointments 
5  {% endblock title %} 
6  {% block content %} 
7      <style>
8          .form-control{
9              width: 50%;
10             display: inline-block;
11         }
12     </style>
13     <div style="margin-top: 10px;">
14         {% for message in messages %}
15             <div class="alert alert-{{ message.tags }}" role="alert" style="text-align: center;">
16                 {{ message }}
17             </div>
18         {% endfor %}
19     </div>
20     <h1 class="text-center" style="margin-top: 40px;">Book an Appointment</h1>
21     <br>
22     <div style="width: 60%; margin-left: 20%; ">
23         <form action="" method="POST" class="text-center">
24             {% csrf_token %}
25
26             {{ form.as_p }}
27             <script>datePickerId.min = new Date().toLocaleDateString('en-ca')</script>
28
29             <button type="submit" value="Submit" onclick="return confirm('Are you sure you want to book this appointment?')" style="margin-bottom: 3%; ">Submit</button>
30             <br />
31
32         </form>
33
34     </div>
35     <a type="button" class="btn btn-secondary" href="{% url 'admin_view_appointment' %}" style="margin-left: 40%;">View all appointments</a>
36     <a type="button" class="btn btn-secondary" href="{% url 'admin_approve_appointment' %}">Pending appointments</a>
37     {% endblock content %}
```

Figure 263: Sample Code of UI - 39

Admin_view_doctors.html

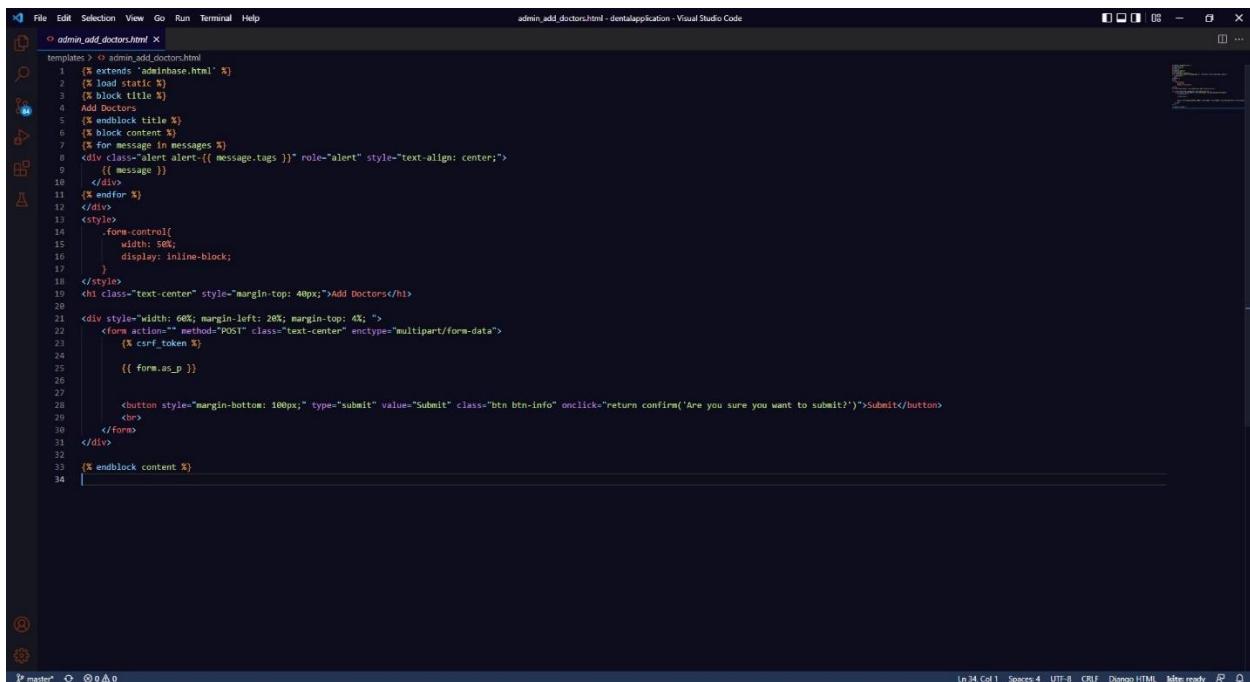
```

File Edit Selection View Go Run Terminal Help admin_view_doctors.html - dentalapplication - Visual Studio Code
templates > admin_view_doctors.html
1  {% extends 'adminbase.html' %} admin_view_doctors.html - dentalapplication - Visual Studio Code
2  {% load static %}
3  {% block title %}
4  Doctors
5  {% endblock title %}
6  {% block content %}
7  {% if messages %}
8  <div class="alert alert-{% message.tags %}" role="alert" style="text-align: center;">
9  {{ message }}
10 </div>
11 {% endif %}
12 </div>
13 <h1 class="text-center" style="margin-top: 40px;">All Doctors</h1>
14 <table id="example" class="table table-hover">
15   <thead>
16     <tr>
17       <th>Name</th>
18       <th>Profile Picture</th>
19       <th>Qualification</th>
20       <th>Experience</th>
21       <th>Date of Birth</th>
22       <th>Email</th>
23       <th>Mobile</th>
24       <th>Address</th>
25       <th>Update</th>
26       <th>Delete</th>
27     </tr>
28   </thead>
29   <tbody>
30     {% for d in doctors %}
31       <tr>
32         <td>{{d.full_name}}</td>
33         <td></td>
34         <td>{{d.qualification}}</td>
35         <td>{{d.experience}}</td>
36         <td>{{d.date_of_birth}}</td>
37         <td>{{d.email}}</td>
38         <td>{{d.phone_number}}</td>
39         <td>{{d.address}}</td>
40         <td><a href="#" class="btn-primary btn-xs" name="update_button" href="{% url 'update_doctor' d.id %}></a></td>
41         <td><a href="#" class="btn btn-danger btn-xs" onclick="return confirm('Are you sure you want to delete this Doctor?')></a></td>
42     {% endfor %}
43   </tbody>
44 </table>
45 
```

Figure 264: Sample Code of UI – 40

Figure 265: Sample Code of UI - 41

Admin_add_doctors.html



The screenshot shows a Visual Studio Code window with the file "admin_add_doctors.html" open. The code is a Django template for adding doctors. It includes a header section with a title and a message block, followed by a style block with CSS for form controls, and a main content block containing a heading and a form for submitting doctor information. The code uses Django's templating syntax ({{ }}) and CSS classes like "text-center" and "btn btn-info".

```
File Edit Selection View Go Run Terminal Help
templates > admin_add_doctors.html
admin_add_doctors.html - dentalapplication - Visual Studio Code
1  {% extends 'adminbase.html' %} 
2  {% load static %} 
3  {% block title %} 
4  Add Doctors 
5  {% endblock title %} 
6  {% block content %} 
7  {% for message in messages %} 
8  <div class="alert alert-{{ message.tags }}" role="alert" style="text-align: center;">> 
9  {{ message }} 
10 </div> 
11 {% endfor %} 
12 </div> 
13 <style> 
14   .form-control{ 
15     width: 50%; 
16     display: inline-block; 
17   } 
18 </style> 
19 <h1 class="text-center" style="margin-top: 40px;">Add Doctors</h1> 
20 <div style="width: 60%; margin-left: 20%; margin-top: 4%;"> 
21   <form action="" method="POST" class="text-center" enctype="multipart/form-data"> 
22     {% csrf_token %} 
23     {{ form.as_p }} 
24     <button style="margin-bottom: 100px;" type="submit" value="Submit" class="btn btn-info" onclick="return confirm('Are you sure you want to submit?')">Submit</button> 
25   </form> 
26 </div> 
27 </div> 
28 {% endblock content %} 
29 
```

Figure 266: Sample Code of UI - 42

Admin_view_services.html

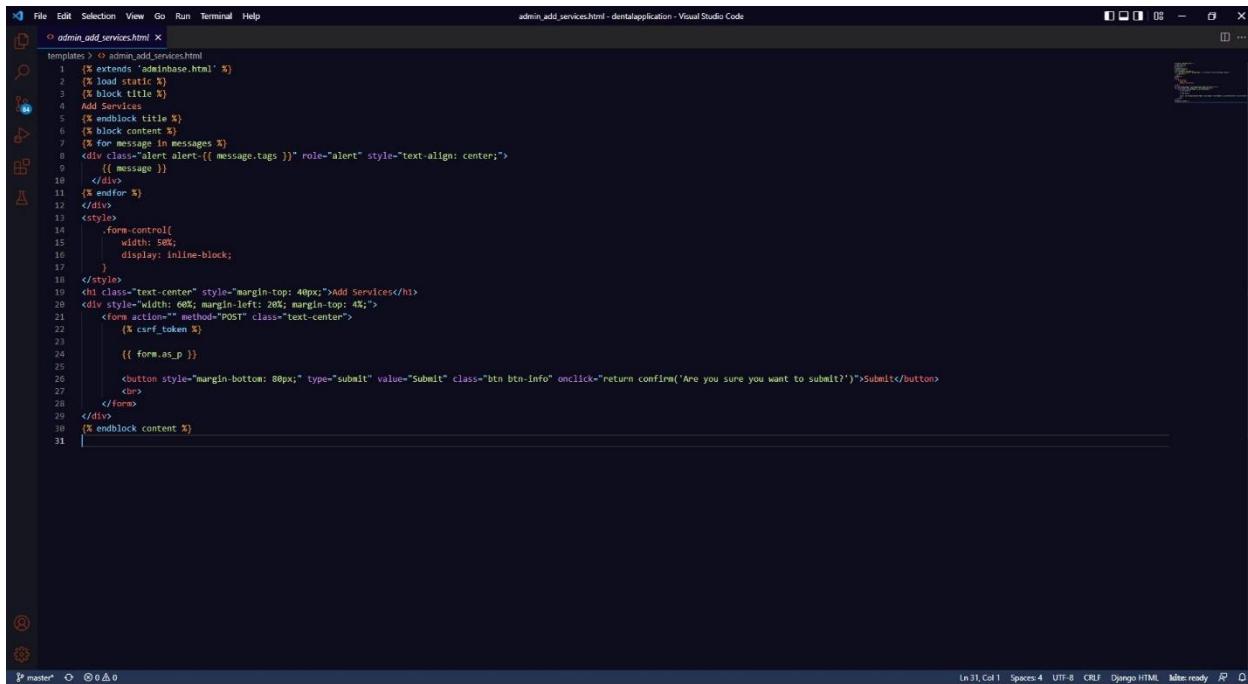
File Edit Selection View Go Run Terminal Help

admin_view_services.html - dentalapplication - Visual Studio Code

```
admin_view_services.html X
templates > admin_view_services.html
1  [% extends "adminbase.html" %]
2  [% load static %]
3  [% block title %]
4  Services
5  [% endblock title %]
6  [% block content %]
7  [% for message in messages %]
8  <div class="alert alert-{{ message.tags }}" role="alert" style="text-align: center;">
9   {{ message }}
10  </div>
11  [% endfor %]
12 </div>
13 <h1 class="text-center" style="margin-top: 40px;">All Services</h1>
14 <div style="margin-top: 40px;">
15  <table id="example" class="table table-hover">
16    <thead>
17      <tr>
18        <th>Name</th>
19        <th>Cost</th>
20        <th>Description</th>
21        <th>Status</th>
22        <th>Balance</th>
23      </tr>
24    </thead>
25    <% for s in services %>
26      <tr>
27        <td>{{s.service_name}}</td>
28        <td>{{s.description}}</td>
29        <td>{{s.cost}}</td>
30        <td>{{s.description}}</td>
31        <td><a href="#" class="btn btn-primary btn-xs" name="update_button" href="[% url 'update_service' s.id %]"><img alt="Edit icon" style="vertical-align: middle;"/> Update</a></td>
32        <a href="#" class="btn btn-danger btn-xs" onclick="return confirm('Are you sure you want to delete this service?')" name="delete_button" href="[% url 'delete_service' s.id %]"><img alt="Delete icon" style="vertical-align: middle;"/> Delete</a></td>
33      </tr>
34    <% endfor %>
35  </table>
36 </div>
37 <div style="text-align: right; margin-top: 20px;">
38  <a href="#" type="button" class="btn btn-info" href="/admin_add_services" style="margin-left: 84%; margin-bottom: 10%; width:15%; margin-top: 3%;">Add New Service</a>
39 </div>
40 [% endblock content %]
41
42
43
44
45
46
47
```

Figure 267: Sample Code of UI - 43

Admin_add_services.html

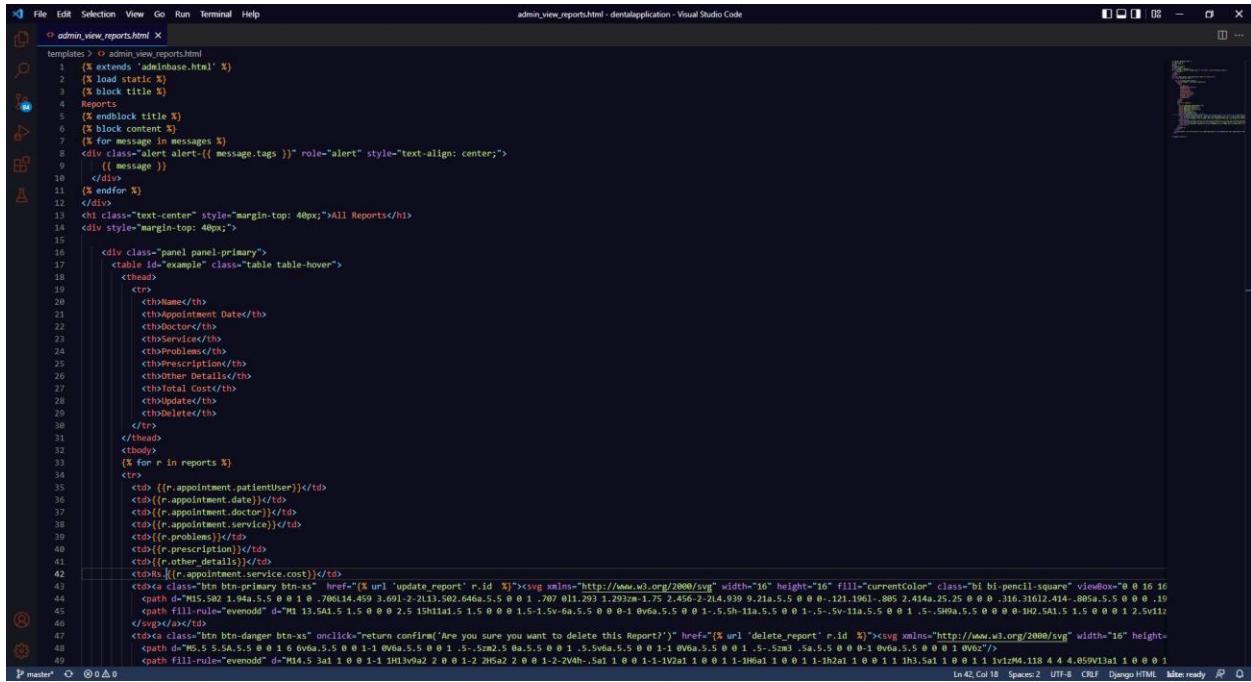


The screenshot shows a Visual Studio Code window with the file "admin_add_services.html" open. The code is a Django template for adding services. It includes a title block, a message block, a style section with inline-block styling, and a form block containing a csrf_token input and a submit button with an onclick confirmation. The code is written in Python's Jinja2 syntax.

```
File Edit Selection View Go Run Terminal Help admin_add_services.html - dentalapplication - Visual Studio Code
templates > admin_add_services.html
1  {% extends 'adminbase.html' %} 
2  {% load static %} 
3  {% block title %} 
4  Add Services 
5  {% endblock title %} 
6  {% block content %} 
7  {% for message in messages %} 
8  <div class="alert alert-{{ message.tags }}" role="alert" style="text-align: center;"> 
9  {{ message }} 
10 </div> 
11 {% endiffor %} 
12 </div> 
13 <style> 
14   .form-control{ 
15     width: 86%; 
16     display: inline-block; 
17   } 
18 </style> 
19 <h1 class="text-center" style="margin-top: 40px;">Add Services 
20 <div style="width: 60%; margin-left: 20%; margin-top: 4%;"> 
21   <form action="" method="POST" class="text-center"> 
22     {% csrf_token %} 
23     {{ form.as_p }} 
24     <button style="margin-bottom: 80px;" type="submit" value="Submit" class="btn btn-info" onclick="return confirm('Are you sure you want to submit?')>Submit</button> 
25     <br> 
26   </form> 
27 </div> 
28 </div> 
29 {% endblock content %}
```

Figure 268: Sample Code of UI - 44

Admin_view_reports.html



```

File Edit Selection View Go Run Terminal Help admin_view_reports.html - dentalapplication - Visual Studio Code
templates > admin_view_reports.html
1  {% extends 'adminbase.html' %}
2  {% load static %}
3  {% block title %}
4  Reports
5  {% endblock content %}
6  {% block content %}
7  {% messages %}
8  <div class="alert alert-{% message.tags %}" role="alert" style="text-align: center;">
9  {{ message }}
10 </div>
11 {% endif %}
12 </div>
13 <h1 class="text-center" style="margin-top: 40px;">All Reports</h1>
14 <div style="margin-top: 40px;">
15
16 <div class="panel panel-primary">
17   <table id="example" class="table table-hover">
18     <thead>
19       <tr>
20         <th>Name</th>
21         <th>Appointment Date</th>
22         <th>Doctor</th>
23         <th>Service</th>
24         <th>Problem</th>
25         <th>Prescription</th>
26         <th>Other Details</th>
27         <th>Total Cost</th>
28         <th>Update</th>
29         <th>Delete</th>
30       </tr>
31     </thead>
32     <tbody>
33       {% for r in reports %}
34         <tr>
35           <td>{{r.appointment.patientUser}}</td>
36           <td>{{r.appointment.date}}</td>
37           <td>{{r.appointment.doctor}}</td>
38           <td>{{r.appointment.service}}</td>
39           <td>{{r.problem}}</td>
40           <td>{{r.prescription}}</td>
41           <td>{{r.other_details}}</td>
42           <td>{{r.appointment.Service.cost}}</td>
43           <td><a href="{% url 'update_report' r.id %}"><img alt="Edit icon" width="16" height="16" style="vertical-align: middle;"/> Update</a></td>
44           <td><a href="#" onclick="return confirm('Are you sure you want to delete this Report?')><img alt="Delete icon" width="16" height="16" style="vertical-align: middle;"/> Delete</a></td>
45         </tr>
46       {% endfor %}
47     </tbody>
48   </table>
49 </div>

```

Figure 269: Sample Code of UI – 45

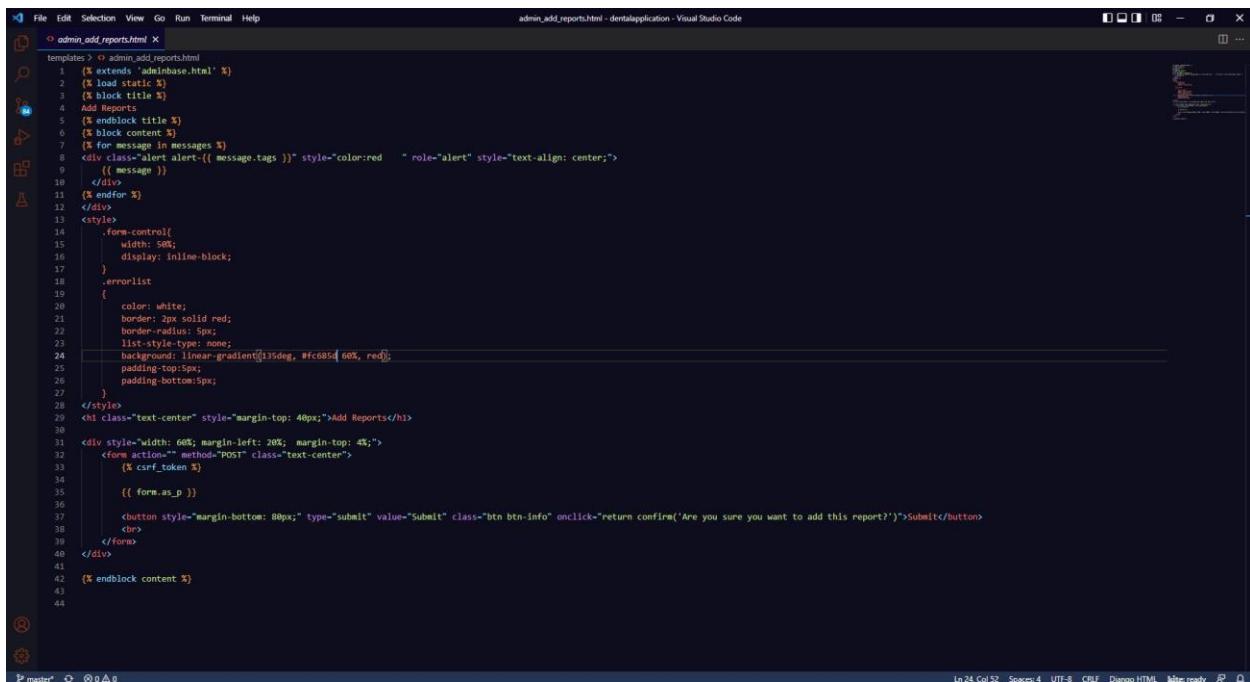
File Edit Selection View Go Run Terminal Help

admin_view_reports.html - dentalapplication - Visual Studio Code

```
File Edit Selection View Go Run Terminal Help
  admin_view_reports.html x
templates > admin_view_reports.html
  25         <th>Prescription</th>
  26         <th>Other Details</th>
  27         <th>Total Cost</th>
  28         <th>Update</th>
  29         <th>Delete</th>
  30     </tr>
  31 </thead>
  32 </table>
  33 (% for r in reports %)
  34 <tr>
  35     <td>{{r.appointment.patientUser}}</td>
  36     <td>{{r.appointment.date}}</td>
  37     <td>{{r.appointment.doctor}}</td>
  38     <td>{{r.appointment.service}}</td>
  39     <td>{{r.prescription}}</td>
  40     <td>{{r.other_details}}</td>
  41     <td>{{r.appointment.service.cost}}</td>
  42     <td><a class="btn-primary btn-xs" href="{% url 'update_report' r.id %}"><img alt="Edit icon" style="vertical-align: middle;"/> Update</a></td>
  43     <td><a class="btn-danger btn-xs" onclick="return confirm('Are you sure you want to delete this Report?')" href="{% url 'delete_report' r.id %}"><img alt="Delete icon" style="vertical-align: middle;"/> Delete</a></td>
  44 </tr>
  45 (% endfor %)
  46 </table>
  47 </div>
  48 <div style="text-align: right; margin-top: 10px;">
  49     <a type="button" class="btn btn-info" href="/admin_add_reports" style="margin-left: 84%; margin-bottom: 10px; width:15%;">Add New Report</a>
  50 </div>
  51 (% endblock content %)
```

Figure 270: Sample Code of UI - 46

Admin_add_report.html



The screenshot shows a Visual Studio Code window with the file "admin_add_report.html" open. The code is a Django template for an "Add Reports" page. It includes a header section with a title and a message block, followed by a style block containing CSS for error messages, and a main content block with a form for adding a report. The code uses Django's templating syntax (e.g., {% extends %}, {% load %}, {% block %}) and CSS classes like "form-control" and "errorlist".

```
File Edit Selection View Go Run Terminal Help admin_add_report.html - dentalapplication - Visual Studio Code
templates > admin_add_report.html
admin_add_report.html
1  {% extends 'adminbase.html' %}
2  {% load static %}
3  {% block title %}
4  Add Reports
5  {% endblock title %}
6  {% block content %}
7  {% for message in messages %}
8  <div class="alert alert-{% if message.tags %}{{ message.tags }}{% else %}info{% endif %}" role="alert" style="text-align: center;">
9    {{ message }}
10   </div>
11  {% endfor %}
12 </div>
13 <style>
14   .form-control {
15     width: 300px;
16     display: inline-block;
17   }
18   .errorlist {
19     color: white;
20     border: 2px solid red;
21     border-radius: 5px;
22     list-style-type: none;
23     background-color: linear-gradient(135deg, #ffccbc 60%, red);
24     padding-top: 5px;
25     padding-bottom: 5px;
26   }
27 </style>
28 <h1 class="text-center" style="margin-top: 40px;">Add Reports
29 <div style="width: 40%; margin-left: 20%; margin-top: 4%;">
30   <form action="" method="POST" class="text-center">
31     {% csrf_token %}
32     {{ form.as_p }}
33     <br>
34     <button style="margin-bottom: 80px;" type="submit" value="Submit" class="btn btn-info" onclick="return confirm('Are you sure you want to add this report?')">Submit</button>
35   </form>
36 </div>
37 {% endblock content %}
38
39
40
41
42
43
44
```

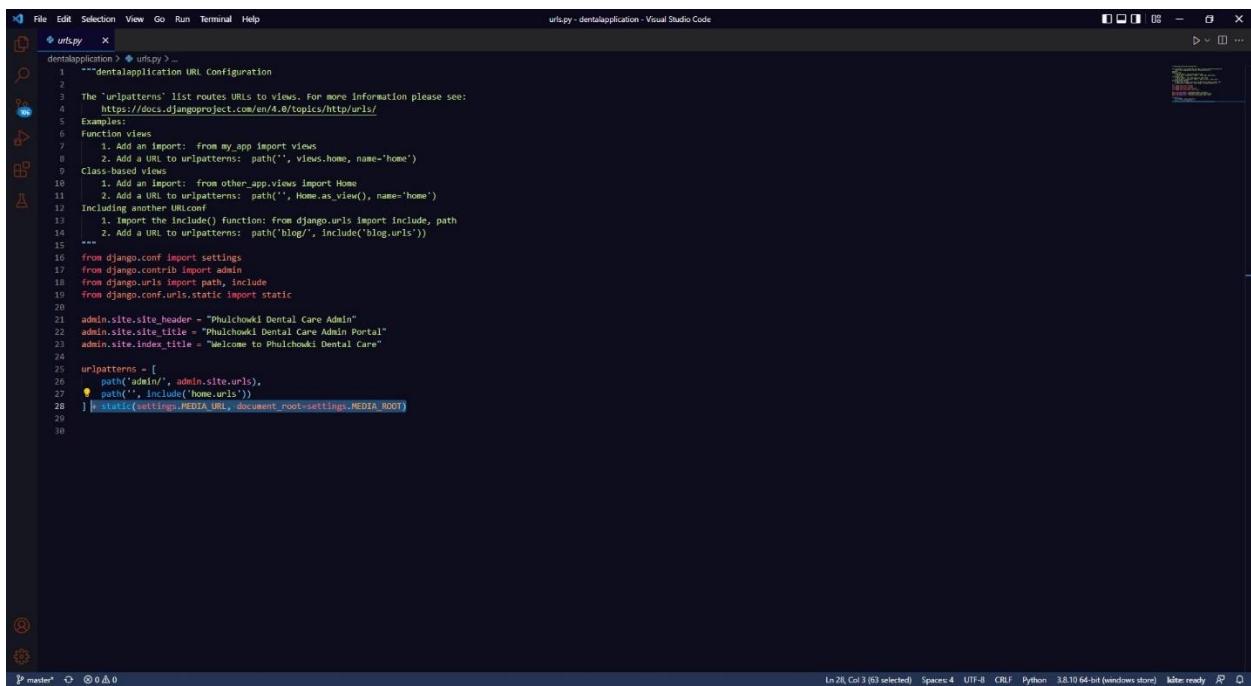
Figure 271: Sample Code of UI - 47

Admin_view_users.html

Figure 272: Sample Code of UI - 48

7.5.2 Sample code for the Automation Script

Urls.py

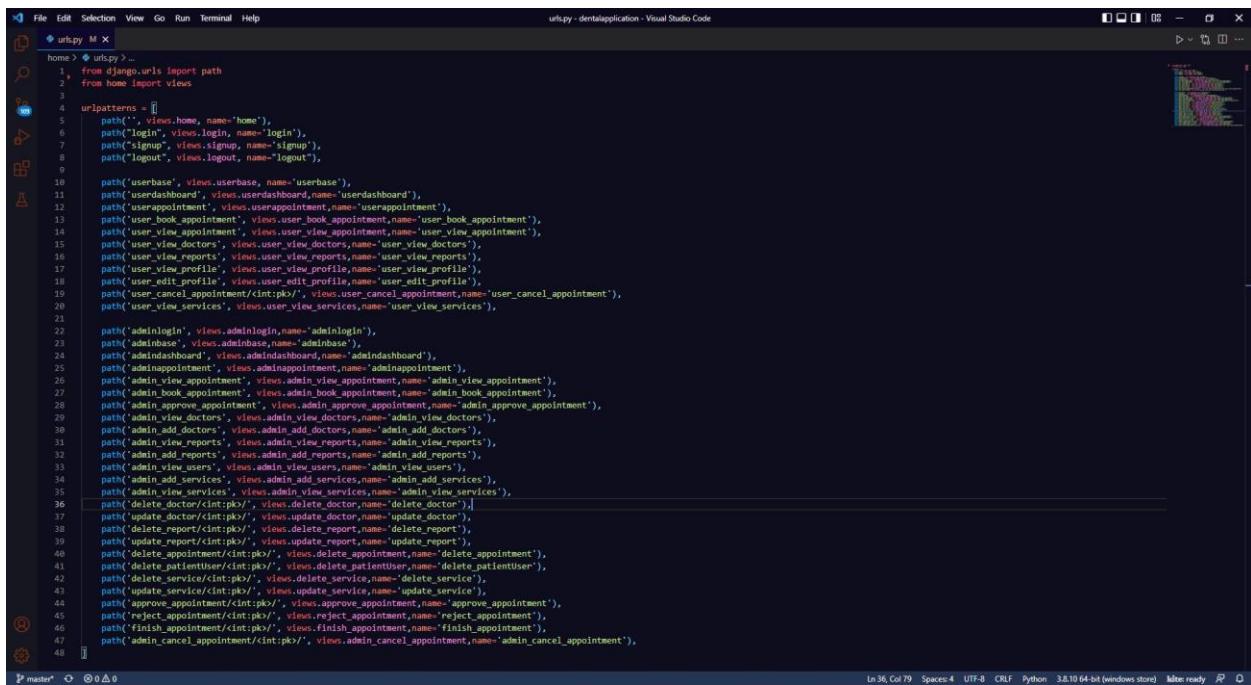


```

File Edit Selection View Go Run Terminal Help
urls.py - dentalapplication - Visual Studio Code
dentalapplication > urls.py ...
1 """ dentalapplication URL Configuration
2
3     The 'urlpatterns' list routes URLs to views. For more information please see:
4         https://docs.djangoproject.com/en/4.0/topics/http/urls/
5     Examples:
6     Function views
7         1. Add an import: from my_app import views
8             2. Add a URL to urlpatterns: path('', views.home, name='home')
9     Class-based views
10        1. Add an import: from other_app.views import Home
11            2. Add a URL to urlpatterns: path('', Home.as_view(), name='home')
12    Including another URLconf
13        1. Import the include() function: from django.urls import include, path
14            2. Add a URL to urlpatterns: path("blog/", include("blog.urls"))
15
16    from django.conf import settings
17    from django.contrib import admin
18    from django.urls import path, include
19    from django.conf.urls.static import static
20
21    admin.site.site_header = "Phulchowki Dental Care Admin"
22    admin.site.site_title = "Phulchowki Dental Care Admin Portal"
23    admin.site.index_title = "Welcome to Phulchowki Dental Care"
24
25    urlpatterns = [
26        path("admin/", admin.site.urls),
27        path("", include("home.urls")),
28    ] + static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
29
30
31
32
33
34
35
36

```

Figure 273: Sample Code of Automation Script - 1



```

File Edit Selection View Go Run Terminal Help
urls.py M X
urls.py - dentalapplication - Visual Studio Code
home > urls.py >
1 from django.urls import path
2 from home import views
3
4 urlpatterns = [
5     path('', views.home, name='home'),
6     path('login', views.login, name='login'),
7     path('signup', views.signup, name='signup'),
8     path('logout', views.logout, name='logout'),
9
10    path('userbase', views.userbase, name='userbase'),
11    path('usersashboard', views.usersashboard, name='usersashboard'),
12    path('userappointment', views.userappointment, name='userappointment'),
13    path('user_book_appointment', views.user_book_appointment, name='user_book_appointment'),
14    path('user_view_appointment', views.user_view_appointment, name='user_view_appointment'),
15    path('user_view_doctors', views.user_view_doctors, name='user_view_doctors'),
16    path('user_view_reports', views.user_view_reports, name='user_view_reports'),
17    path('user_edit_profile', views.user_edit_profile, name='user_edit_profile'),
18    path('user_cancel_appointment<int:pk>', views.user_cancel_appointment, name='user_cancel_appointment'),
19    path('user_view_services', views.user_view_services, name='user_view_services'),
20
21    path('adminlogin', views.adminlogin, name='adminlogin'),
22    path('adminbase', views.adminbase, name='adminbase'),
23    path('adminashboard', views.adminashboard, name='adminashboard'),
24    path('adminappointment', views.adminappointment, name='adminappointment'),
25    path('admin_book_appointment', views.admin_book_appointment, name='admin_book_appointment'),
26    path('admin_approve_appointment', views.admin_approve_appointment, name='admin_approve_appointment'),
27    path('admin_view_doctors', views.admin_view_doctors, name='admin_view_doctors'),
28    path('admin_addDoctors', views.admin_addDoctors, name='admin_addDoctors'),
29    path('admin_view_reports', views.admin_view_reports, name='admin_view_reports'),
30    path('admin_add_reports', views.admin_add_reports, name='admin_add_reports'),
31    path('admin_view_users', views.admin_view_users, name='admin_view_users'),
32    path('admin_add_users', views.admin_add_users, name='admin_add_users'),
33    path('admin_view_services', views.admin_view_services, name='admin_view_services'),
34    path('admin_delete_doctor<int:pk>', views.delete_doctor, name='delete_doctor'),
35    path('update_doctor<int:pk>', views.update_doctor, name='update_doctor'),
36    path('delete_report<int:pk>', views.delete_report, name='delete_report'),
37    path('update_report<int:pk>', views.update_report, name='update_report'),
38    path('delete_appointment<int:pk>', views.delete_appointment, name='delete_appointment'),
39    path('delete_patientUser<int:pk>', views.delete_patientUser, name='delete_patientUser'),
40    path('delete_service<int:pk>', views.delete_service, name='delete_service'),
41    path('update_service<int:pk>', views.update_service, name='update_service'),
42    path('approve_appointment<int:pk>', views.approve_appointment, name='approve_appointment'),
43    path('reject_appointment<int:pk>', views.reject_appointment, name='reject_appointment'),
44    path('finish_appointment<int:pk>', views.finish_appointment, name='finish_appointment'),
45    path('admin_cancel_appointment<int:pk>', views.admin_cancel_appointment, name='admin_cancel_appointment'),
46
47]

```

In 36 Col 79 Spaces:4 UTF-8 CRLF Python 3.8.10 64-bit (windows store) Mine ready

Figure 274: Sample Code of Automation Script - 2

Models.py

```

File Edit Selection View Go Run Terminal Help
models.py ->
home > models.py
1 from django.db import models
2 from django.contrib.auth.models import User
3
4 # This is a model for service
5 class Service(models.Model):
6     service_name = models.CharField('Service', max_length=50)
7     cost = models.IntegerField()
8     description = models.TextField()
9
10    #This to display the data of the service by the service name
11    def __str__(self):
12        return self.service_name
13
14
15    # This is a model for Doctor
16    class Doctor(models.Model):
17        full_name = models.CharField(max_length=100)
18        qualification = models.CharField(max_length=50)
19        date_of_birth = models.DateField()
20        email = models.EmailField(max_length=100)
21        phone_number = models.CharField(max_length=15)
22        address = models.CharField(max_length=50)
23        profile_picture = models.ImageField(upload_to='profile_pic/doctor_profile_pic/', null=True, blank=True)
24        experience = models.CharField(max_length=10, default="1 year")
25
26        #This to display the data of the doctor by the doctor name
27        def __str__(self):
28            return self.full_name
29
30    # This is a model for PatientUser
31    class PatientUser(models.Model):
32        user=models.OneToOneField(User, on_delete=models.CASCADE, null=True, blank=False, on_delete=models.CASCADE)
33        date_of_birth = models.DateField()
34        phone_number = models.CharField(max_length=15)
35        address = models.CharField(max_length=50)
36        profile_picture = models.ImageField(upload_to='profile_pic/user_profile_pic/', default='img/defaultuser.png')
37
38        @property
39        def get_name(self):
40            return self.user.first_name + self.user.last_name
41
42        @property
43        def get_id(self):
44            return self.user.id
45
46        #This to display the data of the PatientUser by the User's first name and last name
47        def __str__(self):
48            return self.user.first_name + self.user.last_name
49

```

In 89, Col 1 | Spaces: 4 | UTF-8 | CR/LF | Python 3.8.10 64-bit (Windows Store) | Editor ready | JP

Figure 275: Sample Code of Automation Script - 2

```

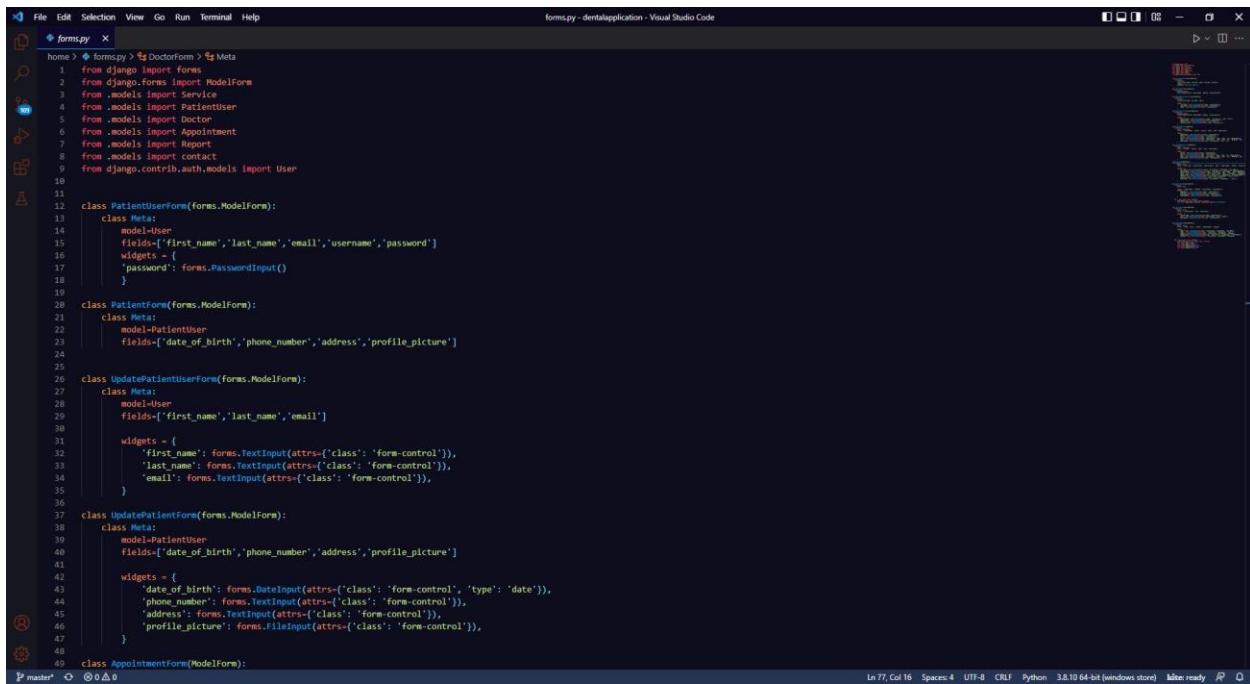
File Edit Selection View Go Run Terminal Help
models.py - dentalapplication - Visual Studio Code
home > models.py X
1 home > models.py
2
3     def get_id(self):
4         return self.user.id
5
6     #this to display the data of the PatientUser by the user's first name and last name
7     def __str__(self):
8         return self.user.first_name+" "+self.user.last_name
9
10
11 # This is a model for Appointment:
12 class Appointment(models.Model):
13     patientUser = models.ForeignKey(PatientUser, blank=False, null=False, on_delete=models.CASCADE)
14     doctor = models.ForeignKey(Doctor, blank=False, null=False, on_delete=models.CASCADE)
15     service = models.ForeignKey(Service, blank=False, null=False, on_delete=models.CASCADE)
16     date = models.DateField()
17     time = models.TimeField()
18     description = models.TextField()
19     status = models.IntegerField('Status', default=0)
20
21     #this to display the data of the appointment by the user's name, last name, appointment's date and time
22     def __str__(self):
23         return self.patientUser.user.first_name+" "+self.patientUser.user.last_name+" ("+str(self.date)+") ("+str(self.time)+")"
24
25
26 # This is a model for Report:
27 class Report(models.Model):
28     appointment = models.OneToOneField(Appointment, unique=True, blank=False, null=False, on_delete=models.CASCADE)
29     problems = models.TextField()
30     prescription = models.TextField()
31     other_details = models.TextField()
32
33     #this to display the data of the report by the user's name, last name, appointment's date and time
34     def __str__(self):
35         return self.appointment.patientUser.user.first_name+" "+self.appointment.patientUser.user.last_name+" ("+str(self.appointment.date)+") ("+str(self.appointment.time)+")"
36
37
38 # This is a model to collect the data from the contact us form in the home page
39 class Contact(models.Model):
40     name = models.CharField(max_length=100)
41     email = models.EmailField(max_length=100)
42     subject = models.CharField(max_length=100)
43     phone_number = models.CharField(max_length=15)
44     message = models.TextField()
45
46     #this to display the data of the contact by the user's name
47     def __str__(self):
48         return self.name
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89

```

Ln 89, Col 1 Spaces: 4 UTT-B CRLF Python 3.8.10 64-bit (Windows store) idle ready

Figure 276: Sample Code of Automation Script - 3

Forms.py



```

File Edit Selection View Go Run Terminal Help
forms.py > DoctorForm > Meta
home > forms.py > DoctorForm > Meta
1 from django import forms
2 from django.forms import ModelForm
3 from .models import Service
4 from .models import PatientUser
5 from .models import Doctor
6 from .models import Appointment
7 from .models import Report
8 from .models import Contact
9 from django.contrib.auth.models import User
10
11 class PatientUserForm(forms.ModelForm):
12     class Meta:
13         model=User
14         fields=['first_name','last_name','email','username','password']
15         widgets = {
16             'password': forms.PasswordInput()
17         }
18
19 class PatientForm(forms.ModelForm):
20     class Meta:
21         model=PatientUser
22         fields=['date_of_birth','phone_number','address','profile_picture']
23
24
25 class UpdatePatientUserForm(forms.ModelForm):
26     class Meta:
27         model=User
28         fields=['first_name','last_name','email']
29
30         widgets = {
31             'first_name': forms.TextInput(attrs={'class': 'form-control'}),
32             'last_name': forms.TextInput(attrs={'class': 'form-control'}),
33             'email': forms.TextInput(attrs={'class': 'form-control'})
34         }
35
36 class UpdatePatientForm(forms.ModelForm):
37     class Meta:
38         model=PatientUser
39         fields=['date_of_birth','phone_number','address','profile_picture']
40
41         widgets = {
42             'date_of_birth': forms.DateInput(attrs={'class': 'form-control', 'type': 'date'}),
43             'phone_number': forms.TextInput(attrs={'class': 'form-control'}),
44             'address': forms.TextInput(attrs={'class': 'form-control'}),
45             'profile_picture': forms.FileInput(attrs={'class': 'form-control'})
46         }
47
48
49 class AppointmentForm(ModelForm):

```

In 77, Col 16 | Spaces: 4 | UTF-8 | CRLF | Python 3.8.10 64-bit (windows store) | idle ready |  

Figure 277: Sample Code of Automation Script - 4

```

File Edit Selection View Go Run Terminal Help
forms.py - dentalapplication - Visual Studio Code
home > forms.py > DoctorForm > Meta
49 class AppointmentForm(ModelForm):
50     class Meta:
51         model = Appointment
52         fields = ('patientUser', 'doctor', 'service', 'date', 'time', 'description')
53
54     widgets = {
55         'patientUser': forms.Select(attrs={'class': 'form-control'}),
56         'doctor': forms.Select(attrs={'class': 'form-control'}),
57         'service': forms.Select(attrs={'class': 'form-control'}),
58         'date': forms.DateInput(attrs={'class': 'form-control', 'type': 'date', 'id': 'datePickerId'}),
59         'time': forms.TimeInput(attrs={'class': 'form-control', 'type': 'time', 'min': '09:00:00', 'max': '17:00:00'}),
60         'description': forms.Textarea(attrs={'class': 'form-control', 'rows': '4'})
61     }
62
63 class UserAppointmentForm(ModelForm):
64     class Meta:
65         model = Appointment
66         fields = ('doctor', 'service', 'date', 'time', 'description')
67
68     widgets = {
69         'doctor': forms.Select(attrs={'class': 'form-control'}),
70         'service': forms.Select(attrs={'class': 'form-control'}),
71         'date': forms.DateInput(attrs={'class': 'form-control', 'type': 'date', 'id': 'datePickerId'}),
72         'time': forms.TimeInput(attrs={'class': 'form-control', 'type': 'time', 'min': '09:00:00', 'max': '17:00:00'}),
73         'description': forms.Textarea(attrs={'class': 'form-control'})
74     }
75
76 class DoctorForm(ModelForm):
77     class Meta:
78         model = Doctor
79         fields = ('full_name', 'qualification', 'date_of_birth', 'email', 'phone_number', 'address', 'profile_picture', 'experience')
80
81     widgets = {
82         'full_name': forms.TextInput(attrs={'class': 'form-control', 'placeholder': 'Full Name'}),
83         'qualification': forms.TextInput(attrs={'class': 'form-control', 'placeholder': 'Bachelors/Masters in Dental Science'}),
84         'date_of_birth': forms.DateInput(attrs={'class': 'form-control', 'type': 'date', 'max': '2000-01-01'}),
85         'email': forms.EmailInput(attrs={'class': 'form-control', 'type': 'email', 'placeholder': 'example@gmail.com'}),
86         'phone_number': forms.TextInput(attrs={'class': 'form-control', 'placeholder': '9977-mobileNumber'}),
87         'address': forms.TextInput(attrs={'class': 'form-control', 'placeholder': 'Street, City'}),
88         'profile_picture': forms.FileInput(attrs={'class': 'form-control'}),
89         'experience': forms.TextInput(attrs={'class': 'form-control', 'placeholder': 'years'})
90     }
91
92 class ReportForm(forms.ModelForm):
93     class Meta:
94         model = Report
95
96
In 77, Col 16  Spaces: 4  UTF-8  CRLF  Python  3.8.10 64-bit (windows store)  Editor ready

```

Figure 278: Sample Code of Automation Script - 5

```

File Edit Selection View Go Run Terminal Help
forms.py - dentalapplication - Visual Studio Code
home > forms.py > DoctorForm > Meta
96     fields = ('appointment', 'problems', 'prescription', 'other_details')
97     widgets = {
98         'appointment': forms.Select(attrs={'class': 'form-control'}),
99         'problems': forms.Textarea(attrs={'class': 'form-control'}),
100        'prescription': forms.Textarea(attrs={'class': 'form-control'}),
101        'other_details': forms.Textarea(attrs={'class': 'form-control'})
102    }
103
104    def __init__(self, *args, **kwargs):
105        super(ReportForm, self).__init__(*args, **kwargs)
106        self.fields['appointment'].queryset = Appointment.objects.filter(status=4)
107
108
109
110 class ServiceForm(forms.ModelForm):
111     class Meta:
112         model = Service
113         fields = ('service_name', 'cost', 'description')
114
115         widgets = {
116             'service_name': forms.TextInput(attrs={'class': 'form-control'}),
117             'cost': forms.TextInput(attrs={'class': 'form-control', 'type': 'number'}),
118             'description': forms.Textarea(attrs={'class': 'form-control'})
119         }
120
121
122 class ContactForm(forms.ModelForm):
123     class Meta:
124         model = contact
125         fields = ('name', 'email', 'subject', 'phone_number', 'message')
126
127         widgets = {
128             'name': forms.TextInput(attrs={'class': 'form-control', 'placeholder': 'Full Name'}),
129             'email': forms.EmailInput(attrs={'class': 'form-control', 'placeholder': 'Email'}),
130             'subject': forms.TextInput(attrs={'class': 'form-control', 'placeholder': 'Subject'}),
131             'phone_number': forms.TextInput(attrs={'class': 'form-control', 'placeholder': 'Phone Number'}),
132             'message': forms.Textarea(attrs={'class': 'form-control', 'placeholder': 'Message'})
133         }
134
135
136     def __init__(self, *args, **kwargs):
137         super(ContactForm, self).__init__(*args, **kwargs)
138         self.fields['name'].label = ""
139         self.fields['email'].label = ""
140         self.fields['subject'].label = ""
141         self.fields['phone_number'].label = ""
142         self.fields['message'].label = ""
143
144
In 77, Col 16  Spaces: 4  UTF-8  CRLF  Python  3.8.10 64-bit (windows store)  Editor ready

```

Figure 279: Sample Code of Automation Script - 6

Views.py

The screenshot shows the Visual Studio Code interface with the file 'views.py' open. The code is written in Python and defines several view functions for a dental application. The functions include 'home' (which handles the home page and contact form submission), 'signUp' (for user registration), and 'signUpPatientUser' (for patient user registration). The code uses Django's forms and models, including User, Doctor, Patient, Appointment, and Report models. It also uses Django's authentication system and redirect logic.

```
File Edit Selection View Go Run Terminal Help
views.py - dentalapplication - Visual Studio Code

home > views.py > ...
1  from django.shortcuts import redirect, render, HttpResponseRedirect
2  from django.contrib.auth.models import User, auth
3  from django.contrib import messages
4  from django.http import HttpResponseRedirect
5  from .models import Service
6  from .models import Doctor
7  from .models import PatientUser
8  from .models import Appointment
9  from .models import Report
10 from .forms import AppointmentForm, UpdatePatientForm, UpdatePatientUserForm, UserAppointmentForm
11 from .forms import DoctorForm
12 from .forms import ReportForm
13 from .forms import PatientUserForm
14 from .forms import PatientForm
15 from .forms import ServiceForm
16 from .forms import ContactForm
17 from django.contrib.auth.models import Group
18 from django.http import HttpResponseRedirect
19 from django.contrib.auth.decorators import login_required, user_passes_test
20
21
22
23
24
25 #View for Home page. Created a contact form for the contact us page.
26 def home(request):
27     form = ContactForm()
28     if request.method == "POST" or None:
29         form = ContactForm(request.POST or None)
30         if form.is_valid():
31             form.save()
32             messages.success(request, 'Form successfully submitted. Our team will reach out to you soon.')
33             return HttpResponseRedirect('#')
34     context = {'form': form}
35     return render(request, 'home.html', context)
36
37 #View for User Registration page.
38 def signUp(request):
39     userForm=PatientUserForm()
40     patientForm=PatientForm()
41     mydict={'userForm':userForm,'patientForm':patientForm}
42     if request.method=='POST':
43         userForm=PatientUserForm(request.POST)
44         patientForm=PatientForm(request.POST,request.FILES)
45         if userForm.is_valid() and patientForm.is_valid():
46             userForm.user.set_password(userForm.password)
47             userForm.user.set_password(patientForm.password)
48             userForm.user.save()
49             patientForm.patient.save()

In 1 Col 1 Spaces:4 UTF-8 CS LF Python 3.8.10 64-bit (windows store) later ready
```

Figure 280: Sample Code of Automation Script - 7

```

File Edit Selection View Go Run Terminal Help
views.py M - views.py - views.py
home > views.py > ...
47     patient.save()
48     patient.patientForm.save()
49     patient.user.save()
50     patient.save()
51     my_patient_group = Group.objects.get_or_create(name='PATIENT') #Setting user type as patient.
52     my_patient_group[0].user_set.add(user)
53 
54     else:
55         messages.error(request, 'Invalid Details! Please fill again.')
56     return redirect('/signup')
57     messages.success(request, 'Account Successfully Created. Please log in.')
58     return HttpResponseRedirect('login')
59 
60     return render(request, 'signup.html', context=mydict)
61 
62 #Line For Logout:
63 def logout(request):
64     auth.logout(request)
65     return redirect('/login')
66 
67 #To compare whether the user is admin or patient for the authentication,
68 def is_admin(user):
69     return user.groups.filter(name='ADMIN').exists()
70 
71 def is_patient(user):
72     return user.groups.filter(name='PATIENT').exists()
73 
74 #View For User Login page and validation.
75 def login(request):
76     if request.method == "POST":
77         username = request.POST['username']
78         password = request.POST['password']
79 
80         user = auth.authenticate(username=username, password=password)
81 
82         if user is not None:
83             auth.login(request, user)
84             return redirect("/userdashboard")
85         else:
86             messages.error(request, 'Invalid credentials !')
87             return redirect('/login')
88 
89     #View for Admin Login page and validation.
90     def adminlogin(request):
91         if request.method == "POST":
92             username = request.POST['adminusername']
93             password = request.POST['admindpassword']
94 
95             user = auth.authenticate(username=username, password=password)
96 
97         return render(request, 'adminlogin.html')
98 
99     #For User Side
100 
101     #On the basis of the user that is the side bar.
102     @login_required(login_url='/login') #decorators for authentication
103     @user_passes_test(is_patient) #Checking if the user is patient or not
104     def userbase(request):
105         patientUser = PatientUser.objects.get(user=request.user.id) #To get the profile picture of user in the side bar.
106         return render(request, 'userbase.html', {'patientUser': patientUser})
107 
108 
109     #Day for User Dashboard: Calculating the total count of report of user, user's finished and pending appointment and the doctor and service count available.
110     @login_required(login_url='/login')
111     @user_passes_test(is_patient)
112     def userdashboard(request):
113         patientUser = PatientUser.objects.get(user=request.user.id)
114         doctorscount=doctor.objects.all().count()
115         reportcount=Report.objects.filter(appointment_patientuser=patientUser).all().count()
116         servicecount=Service.objects.all().count()
117         appointment=Appointment.objects.all().filter(patientUser=patientUser,status=1)
118         finishedappointmentcount = Appointment.objects.all().filter(patientUser=patientUser, status=4).count()
119         pendingappointmentcount = Appointment.objects.all().filter(patientUser=patientUser, status=0).count()
120 
121         mydict={
122             'appointment': appointment,
123             'patientUser':patientUser,
124             'doctorscount':doctorscount,
125             'reportcount': reportcount,
126             'servicecount': servicecount,
127             'finishedappointmentcount':finishedappointmentcount,
128             'pendingappointmentcount':pendingappointmentcount,
129         }
130 
131         return render(request, 'userdashboard.html', mydict)
132 
133 
134     #View for User Appointment Page where user can navigate through appointment views.
135     @login_required(login_url='/login')
136     @user_passes_test(is_patient)
137     def userappointment(request):
138         patientUser = PatientUser.objects.get(user=request.user.id)
139 
140 
141 
142 
143 
144 
```

Ln 1, Col 1 | Spaces: 4 | UTF-8 | CRLF | Python | 3.8.10 64-bit (windows store) | idle ready

Figure 281: Sample Code of Automation Script - 8

```

File Edit Selection View Go Run Terminal Help
views.py M - views.py - views.py
home > views.py > ...
96     user = auth.authenticate(username=username, password=password)
97 
98     if user is not None:
99         auth.login(request, user)
100        return redirect("/admindashboard")
101    else:
102        messages.error(request, 'Invalid credentials !')
103        return redirect('/adminlogin')
104 
105    return render(request, 'adminlogin.html')
106 
107 #For User Side
108 
109 #On the basis of the user that is the side bar.
110 @login_required(login_url='/login') #decorators for authentication
111 @user_passes_test(is_patient) #Checking if the user is patient or not
112 def userbase(request):
113     patientUser = PatientUser.objects.get(user=request.user.id) #To get the profile picture of user in the side bar.
114     return render(request, 'userbase.html', {'patientUser': patientUser})
115 
116 
117 #Day for User Dashboard: Calculating the total count of report of user, user's finished and pending appointment and the doctor and service count available.
118 @login_required(login_url='/login')
119 @user_passes_test(is_patient)
120 def userdashboard(request):
121     patientUser = PatientUser.objects.get(user=request.user.id)
122     doctorscount=doctor.objects.all().count()
123     reportcount=Report.objects.filter(appointment_patientuser=patientUser).all().count()
124     servicecount=Service.objects.all().count()
125     appointment=Appointment.objects.all().filter(patientUser=patientUser,status=1)
126     finishedappointmentcount = Appointment.objects.all().filter(patientUser=patientUser, status=4).count()
127     pendingappointmentcount = Appointment.objects.all().filter(patientUser=patientUser, status=0).count()
128 
129     mydict={
130         'appointment': appointment,
131         'patientUser':patientUser,
132         'doctorscount':doctorscount,
133         'reportcount': reportcount,
134         'servicecount': servicecount,
135         'finishedappointmentcount':finishedappointmentcount,
136         'pendingappointmentcount':pendingappointmentcount,
137     }
138 
139     return render(request, 'userdashboard.html', mydict)
140 
141 
142 
143 
144 #View for User Appointment Page where user can navigate through appointment views.
145 @login_required(login_url='/login')
146 @user_passes_test(is_patient)
147 def userappointment(request):
148     patientUser = PatientUser.objects.get(user=request.user.id)
149 
```

Ln 1, Col 1 | Spaces: 4 | UTF-8 | CRLF | Python | 3.8.10 64-bit (windows store) | idle ready

Figure 282: Sample Code of Automation Script - 9

```

File Edit Selection View Go Run Terminal Help
views.py M X
views.py - dentalapplication - Visual Studio Code
home > views.py > ...
141 # View for User Appointment Form Page and validation.
142 @login_required(login_url='/login')
143 @user_passes_test(is_patient)
144 def user_book_appointment(request):
145     patientUser = PatientUser.objects.get(user=request.user.id)
146     return render(request,'user-book-appointment.html',{'patientUser':patientUser})
147
148 #View for User Appointment Form Page and validation.
149 @login_required(login_url='/login')
150 @user_passes_test(is_patient)
151 def user_book_appointment(request):
152     form = UserAppointmentForm()
153     patientUser = PatientUser.objects.get(user_id=request.user.id)
154     if request.method == "POST":
155         form = UserAppointmentForm(request.POST)
156         if form.is_valid():
157             appointment=form.save(commit=False)
158             appointment.patientUser=patientUser
159             appointment.status = 0
160             appointment.save()
161             messages.success(request, 'Booking done successfully. Please wait for approval.')
162             return HttpResponseRedirect('user-view-appointment')
163
164     return render(request,'user-book-appointment.html',{'form':form, 'patientUser':patientUser})
165
166 #View for seeing all of user's appointment.
167 @login_required(login_url='/login')
168 @user_passes_test(is_patient)
169 def user_view_appointment(request):
170     patientUser = PatientUser.objects.get(user=request.user.id) #to get the profile picture in the sidebar
171     appointment = Appointment.objects.filter(patientUser=patientUser).all()
172     return render(request,'user-view-appointment.html',{'appointment':appointment , 'patientUser':patientUser})
173
174 #View for seeing all the doctors available.
175 @login_required(login_url='/login')
176 @user_passes_test(is_patient)
177 def user_view_doctors(request):
178     patientUser = PatientUser.objects.get(user=request.user.id)
179     doctors = Doctor.objects.all()
180     return render(request,'user-view-doctors.html',{'doctors':doctors , 'patientUser':patientUser})
181
182 #View for seeing all of user's report.
183 @login_required(login_url='/login')
184 @user_passes_test(is_patient)
185 def user_view_reports(request):
186     patientUser = PatientUser.objects.get(user=request.user.id)
187     reports = Report.objects.filter(appointment__patientUser=patientUser).all()
188     return render(request,'user-view-reports.html',{'reports':reports , 'patientUser':patientUser})
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236

```

In 1, Col 1 | Spaces: 4 | UTF-8 | CRLF | Python | 3.8.10 64-bit (windows store) | idle ready | ⚡

Figure 283: Sample Code of Automation Script - 10

```

File Edit Selection View Go Run Terminal Help
views.py M X
views.py - dentalapplication - Visual Studio Code
home > views.py > ...
188 #View for seeing all of the services available.
189 @login_required(login_url='/login')
190 @user_passes_test(is_patient)
191 def user_view_services(request):
192     patientUser = PatientUser.objects.get(user=request.user.id) #to get logged in user's image
193     services = Service.objects.all()
194     return render(request,'user-view-services.html',{'services':services , 'patientUser':patientUser})
195
196 #View for seeing all of user's profile.
197 @login_required(login_url='/login')
198 @user_passes_test(is_patient)
199 def user_view_profile(request):
200     patientUser = PatientUser.objects.get(user=request.user.id)
201     Patient = Patient.objects.filter(user=request.user)
202     return render(request,'user-view-profile.html',{'patientUser':patientUser, 'Patient':Patient})
203
204 #View to cancel user's pending or confirmed appointment.
205 @login_required(login_url='/login')
206 @user_passes_test(is_patient)
207 def user_cancel_appointment(request, pk):
208     appointment = Appointment.objects.get(pk=pk)
209     appointment.status=-1
210     appointment.save()
211     messages.warning(request, 'Booking cancelled.')
212     return redirect('user-view-appointment')
213
214 #View for user to edit their data and its validation.
215 @login_required(login_url='/login')
216 @user_passes_test(is_patient)
217 def user_edit_profile(request):
218     patientUser = PatientUser.objects.get(user=request.user.id) #to get the profile picture in the sidebar
219     u_form = UpdatePatientUserForm(instance=request.user.patientuser)
220     p_form = UpdatePatientUserForm(instance=request.user.patientuser)
221
222     if request.method == "POST":
223         u_form = UpdatePatientUserForm(request.POST, instance=request.user)
224         p_form = UpdatePatientUserForm(request.POST, request.FILES, instance=request.user.patientuser)
225         if u_form.is_valid() and p_form.is_valid():
226             u_form.save()
227             p_form.save()
228             messages.success(request, 'Account successfully updated.')
229             return redirect('user-view-profile')
230
231     context = { 'u_form': u_form, 'p_form': p_form, 'patientUser':patientUser }
232     return render(request,'user-edit-profile.html', context)
233
234
235 #For Admin Side
236

```

In 1, Col 1 | Spaces: 4 | UTF-8 | CRLF | Python | 3.8.10 64-bit (windows store) | idle ready | ⚡

Figure 284: Sample Code of Automation Script - 11

Figure 285: Sample Code of Automation Script - 12

This screenshot shows a Visual Studio Code window displaying Python code for a dental application. The file is named `views.py`. The code includes various views for managing appointments, such as viewing all appointments, deleting an appointment, approving a pending appointment, rejecting a pending appointment, and canceling a confirmed appointment. It uses Django's authentication system with `@login_required` and `@user_passes_test(is_admin)` decorators. The code is well-structured with comments explaining the purpose of each view.

```
File Edit Selection View Go Run Terminal Help
views.py - dentalapplication - Visual Studio Code

home > views.py > ...
245
246 #View for seeing all of user appointment.
247 @login_required(login_url='/admin/login')
248 @user_passes_test(is_admin)
249 def admin_view_appointment(request):
250     appointments = Appointment.objects.all().filter(status__gte=1)
251     return render(request,'admin_view_appointment.html', {'appointment':appointments})
252
253 #View for deleting the selected appointment.
254 @login_required(login_url='/admin/login')
255 @user_passes_test(is_admin)
256 def delete_appointment (request, pk):
257     appointments = Appointment.objects.get(pk=pk)
258     appointments.delete()
259     messages.danger(request, 'Appointment deleted.')
260     return redirect('admin_view_appointment')
261
262 #View for viewing all the pending appointment list.
263 @login_required(login_url='/admin/login')
264 @user_passes_test(is_admin)
265 def admin_approve_appointment(request):
266     appointments = Appointment.objects.all().filter(status=0)
267     return render(request,'admin_approve_appointment.html', {'appointment':appointments})
268
269 #View for approving the pending appointment.
270 @login_required(login_url='/admin/login')
271 @user_passes_test(is_admin)
272 def approve_appointment (request, pk):
273     appointments = Appointment.objects.get(pk=pk)
274     appointments.status=1
275     appointments.save()
276     messages.success(request, 'Booking approved.')
277     return redirect('admin_view_appointment')
278
279 #View for rejecting the pending appointment.
280 @login_required(login_url='/admin/login')
281 @user_passes_test(is_admin)
282 def reject_appointment (request, pk):
283     appointments = Appointment.objects.get(pk=pk)
284     appointments.status=2
285     appointments.save()
286     messages.warning(request, 'Booking rejected.')
287     return redirect('admin_view_appointment')
288
289 #View for cancelling the confirmed appointment.
290 @login_required(login_url='/admin/login')
291 @user_passes_test(is_admin)
292 def admin_cancel_appointment (request, pk):
293     appointments = Appointment.objects.get(pk=pk)
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
```

Figure 286: Sample Code of Automation Script - 13

The screenshot shows a Visual Studio Code window with the file 'views.py' open. The code is a Python script containing various functions for managing appointments, doctors, and reports. It includes imports for Django's User and Doctor models, as well as the 'messages' module for displaying success and warning messages. Functions include:

- `cancel_appointment`: Cancels an appointment by setting its status to 3.
- `finish_appointment`: Finishes an appointment by setting its status to 4.
- `view_doctors`: Displays a list of all doctors in the database.
- `remove_doctor`: Deletes a doctor from the database.
- `add_doctor`: Adds a new doctor to the database, validating the form.
- `update_doctor`: Updates an existing doctor's information, validating the form.
- `view_reports`: Displays a list of user reports.
- `remove_report`: Deletes a report from the database.
- `add_report`: Adds a new report to the database, validating the form.

The code uses Django's built-in authentication system with login_required and user_passes_test decorators. It also uses Django's generic view classes like `ListView` and `DetailView`.

Figure 287: Sample Code of Automation Script - 14

This screenshot shows another instance of Visual Studio Code with the same 'views.py' file. The code structure is very similar to Figure 287, but the specific functions and their implementations are different. For example, the `cancel_appointment` function is renamed to `cancel_booking`. The `add_report` function is renamed to `add_complaint`. The `remove_report` function is renamed to `remove_complaint`. The `update_doctor` function is renamed to `update_physician`. The `view_reports` function is renamed to `view_complaints`. The `remove_physician` function is renamed to `remove_reporter`. The `add_physician` function is renamed to `add_complaint`. The `update_physician` function is renamed to `update_complaint`. The `view_complaints` function is renamed to `view_reports`. The `remove_complaint` function is renamed to `remove_report`. The `add_complaint` function is renamed to `add_report`. The `update_complaint` function is renamed to `update_report`.

Figure 288: Sample Code of Automation Script - 15

```

File Edit Selection View Go Run Terminal Help
views.py M X
home > views.py > ...
425     messages.success(request, 'Report added successfully.')
426     return HttpResponseRedirect('/admin_view_reports')
427
428 context = {'form': form, 'appointment': appointment}
429 return render(request,'admin_add_reports.html', context)
430
431 #view for updating the selected report and validation.
432 @login_required(login_url='/adminlogin')
433 @user_passes_test(is_admin)
434 def update_report(request, pk):
435     report = Report.objects.get(id=pk)
436     form = ReportForm(instance=report)
437     if request.method == "POST":
438         form = ReportForm(request.POST, instance=report)
439         if form.is_valid():
440             form.save()
441             messages.success(request, 'Report updated.')
442             return HttpResponseRedirect('/admin_view_reports')
443         context = {'form': form}
444     return render(request,'admin_add_reports.html', context)
445
446
447 #view for seeing all the users
448 @login_required(login_url='/adminlogin')
449 @user_passes_test(is_admin)
450 def admin_view_users(request):
451     patientUsers = PatientUser.objects.all
452     return render(request,'admin_view_users.html', {'patientUser':patientUsers})
453
454 #view for removing the selected users from the database
455 @login_required(login_url='/adminlogin')
456 @user_passes_test(is_admin)
457 def delete_patientUser(request, pk):
458     user = User.objects.get(pk=pk)
459     user.delete()
460     messages.warning(request, 'User has been removed.')
461     return redirect('admin_view_users')
462
463 #view for seeing all the services
464 @login_required(login_url='/adminlogin')
465 @user_passes_test(is_admin)
466 def admin_view_services(request):
467     services = Service.objects.all
468     return render(request,'admin_view_services.html', {'services':services})
469
470
471 #view for adding service to the database and validation
472 @login_required(login_url='/adminlogin')
473 @user_passes_test(is_admin)
474 def admin_add_services(request):
475     form = ServiceForm
476     if request.method == "POST":
477         form = ServiceForm(request.POST)
478         if form.is_valid():
479             form.save()
480             messages.success(request, 'Service added successfully.')
481             return HttpResponseRedirect('/admin_view_services')
482
483
484     return render(request,'admin_add_services.html', {'form':form})
485
486 #view for updating service in the database and validation
487 @login_required(login_url='/adminlogin')
488 @user_passes_test(is_admin)
489 def update_service(request, pk):
490     services = Service.objects.get(id=pk)
491     form = ServiceForm(instance=services)
492     if request.method == "POST":
493         form = ServiceForm(request.POST, instance=services)
494         if form.is_valid():
495             form.save()
496             messages.success(request, 'Service updated.')
497             return HttpResponseRedirect('/admin_view_services')
498         context = {'form': form}
499     return render(request,'admin_add_services.html', context)
500
501
502 #view for deleting the selected service
503 @login_required(login_url='/adminlogin')
504 @user_passes_test(is_admin)
505 def delete_service(request, pk):
506     services = Service.objects.get(pk=pk)
507     services.delete()
508     messages.warning(request, 'Service removed.')
509     return redirect('admin_view_services')
510
511

```

Line 1, Col 1 | Spaces:4 | UTF-8 | CRLF | Python | 3.8.10 64-bit (windows store) | idle ready

Figure 289: Sample Code of Automation Script - 16

```

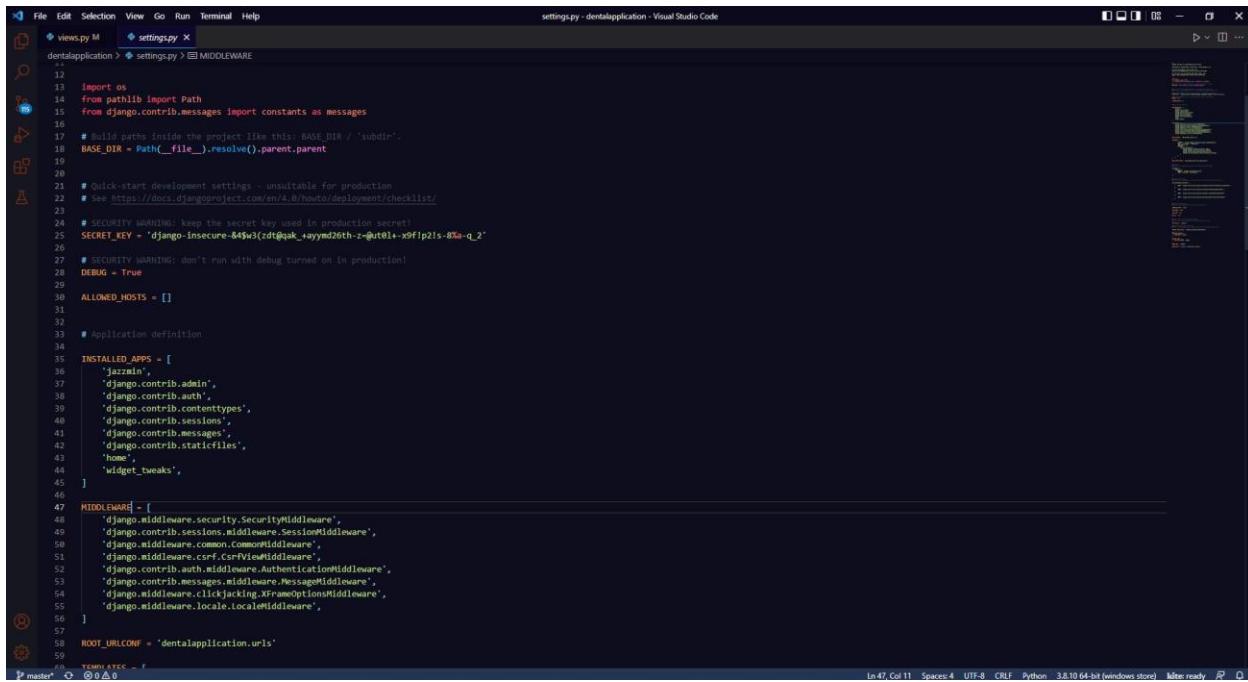
File Edit Selection View Go Run Terminal Help
views.py M X
home > views.py > ...
467     services = Service.objects.all
468     return render(request,'admin_view_services.html', {'services':services})
469
470
471 #view for adding service to the database and validation
472 @login_required(login_url='/adminlogin')
473 @user_passes_test(is_admin)
474 def admin_add_services(request):
475     form = ServiceForm
476     if request.method == "POST":
477         form = ServiceForm(request.POST)
478         if form.is_valid():
479             form.save()
480             messages.success(request, 'Service added successfully.')
481             return HttpResponseRedirect('/admin_view_services')
482
483
484     return render(request,'admin_add_services.html', {'form':form})
485
486 #view for updating service in the database and validation
487 @login_required(login_url='/adminlogin')
488 @user_passes_test(is_admin)
489 def update_service(request, pk):
490     services = Service.objects.get(id=pk)
491     form = ServiceForm(instance=services)
492     if request.method == "POST":
493         form = ServiceForm(request.POST, instance=services)
494         if form.is_valid():
495             form.save()
496             messages.success(request, 'Service updated.')
497             return HttpResponseRedirect('/admin_view_services')
498         context = {'form': form}
499     return render(request,'admin_add_services.html', context)
500
501
502 #view for deleting the selected service
503 @login_required(login_url='/adminlogin')
504 @user_passes_test(is_admin)
505 def delete_service(request, pk):
506     services = Service.objects.get(pk=pk)
507     services.delete()
508     messages.warning(request, 'Service removed.')
509     return redirect('admin_view_services')
510
511

```

Line 1, Col 1 | Spaces:4 | UTF-8 | CRLF | Python | 3.8.10 64-bit (windows store) | idle ready

Figure 290: Sample Code of Automation Script - 17

Settings.py



```

File Edit Selection View Go Run Terminal Help
views.py M settings.py X
dentalapplication > settings.py > MIDDLEWARE
...
12
13 import os
14 from pathlib import Path
15 from django.contrib.messages import constants as messages
16
17 # Build paths inside the project like this: BASE_DIR / 'subdir'.
18 BASE_DIR = Path(__file__).resolve().parent.parent
19
20
21 # Quick-start development settings - unsuitable for production
22 # See https://docs.djangoproject.com/en/4.0/howto/deployment/checklist/
23
24 # SECURITY WARNING: keep the secret key used in production secret!
25 SECRET_KEY = 'django-insecure-4&4w3(dig@ak_ayynd26th-z@ut0l+>OfIp2!s-8%q_2'
26
27 # SECURITY WARNING: don't run with debug turned on in production!
28 DEBUG = True
29
30 ALLOWED_HOSTS = []
31
32
33 # Application definition
34
35 INSTALLED_APPS = [
36     'jazzmin',
37     'django.contrib.admin',
38     'django.contrib.auth',
39     'django.contrib.contenttypes',
40     'django.contrib.sessions',
41     'django.contrib.messages',
42     'django.contrib.staticfiles',
43     'home',
44     'widget_tweaks',
45 ]
46
47 MIDDLEWARE = [
48     'django.middleware.security.SecurityMiddleware',
49     'django.contrib.sessions.middleware.SessionMiddleware',
50     'django.middleware.common.CommonMiddleware',
51     'django.middleware.csrf.CsrfViewMiddleware',
52     'django.contrib.auth.middleware.AuthenticationMiddleware',
53     'django.contrib.messages.middleware.MessageMiddleware',
54     'django.middleware.clickjacking.XFrameOptionsMiddleware',
55     'django.middleware.locale.LocaleMiddleware',
56 ]
57
58 ROOT_URLCONF = 'dentalapplication.urls'
59

```

TITLE: YACEC - F

In 47, Col 11 | Spaces: 4 | UTF-8 | CRLF | Python 3.8.10 64-bit (Windows Store) | Idle ready

Figure 291: Sample Code of Automation Script - 18

```

File Edit Selection View Go Run Terminal Help
views.py M settings.py X
dentalapplication > settings.py > MIDDLEWARE
59
60     TEMPLATES = [
61         {
62             'BACKEND': 'django.template.backends.django.DjangoTemplates',
63             'DIRS': [BASE_DIR / "templates"],
64             'APP_DIRS': True,
65             'OPTIONS': {
66                 'context_processors': [
67                     'django.template.context_processors.debug',
68                     'django.template.context_processors.request',
69                     'django.contrib.auth.context_processors.auth',
70                     'django.contrib.messages.context_processors.messages',
71                 ],
72             },
73         },
74     ]
75
76     MIGRATION_MODULES = {
77         'auth': 'dentalapplication.migrations',
78     }
79
80     # database
81     # https://docs.djangoproject.com/en/4.0/ref/settings/#databases
82     DATABASES = {
83         'default': {
84             'ENGINE': 'django.db.backends.sqlite3',
85             'NAME': BASE_DIR / 'db.sqlite3',
86         }
87     }
88
89     # password validation
90     # https://docs.djangoproject.com/en/4.0/ref/settings/#auth-password-validation
91
92     AUTH_PASSWORD_VALIDATORS = [
93         {
94             'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
95         },
96         {
97             'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
98         },
99         {
100             'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
101         },
102         {
103             'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator',
104         },
105     ],
106 ]
107

```

In 47, Col 11 Spaces: 4 UTF-8 CRLF Python 3.8.10 64-bit (windows store) line: ready

Figure 292: Sample Code of Automation Script - 19

```

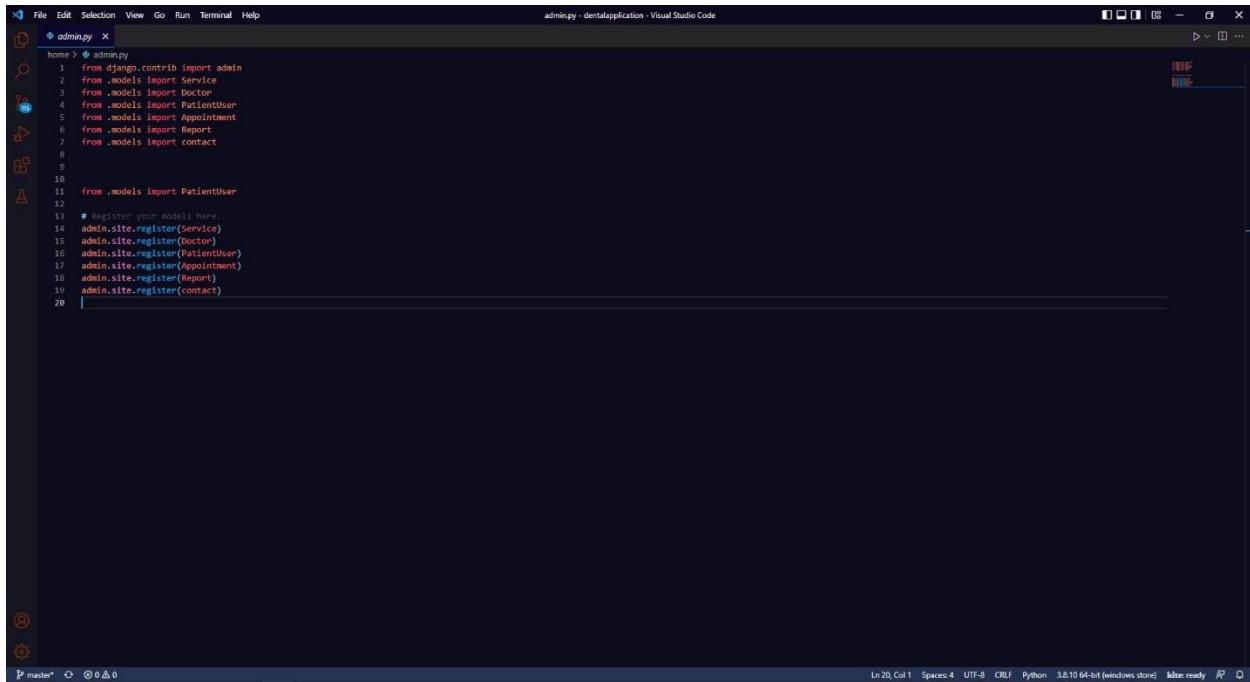
File Edit Selection View Go Run Terminal Help
views.py M settings.py X
dentalapplication > settings.py > MIDDLEWARE
101     'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
102     },
103     {
104         'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator',
105     },
106 ]
107
108 # Internationalization
109 # https://docs.djangoproject.com/en/4.0/topics/i18n/
110
111 LANGUAGE_CODE = 'en-us'
112
113 TIME_ZONE = 'UTC'
114
115 USE_I18N = True
116
117 USE_TZ = True
118
119
120 # static files (CSS, JavaScript, Images)
121 # https://docs.djangoproject.com/en/4.0/howto/static-files/
122
123 STATIC_URL = '/static/'
124
125 # Default primary key field type
126 # https://docs.djangoproject.com/en/4.0/ref/settings/#default-auto-field
127
128 DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'
129
130
131 # Newly added
132 STATICFILES_DIRS = [
133     BASE_DIR / "static"
134 ]
135
136 MESSAGE_TAGS = {
137     messages.ERROR: 'danger'
138 }
139
140 MEDIA_URL = '/media/'
141
142 MEDIA_ROOT = os.path.join(BASE_DIR, 'static')
143
144

```

In 47, Col 11 Spaces: 4 UTF-8 CRLF Python 3.8.10 64-bit (windows store) line: ready

Figure 293: Sample Code of Automation Script - 20

Admin.py



The screenshot shows a Visual Studio Code window with the file 'admin.py' open. The code is a Python script used for registering models in the Django admin interface. It imports various models from the 'models' directory and registers them with the Admin site.

```
File Edit Selection View Go Run Terminal Help
admin.py - dentalapplication - Visual Studio Code
home > admin.py
1  from django.contrib import admin
2  from .models import Service
3  from .models import Doctor
4  from .models import PatientUser
5  from .models import Appointment
6  from .models import Report
7  from .models import contact
8
9
10
11 from .models import PatientUser
12 # Register your models here.
13 admin.site.register(Service)
14 admin.site.register(Doctor)
15 admin.site.register(PatientUser)
16 admin.site.register(Appointment)
17 admin.site.register(Report)
18 admin.site.register(contact)
19
20
```

File master* ① ② ③ ④ ⑤ ⑥ ⑦ ⑧ ⑨ ⑩ ⑪ ⑫ ⑬ ⑭ ⑮ ⑯ ⑰ ⑱ ⑲ ⑳

Lr 20, Col 1 Spaces: 4 UTF-8 CRLF Python 3.8.10 64-bit (windows store) Idle ready ⚙

Figure 294: Sample Code of Automation Script - 21

7.6 Appendix F: Designs

[Continued from here.](#)

UML, which stands for Unified Modelling Language, is a way to visually represent the architecture, design, and implementation of complex software systems. (Lucid, 2022) There are thousands of lines of code in an application, and it's tough to keep track of the links and hierarchies inside a software system when developing code. The software system is divided into components and subcomponents using UML diagrams.

Use Case Diagram: A use case diagram is a way to summarize details of a system and the users within that system. (TechTarget, 2020) It is often depicted as a graphical representation of the interactions between various parts in a system. Use case diagrams define the events in a system and how they flow, but they do not describe how those events are implemented.

Collaboration Diagram: A collaboration diagram, also known as a communication diagram, is an illustration of the relationships and interactions among software objects in the Unified Modelling Language (UML). (Lewis, 2019) These diagrams may be used to depict the dynamic behaviour of a certain use case as well as identify the role of each component.

Activity Diagram: Another significant behavioural diagram in UML diagram is the activity diagram, which is used to explain dynamic characteristics of the system. Activity diagram is essentially an advanced version of flow chart that modelling the flow from one activity to another activity. (Paradigm, 2022)

Sequence Diagram: UML Sequence Diagrams are interaction diagrams that show how processes are performed. They document the interaction of items in the context of a collaborative effort. Sequence Diagrams are time focused, and they graphically express the sequence of the interaction by utilizing the vertical axis of the diagram to indicate time, what messages are conveyed, and when. (Paradigm, 2022)

Class Diagram: In software engineering, a class diagram in the Unified Modelling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects. (Paradigm, 2022)

ER Diagram: ER Diagram stands for Entity Relationship Diagram, also known as ERD is a diagram that displays the relationship of entity sets stored in a database. (Peterson, 2022) In other words, ER diagrams aid in the explanation of database logical structure. Entities, attributes, and relationships are the three main elements around which ER diagrams are built.

7.6.1 Functional UML Diagrams

[Click Here to see User side Functional UML Diagram](#)

Admin

11. Login to the system

High Level Use Case Description

Use Case:	Admin Login
Actor:	Admin
Description:	In order to use the admin side system and its features, the admin must log in to the system. The login details is then verified by the system.

Table 40: High Level Use Case – Admin Login

Use Case Diagram

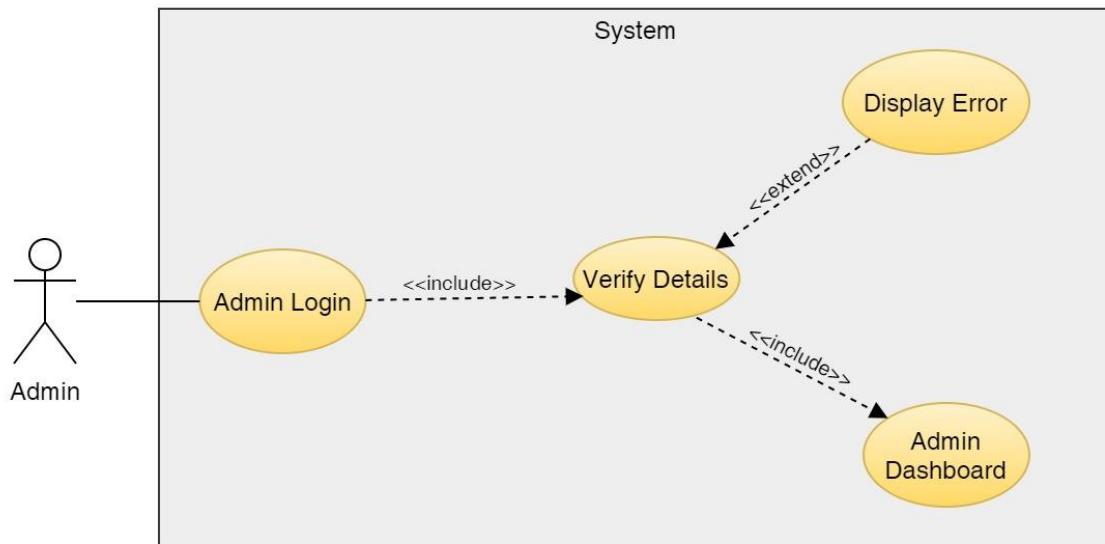


Figure 295: Use Case Diagram – Admin Login

Collaboration Diagram

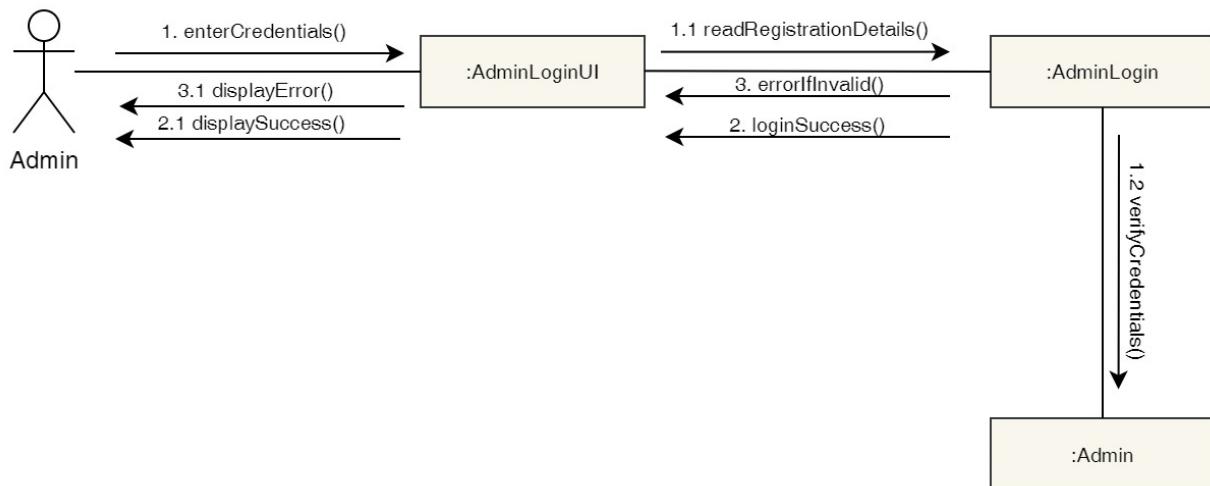


Figure 296: Collaboration Diagram – Admin Login

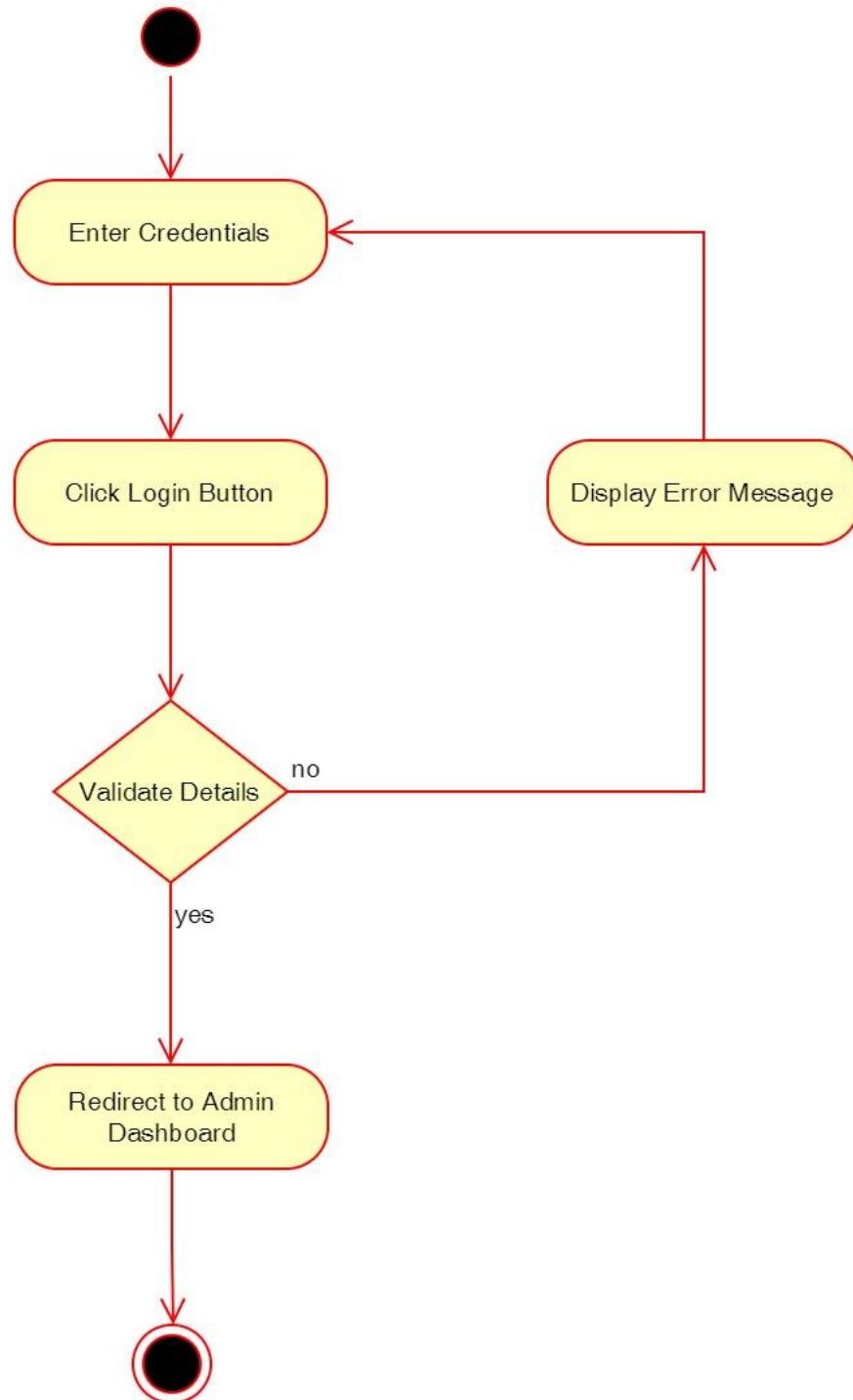
Activity Diagram

Figure 297: Activity Diagram – Admin Login

Sequence Diagram

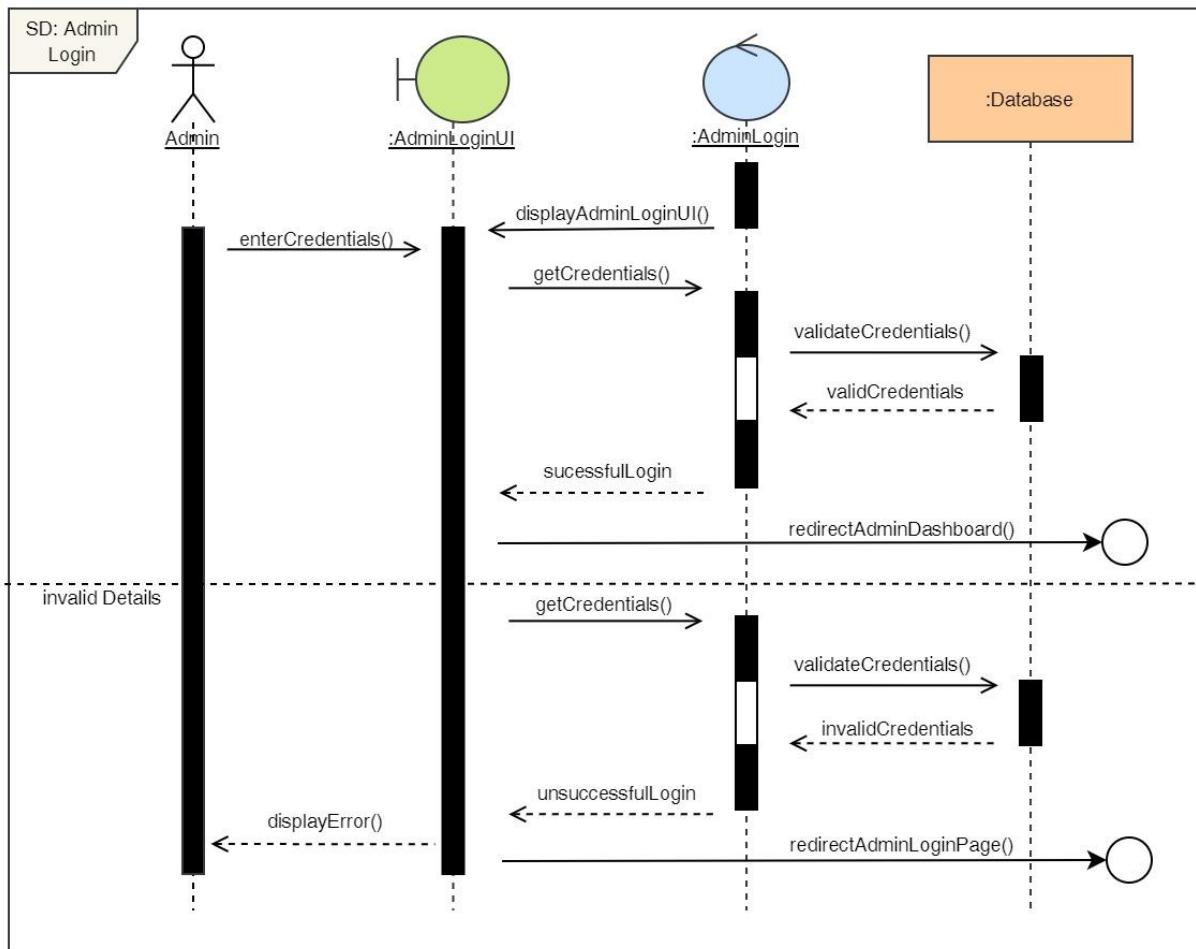


Figure 298: Sequence Diagram – Admin Login

12. View all users

High Level Use Case Description

Use Case:	View all users
Actor:	Admin
Description:	Admin can view all the user that are registered to the system.

Table 41: High Level Use Case – Admin View Users

Use Case Diagram

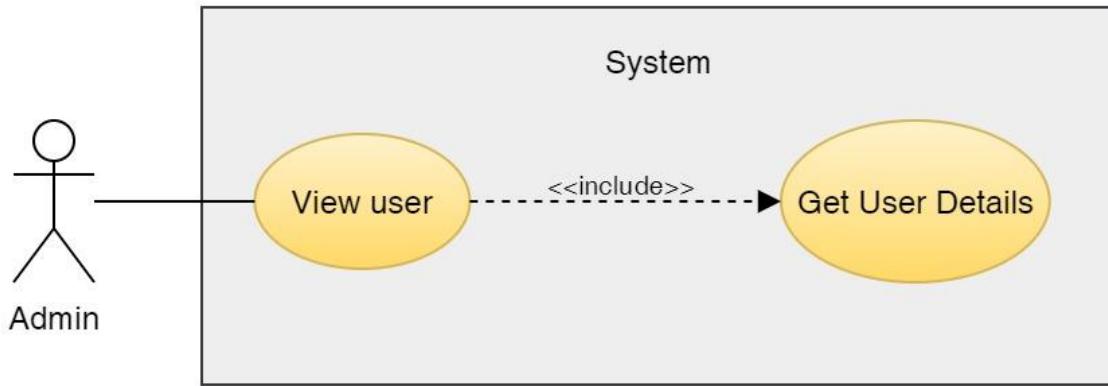


Figure 299: Use Case Diagram – Admin View Users

Collaboration Diagram

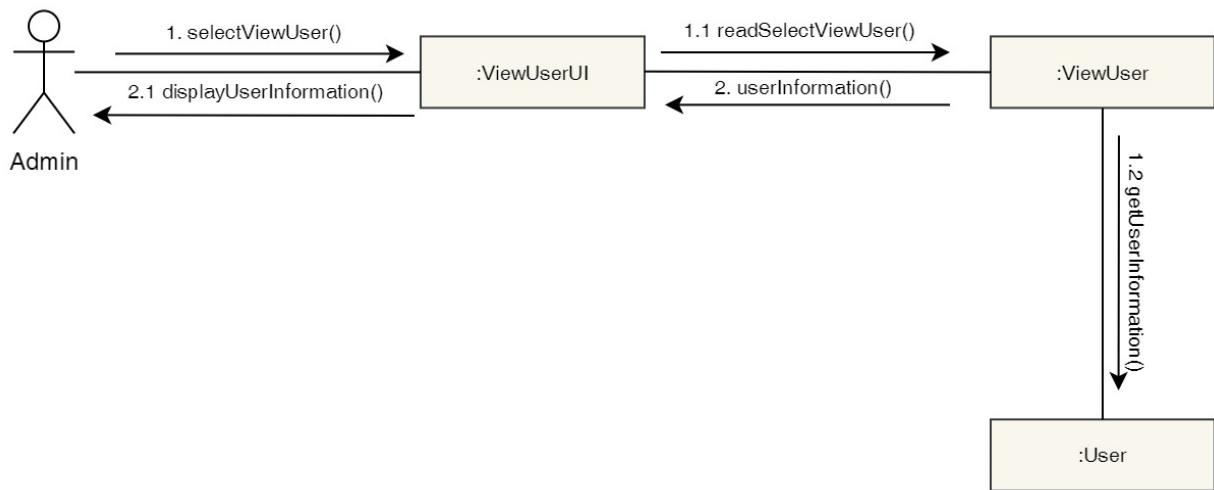


Figure 300: Collaboration Diagram – Admin View Users

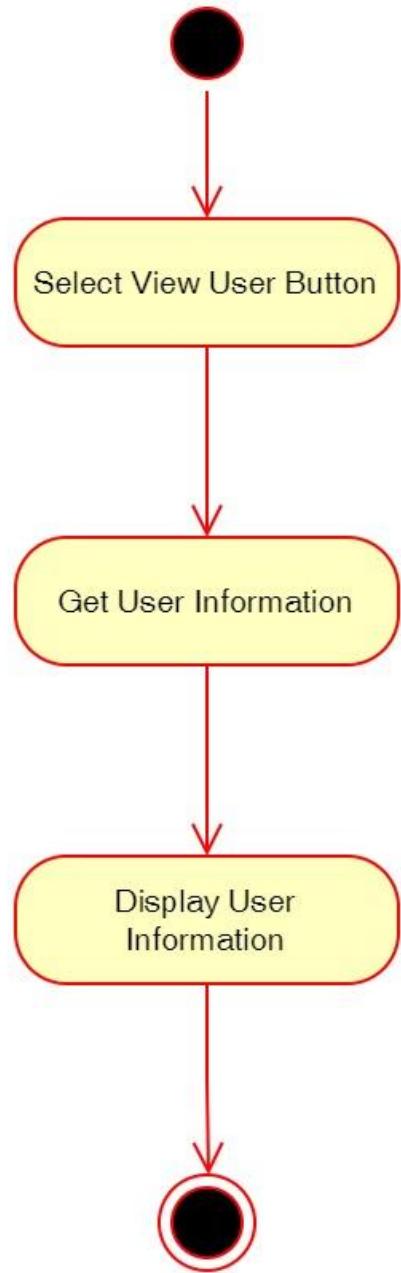
Activity Diagram

Figure 301: Activity Diagram – Admin View Users

Sequence Diagram

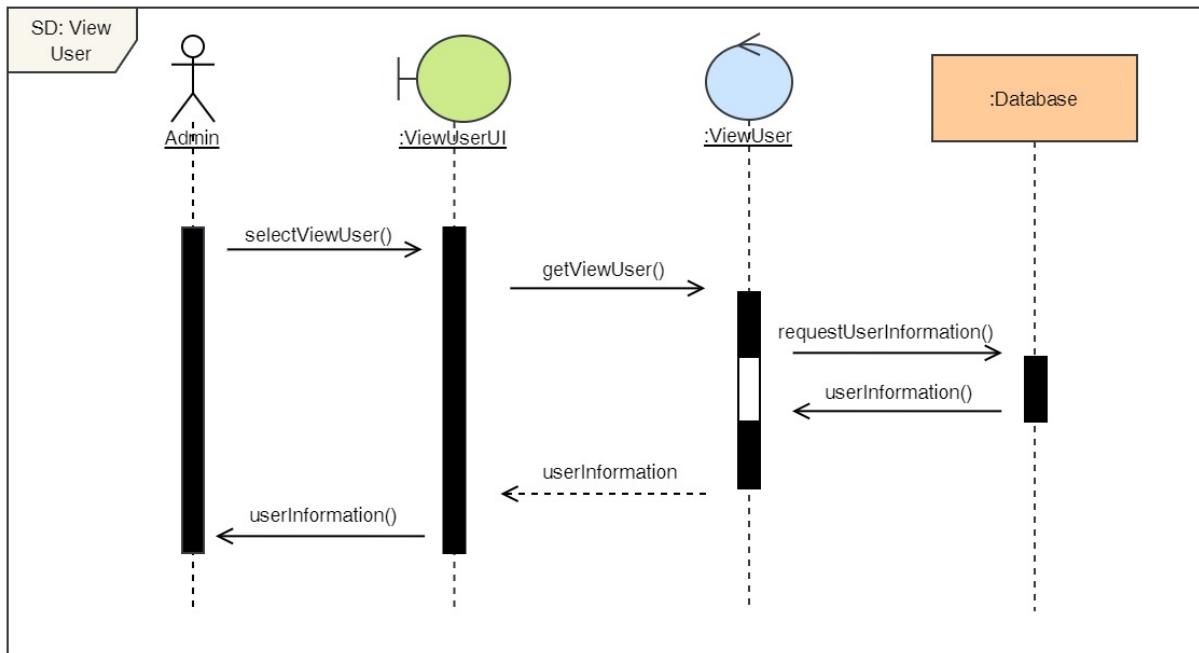


Figure 302: Sequence Diagram – Admin View Users

13. Delete users

High Level Use Case Description

Use Case:	De-Register User
Actor:	Admin
Description:	Admin can delete existing users from the system.

Table 42: High Level Use Case – Admin Delete Users

Use Case Diagram

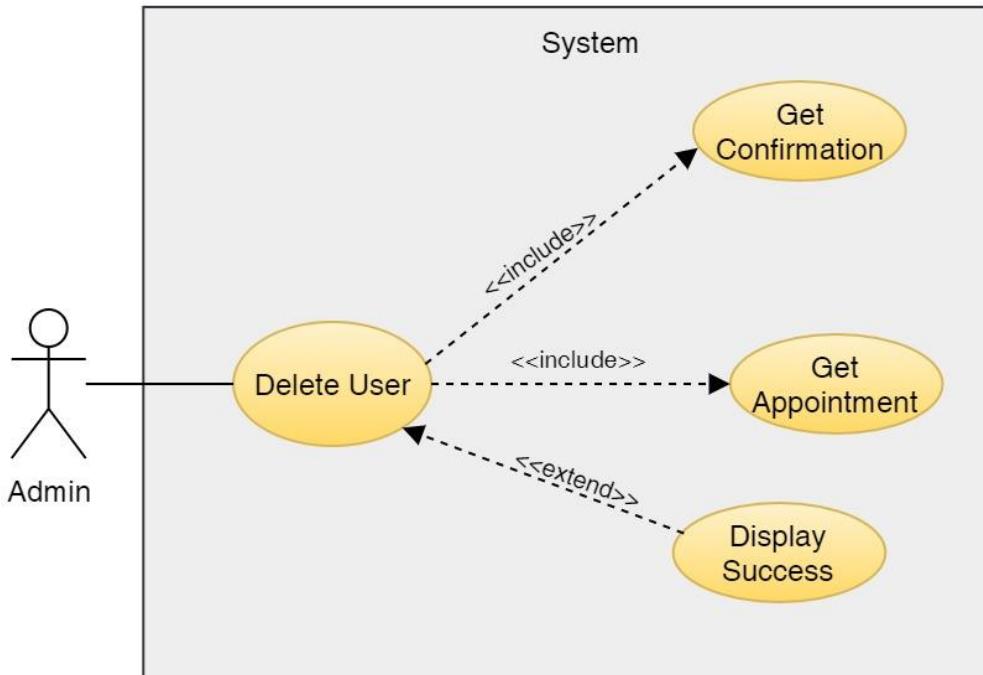


Figure 303: Use Case Diagram – Admin Delete Users

Collaboration Diagram

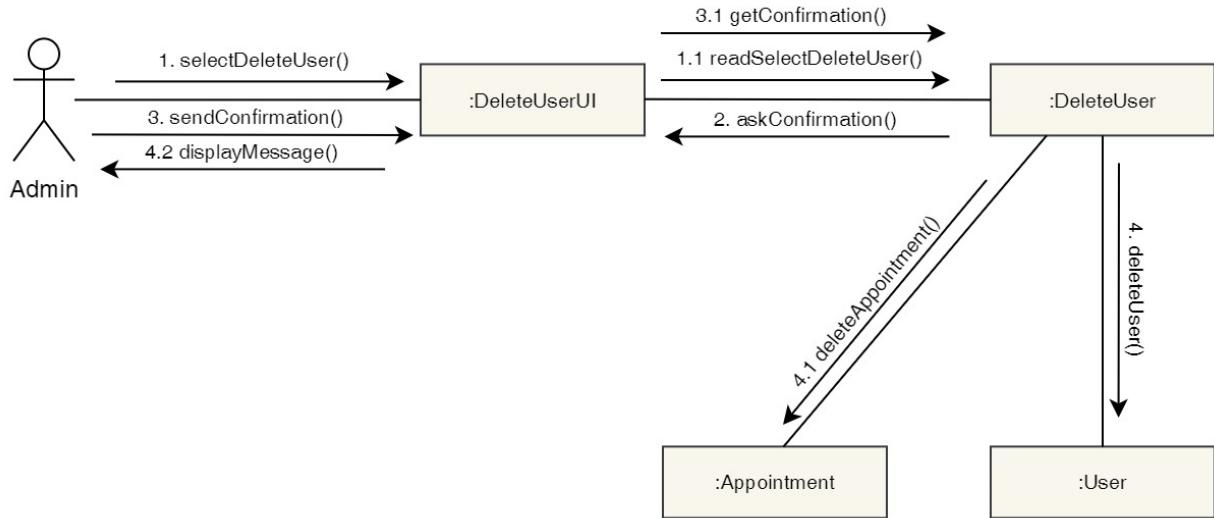
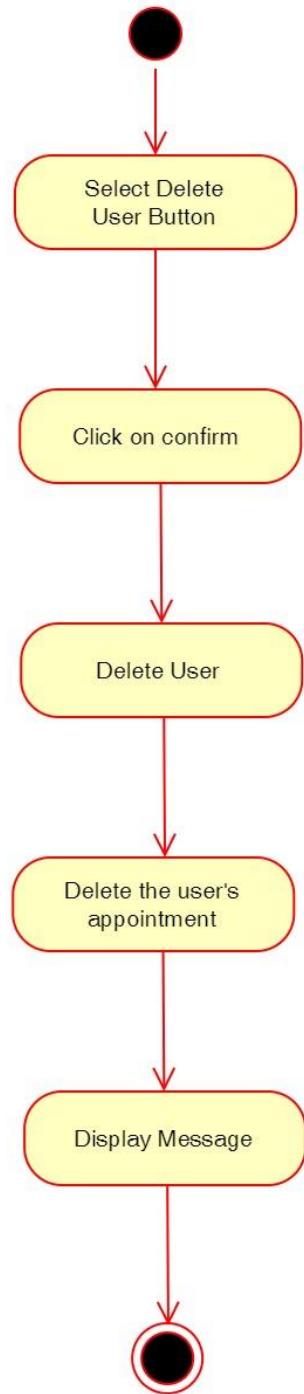


Figure 304: Collaboration Diagram – Admin Delete Users

Activity Diagram*Figure 305: Activity Diagram – Admin Delete Users*

Sequence Diagram

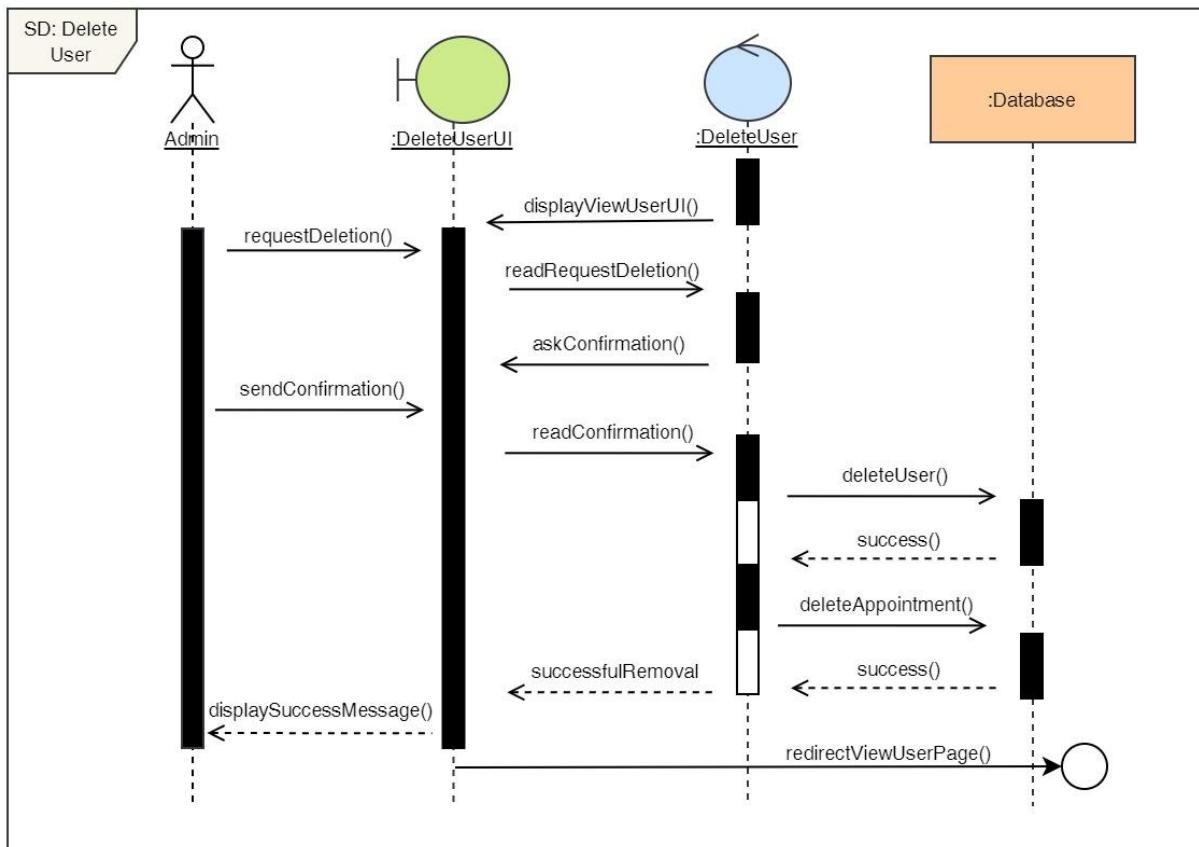


Figure 306: Sequence Diagram – Admin Delete Users

14. View all doctors

High Level Use Case Description

Use Case:	View doctors
Actor:	Admin
Description:	Admin can view all the existing doctors in the system.

Table 43: High Level Use Case – Admin View Doctors

Use Case Diagram

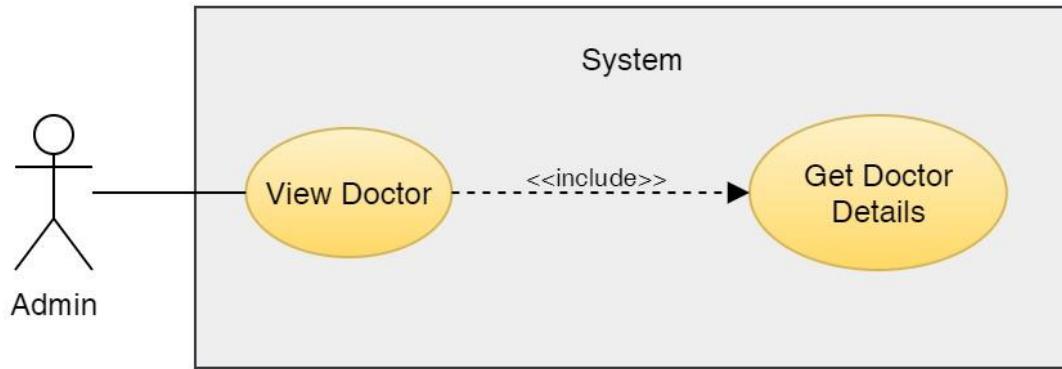


Figure 307: Use Case Diagram – Admin View Doctors

Collaboration Diagram

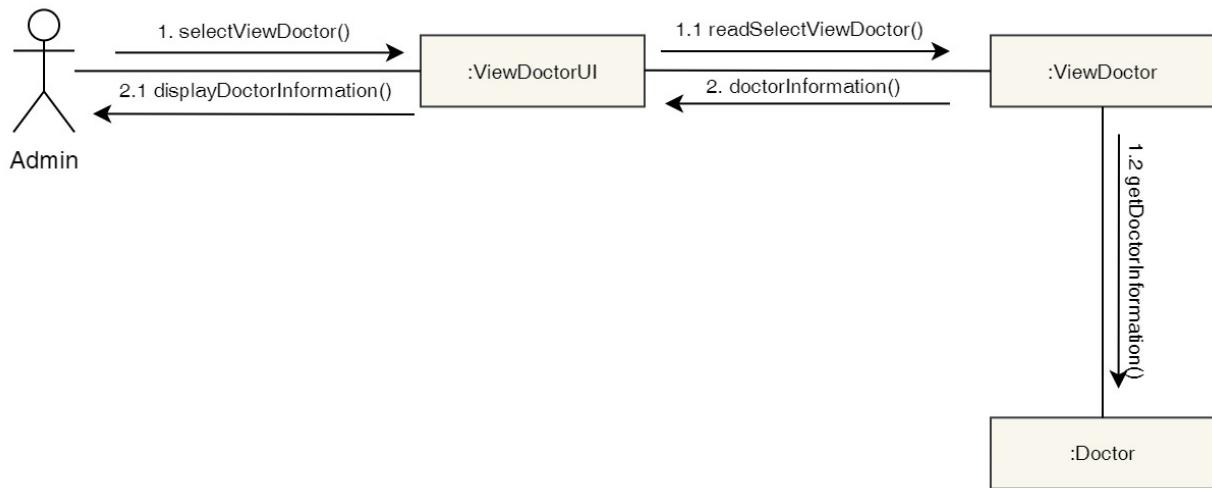


Figure 308: Collaboration Diagram – Admin View Doctors

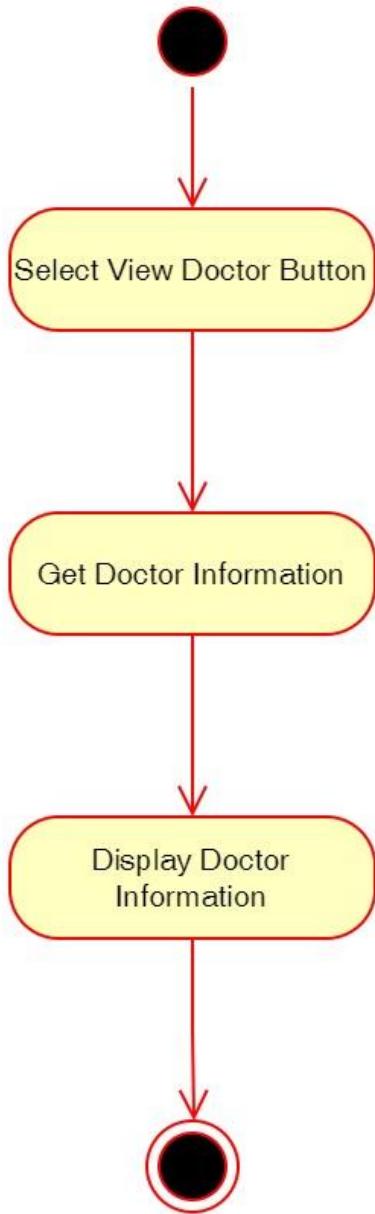
Activity Diagram

Figure 309: Activity Diagram – Admin View Doctors

Sequence Diagram

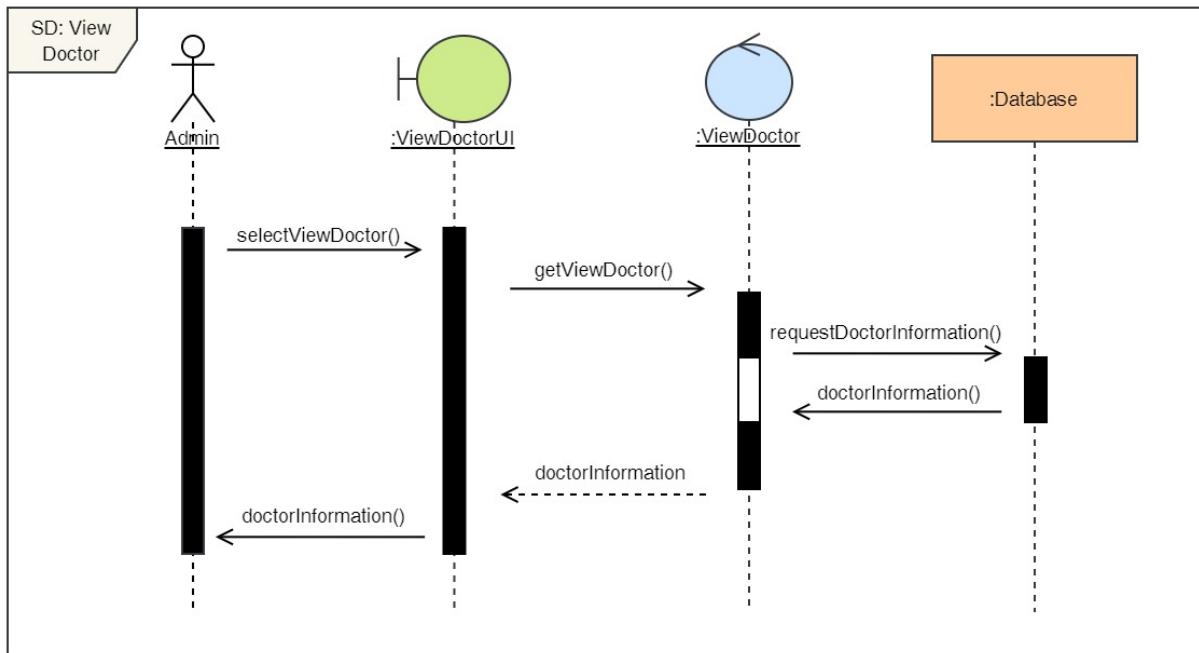


Figure 310: Sequence Diagram – Admin View Doctors

15. Add new doctors

High Level Use Case Description

Use Case:	Add Doctors
Actor:	Admin
Description:	Admin can add new doctors to the system along with their description.

Table 44: High Level Use Case – Admin Add Doctors

Use Case Diagram

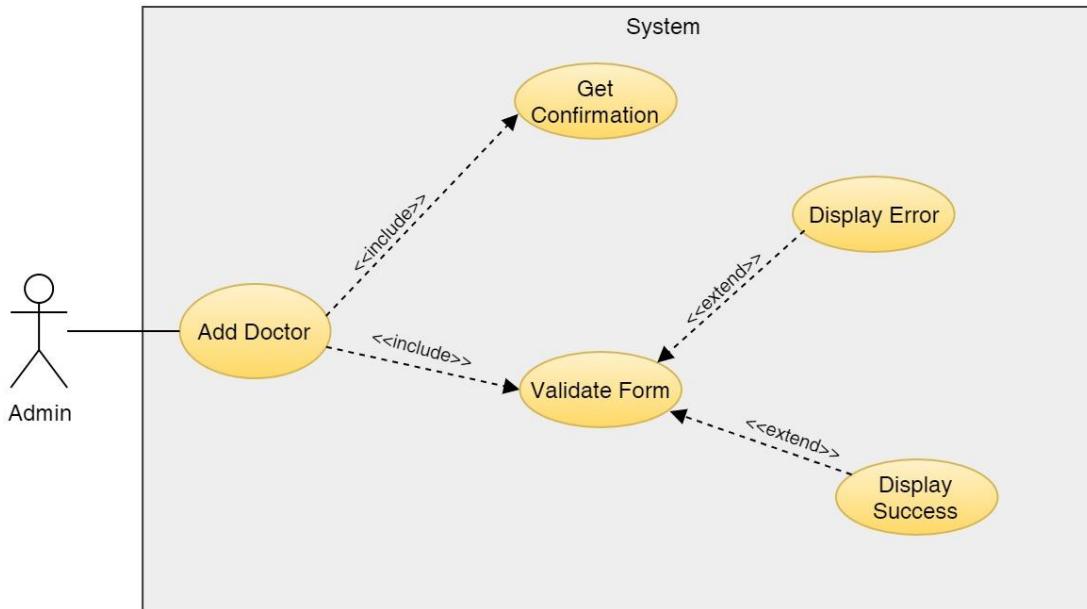


Figure 311: Use Case Diagram – Admin Add Doctors

Collaboration Diagram

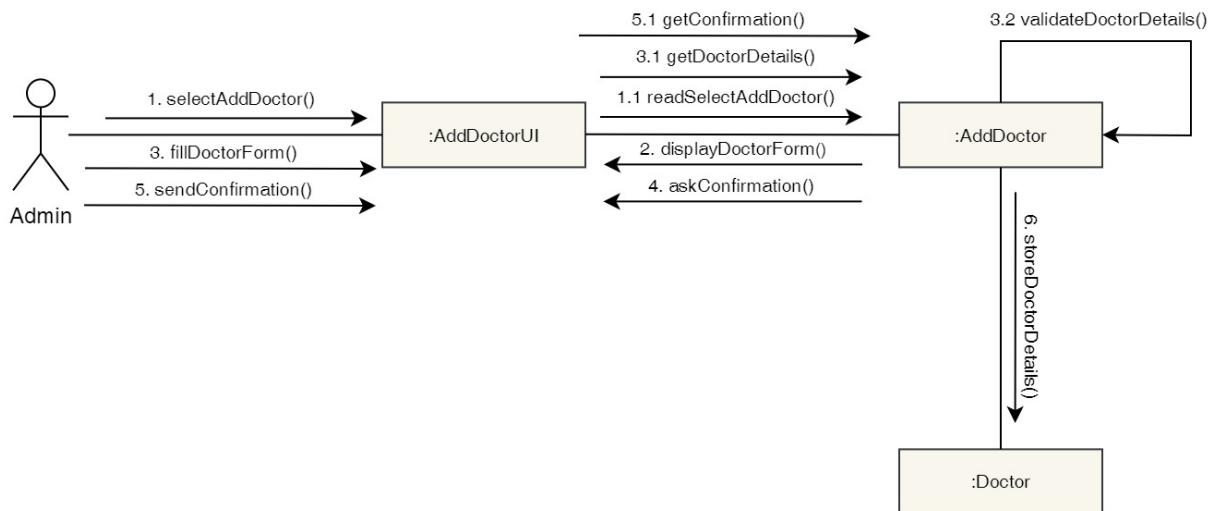
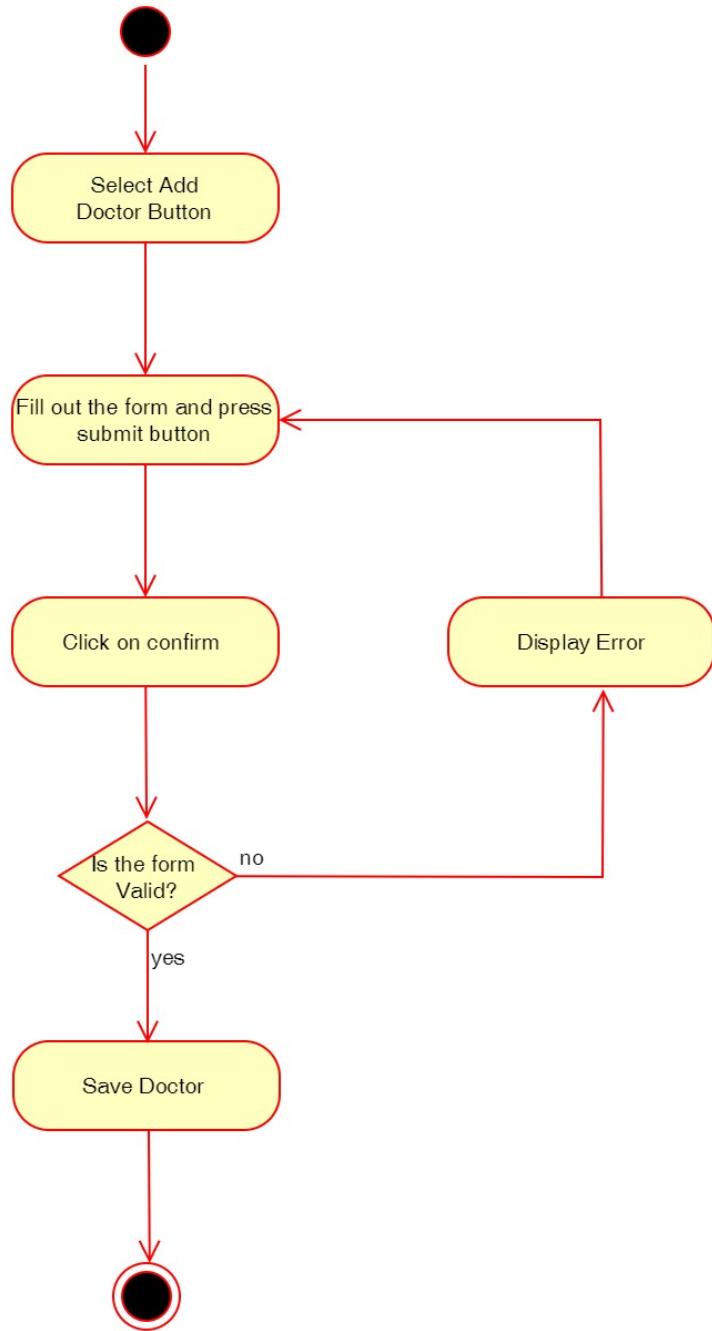


Figure 312: Collaboration Diagram – Admin Add Doctors

Activity Diagram*Figure 313: Activity Diagram – Admin Add Doctors*

Sequence Diagram

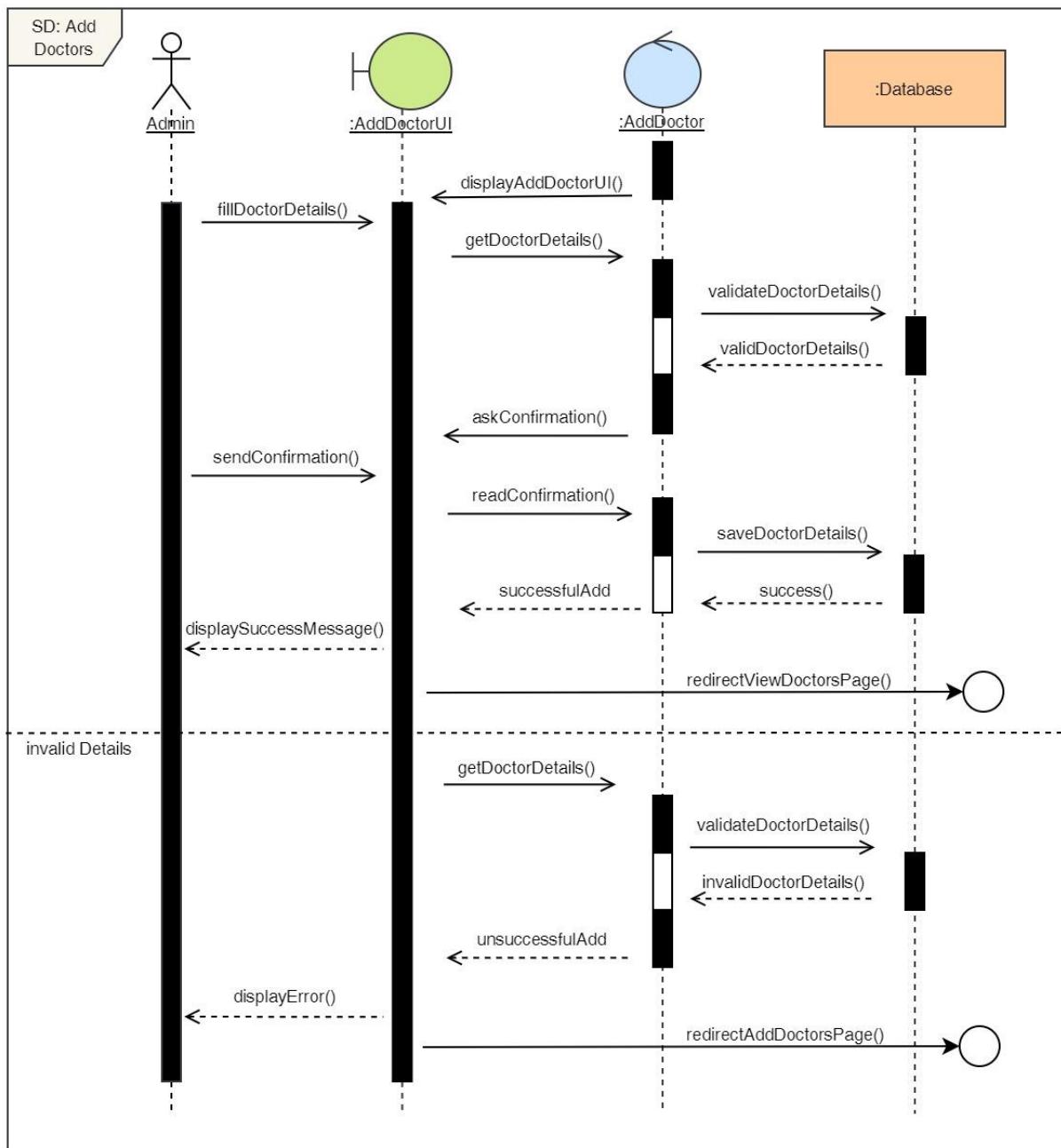


Figure 314: Sequence Diagram – Admin Add Doctors

16. Delete doctors

High Level Use Case Description

Use Case:	Delete Doctors
Actor:	Admin
Description:	Admin can remove existing doctors from the system if the doctor is no longer working in the clinic.

Table 45: High Level Use Case – Admin Delete Doctors

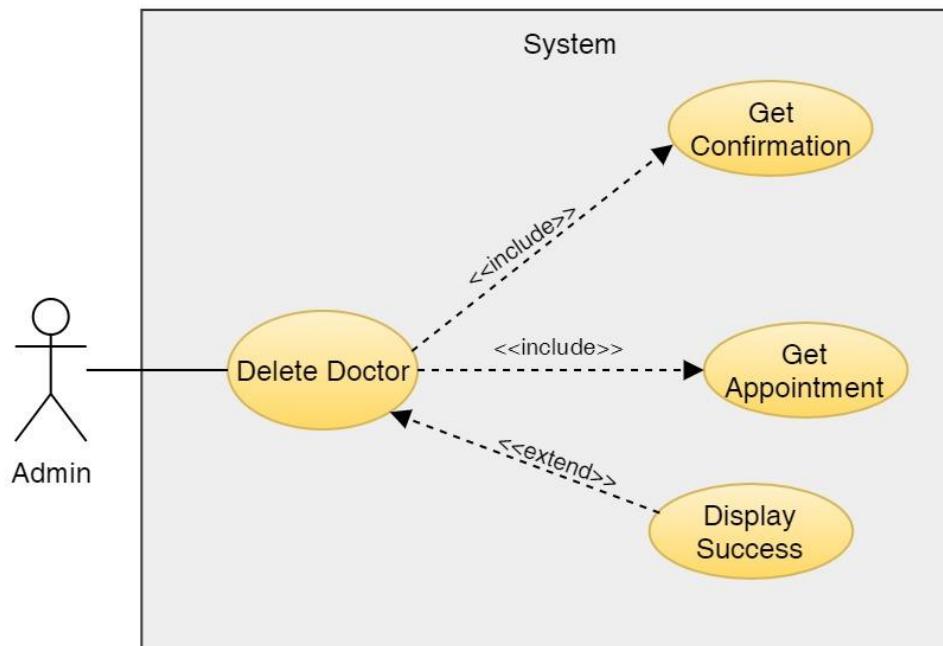
Use Case Diagram

Figure 315: Use Case Diagram – Admin Delete Doctors

Collaboration Diagram

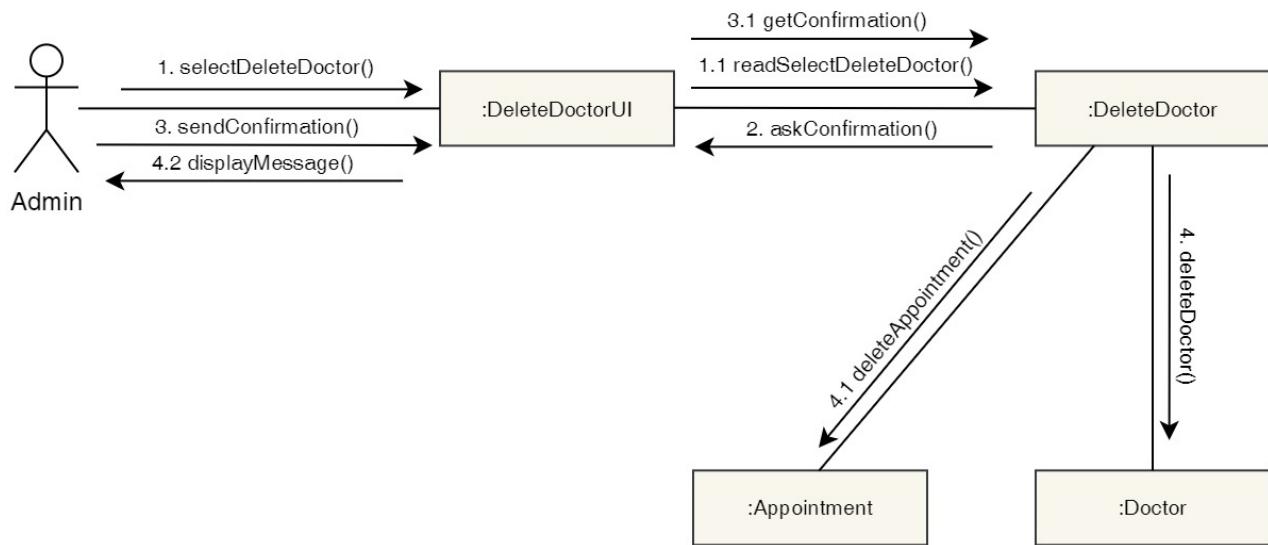


Figure 316: Collaboration Diagram – Admin Delete Doctors

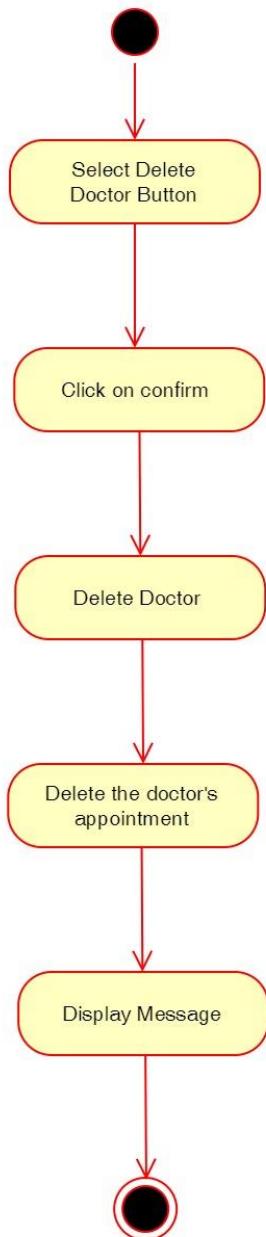
Activity Diagram

Figure 317: Activity Diagram – Admin Delete Doctors

Sequence Diagram

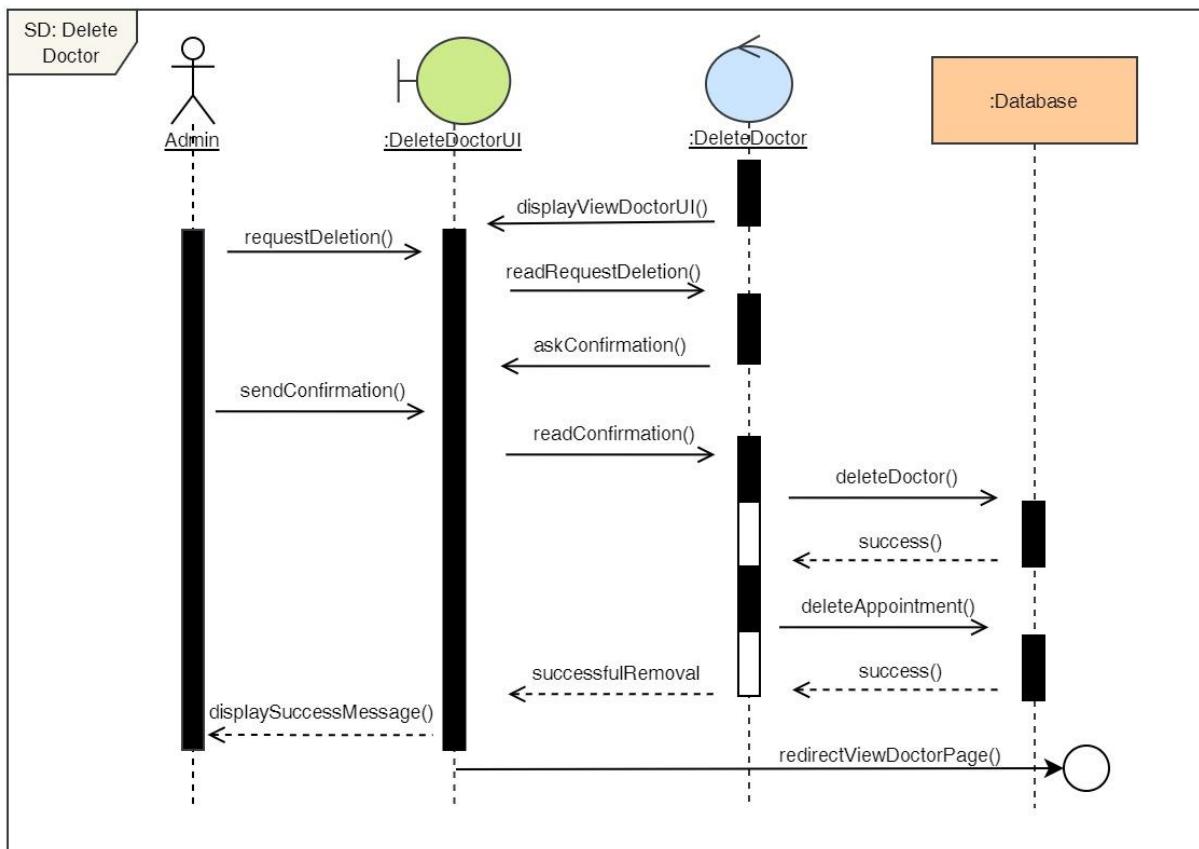


Figure 318: Sequence Diagram – Admin Delete Doctors

17. Edit doctors

High Level Use Case Description

Use Case:	Edit Doctors
Actor:	Admin
Description:	Admin can update the details of the existing doctors in the system.

Table 46: High Level Use Case – Admin Edit Doctors

Use Case Diagram

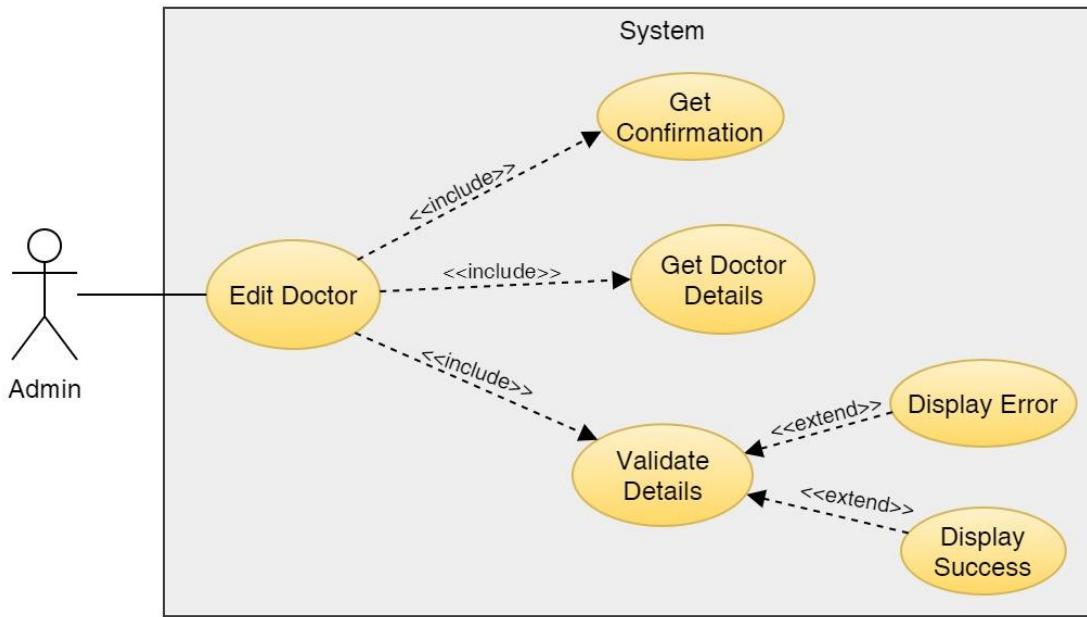


Figure 319: Use Case Diagram – Admin Edit Doctors

Collaboration Diagram

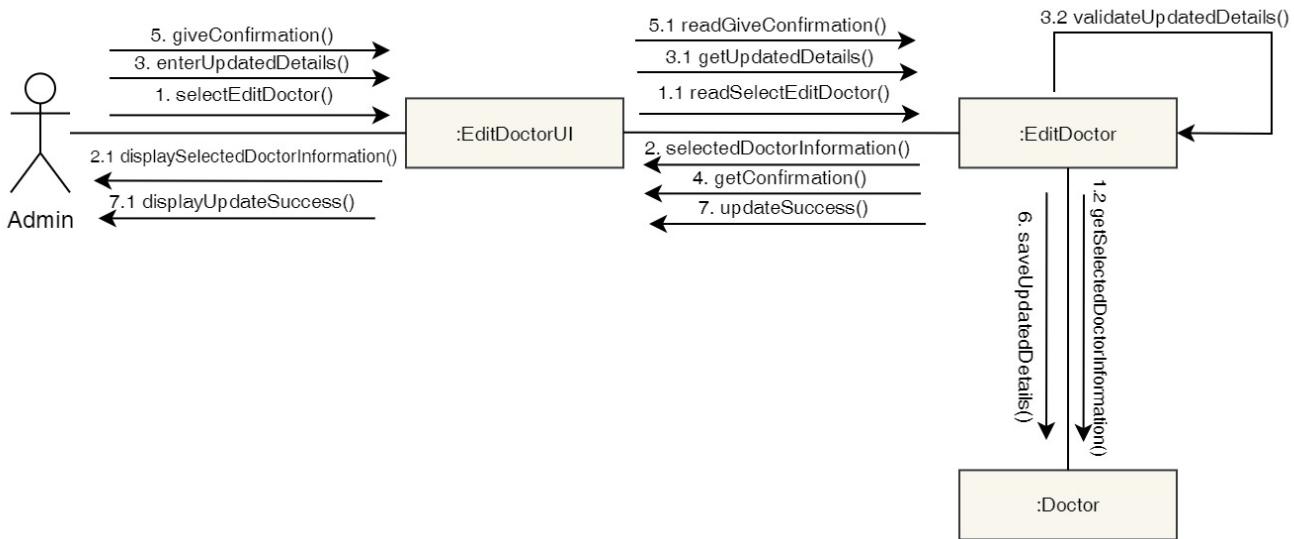
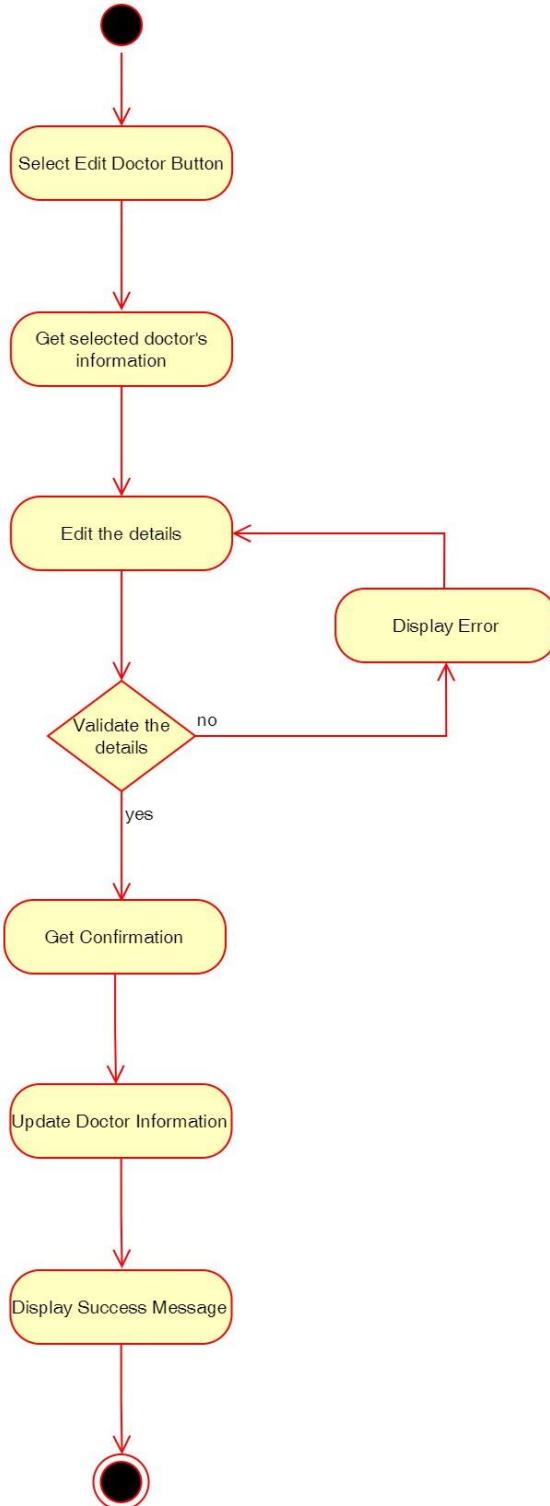


Figure 320: Collaboration Diagram – Admin Edit Doctors

Activity Diagram*Figure 321: Activity Diagram – Admin Edit Doctors*

Sequence Diagram

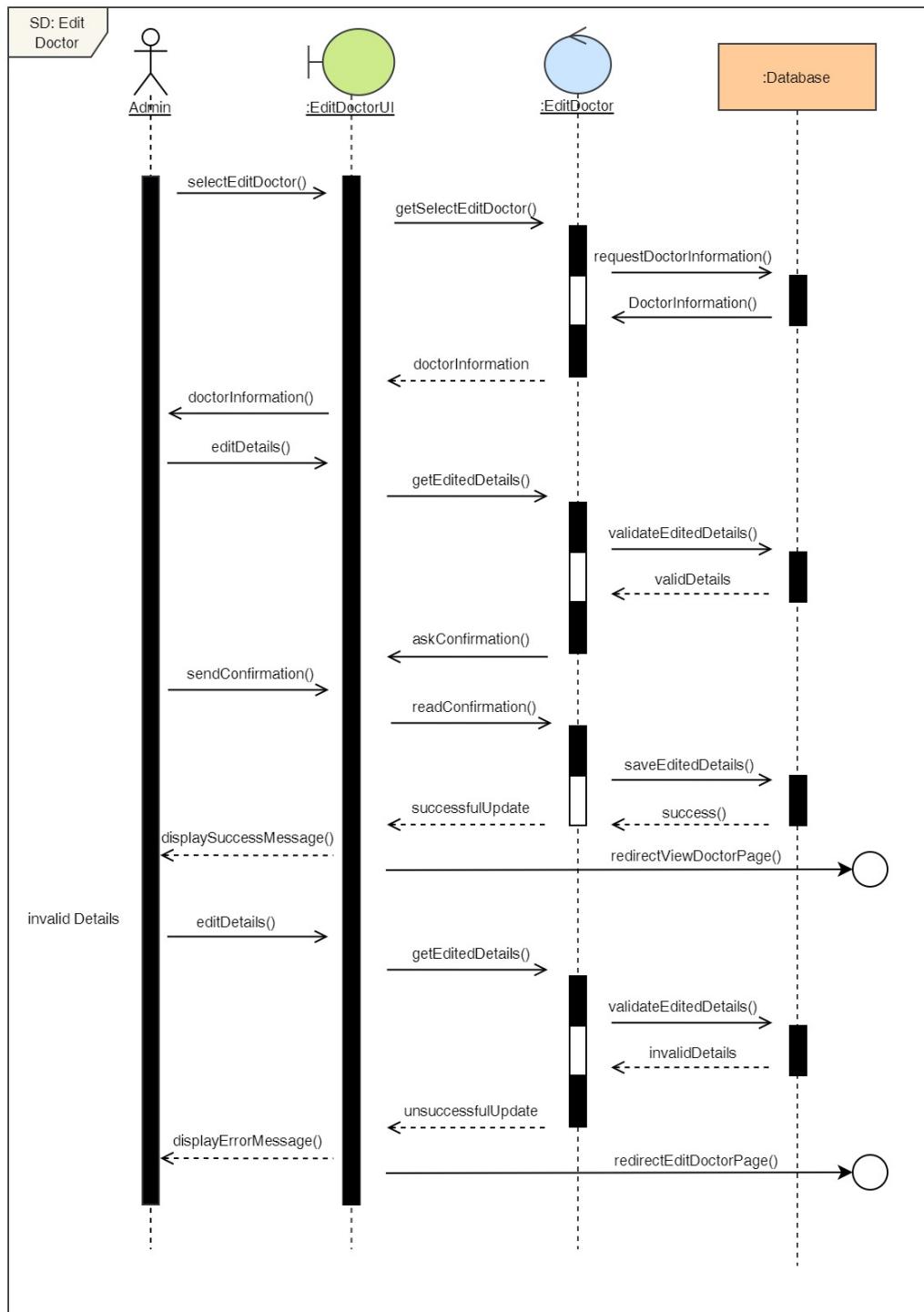


Figure 322: Sequence Diagram – Admin Edit Doctors

18. Add Report

High Level Use Case Description

Use Case:	Add Report
Actor:	Admin
Description:	Admin can add reports of the appointments that has been completed.

Table 47: High Level Use Case – Admin Add Report

Use Case Diagram

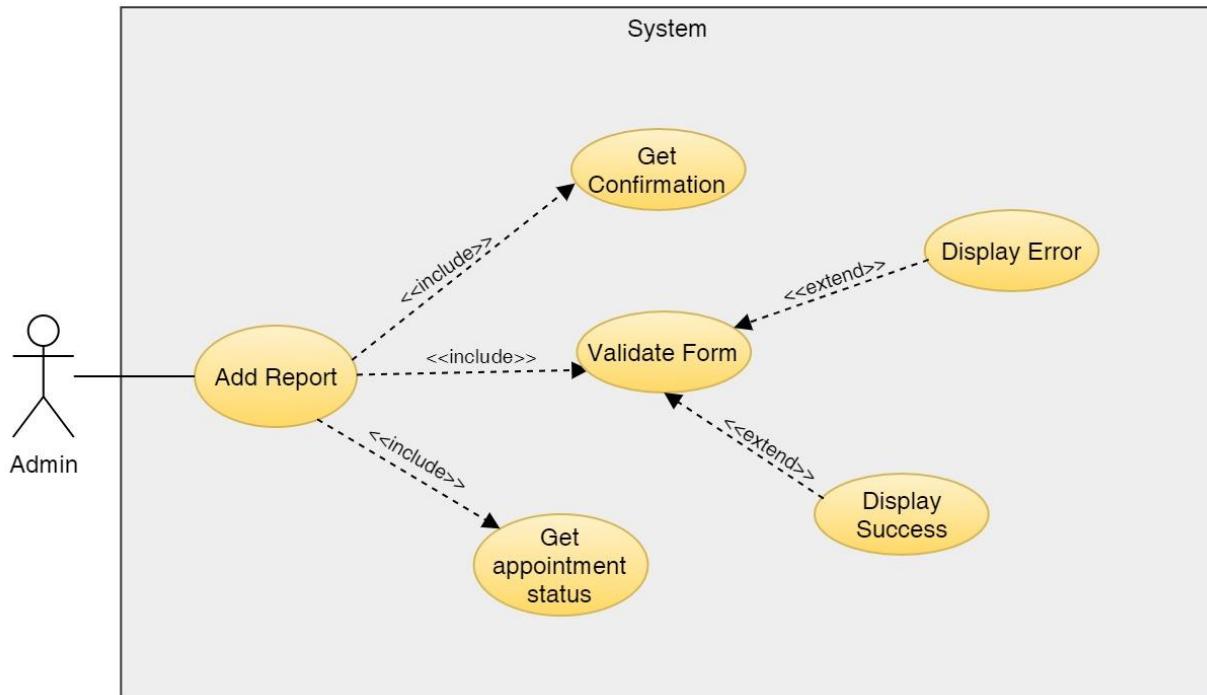


Figure 323: Use Case Diagram – Admin Add Report

Collaboration Diagram

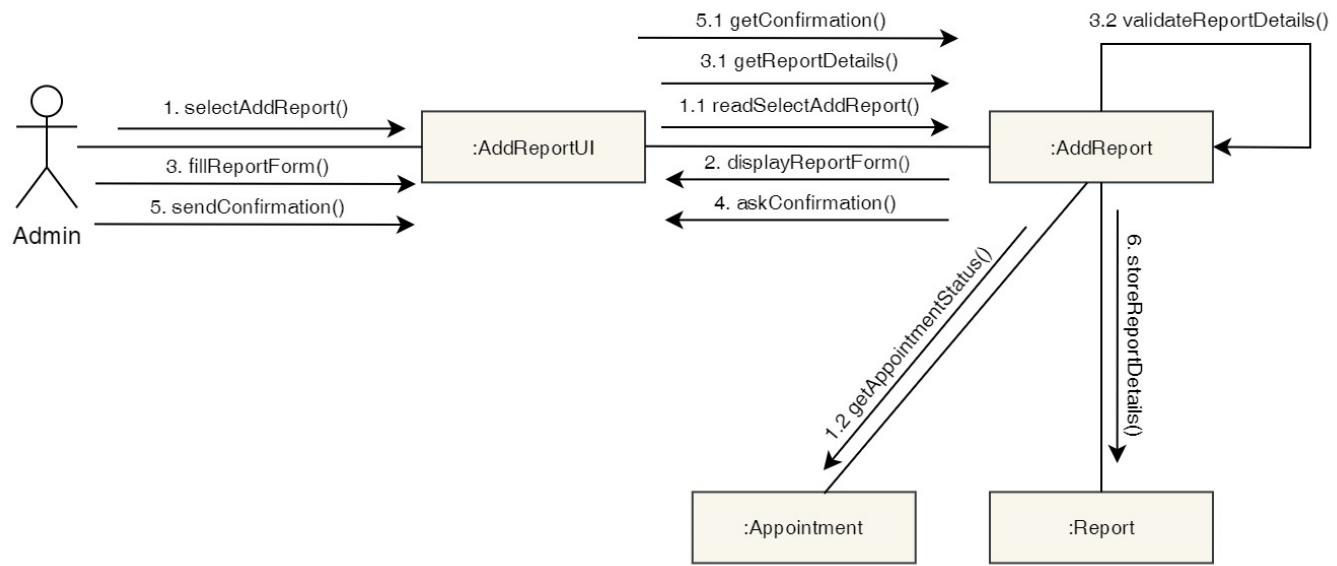
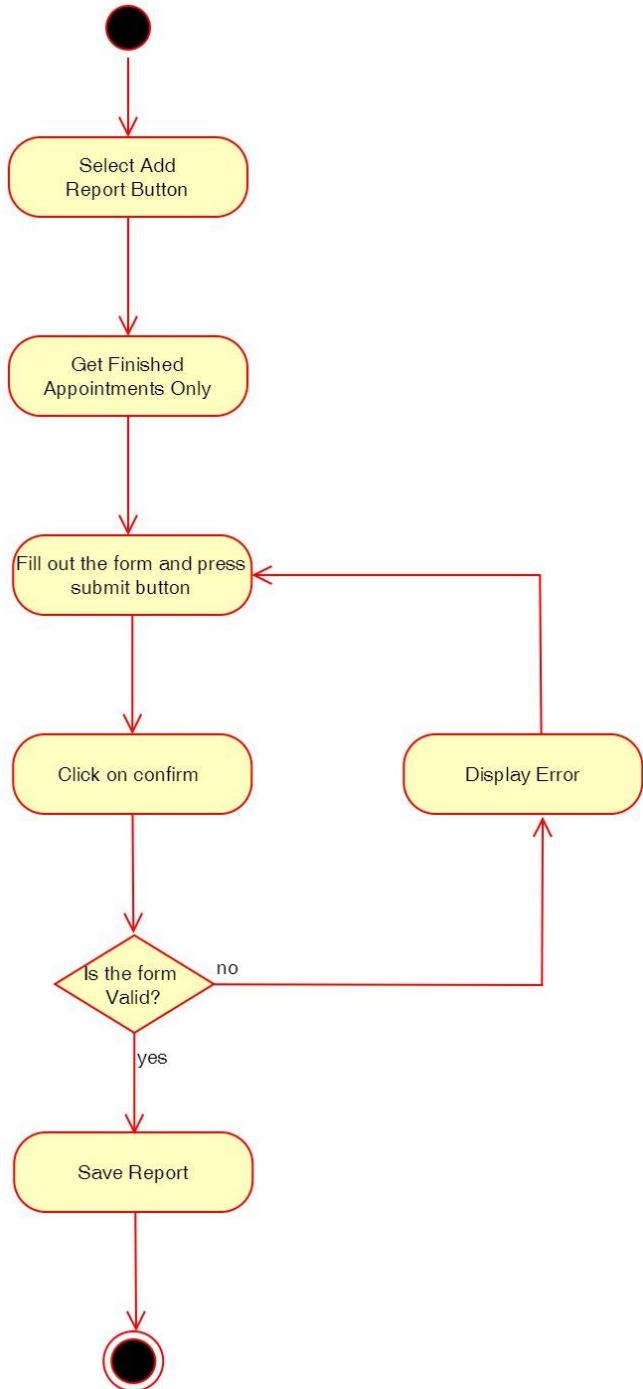


Figure 324: Collaboration Diagram – Admin Add Report

Activity Diagram*Figure 325: Activity Diagram – Admin Add Report*

Sequence Diagram

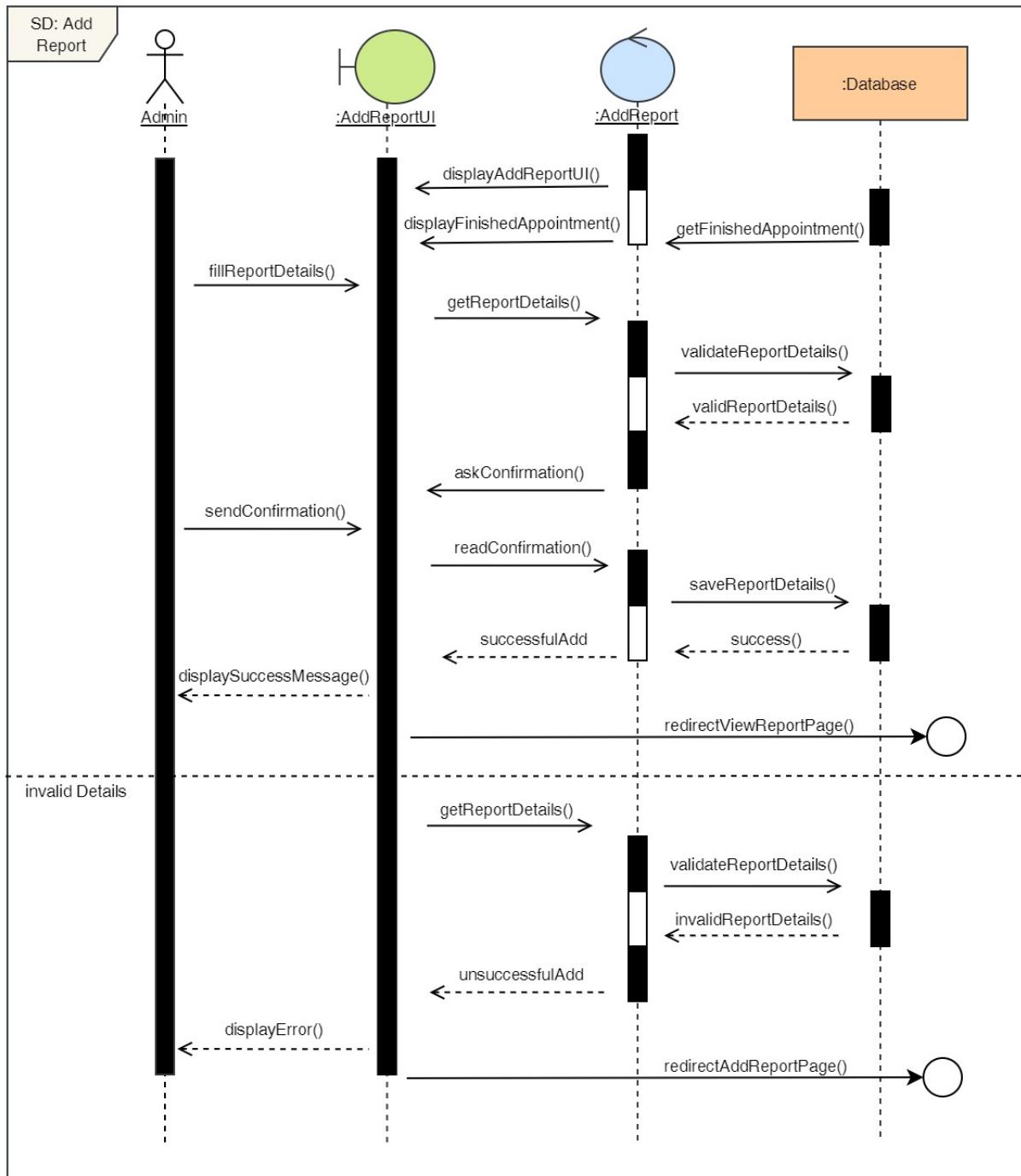


Figure 326: Sequence Diagram – Admin Add Report

19. Delete Report

High Level Use Case Description

Use Case:	Delete Report
Actor:	Admin
Description:	Admin can add remove existing reports from the system.

Table 48: High Level Use Case – Admin Delete Report

Use Case Diagram

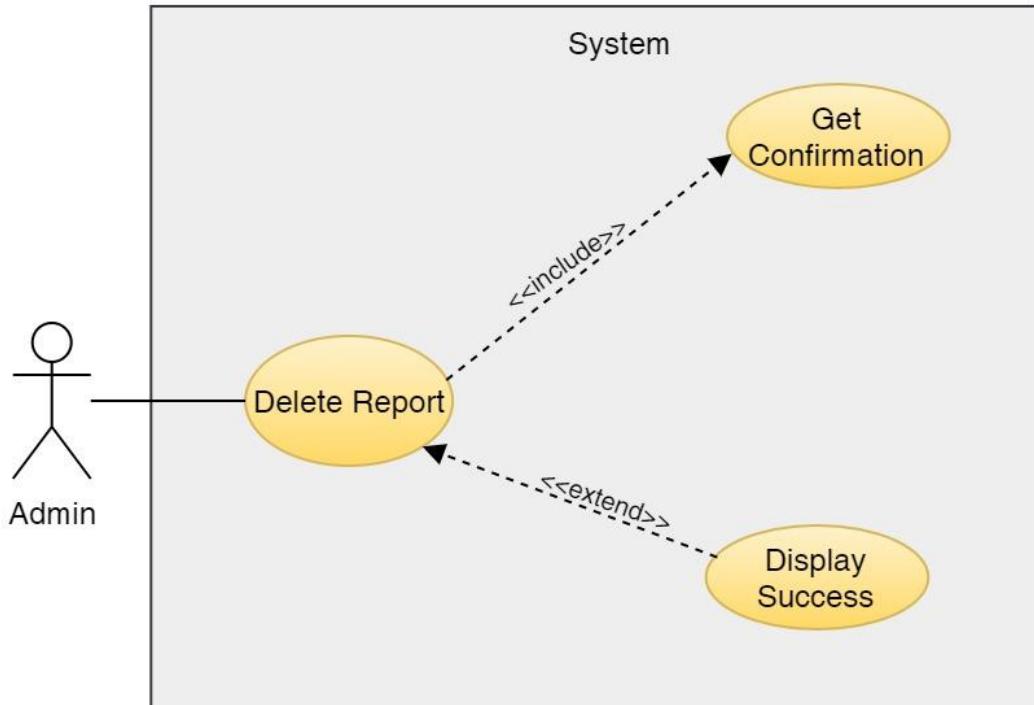


Figure 327: Use Case Diagram – Admin Delete Report

Collaboration Diagram

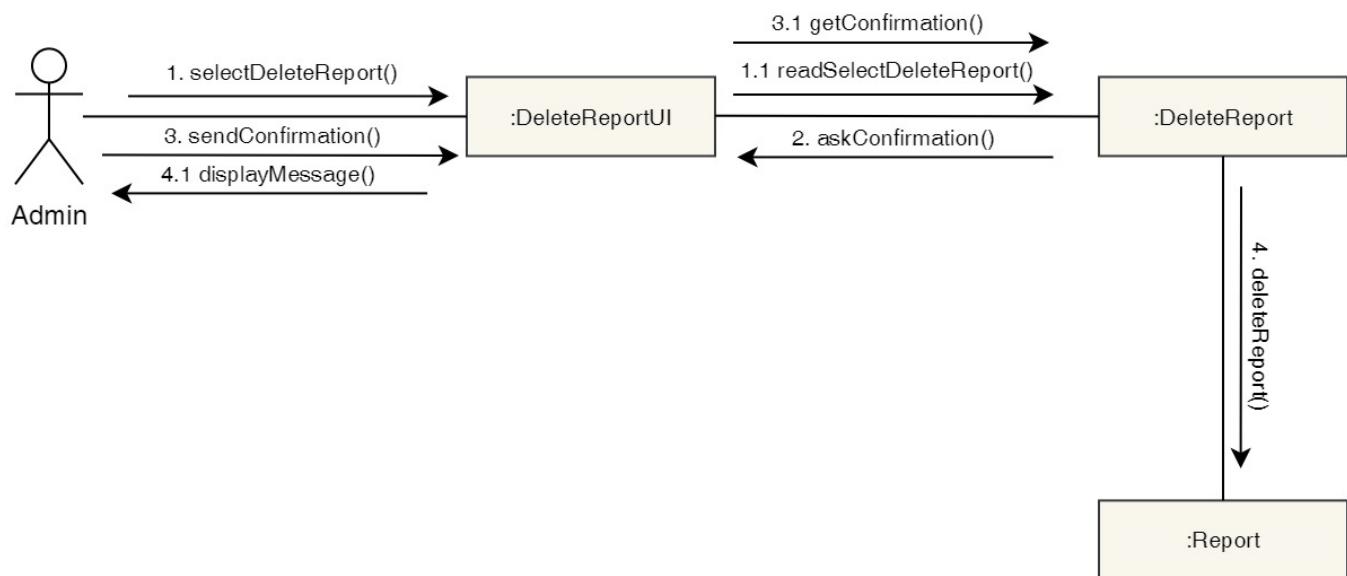


Figure 328: Collaboration Diagram – Admin Delete Report

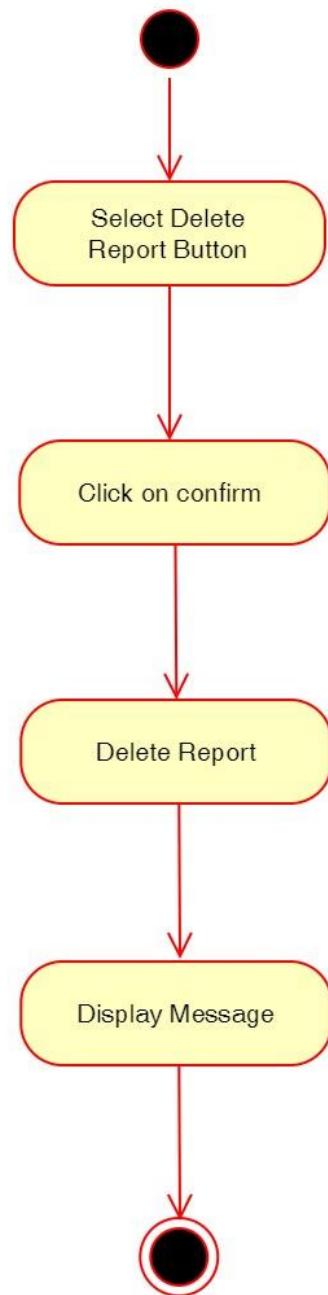
Activity Diagram

Figure 329: Activity Diagram – Admin Delete Report

Sequence Diagram

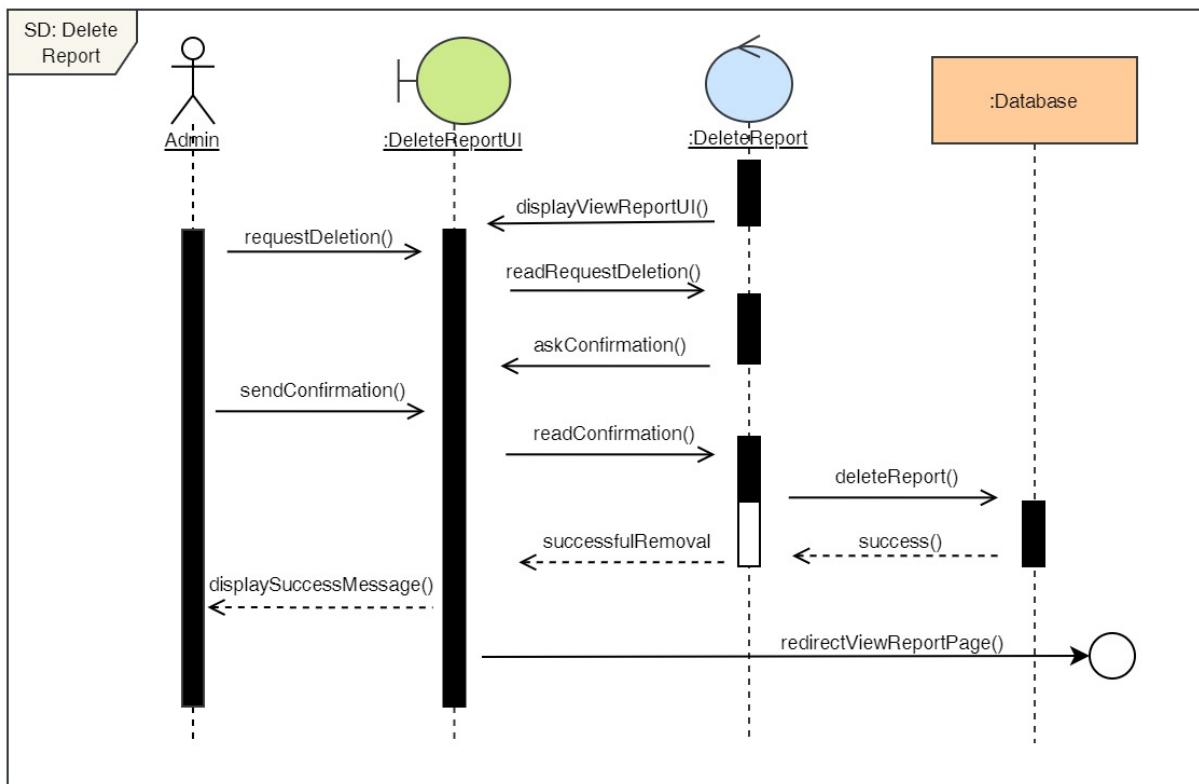


Figure 330: Sequence Diagram – Admin Delete Report

20. Edit reports

High Level Use Case Description

Use Case:	Edit Reports
Actor:	Admin
Description:	Admin can update the details of the existing reports in the system.

Table 49: High Level Use Case – Admin Edit Report

Use Case Diagram

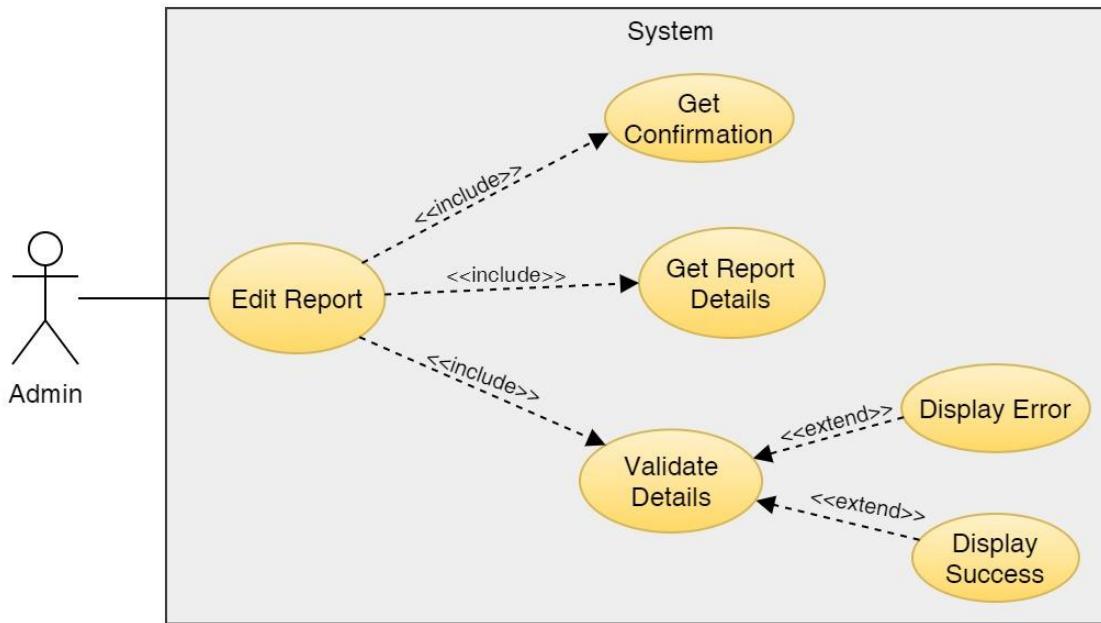


Figure 331: Use Case Diagram – Admin Edit Report

Collaboration Diagram

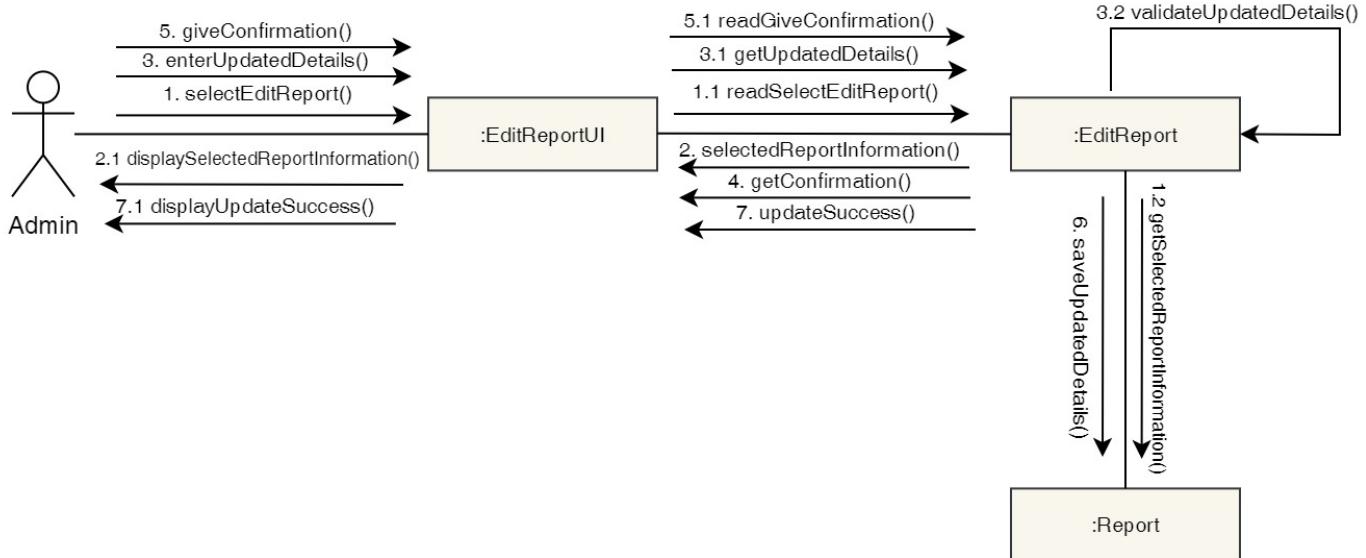
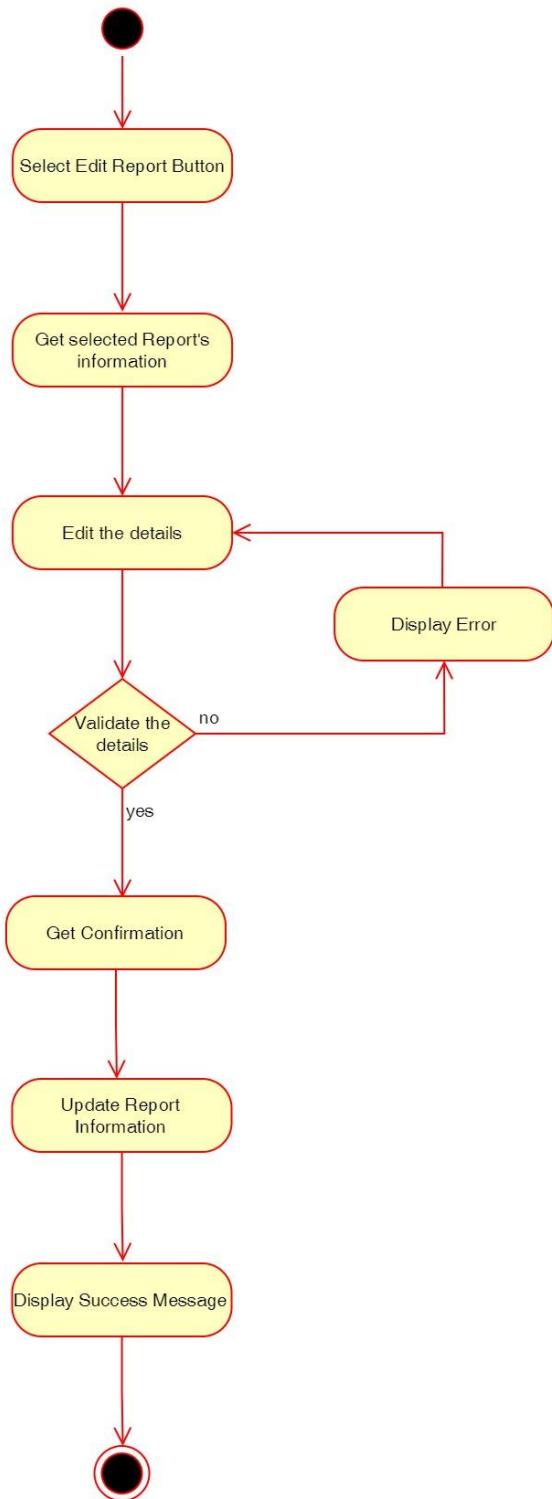


Figure 332: Collaboration Diagram – Admin Edit Report

Activity Diagram*Figure 333: Activity Diagram – Admin Edit Report*

Sequence Diagram

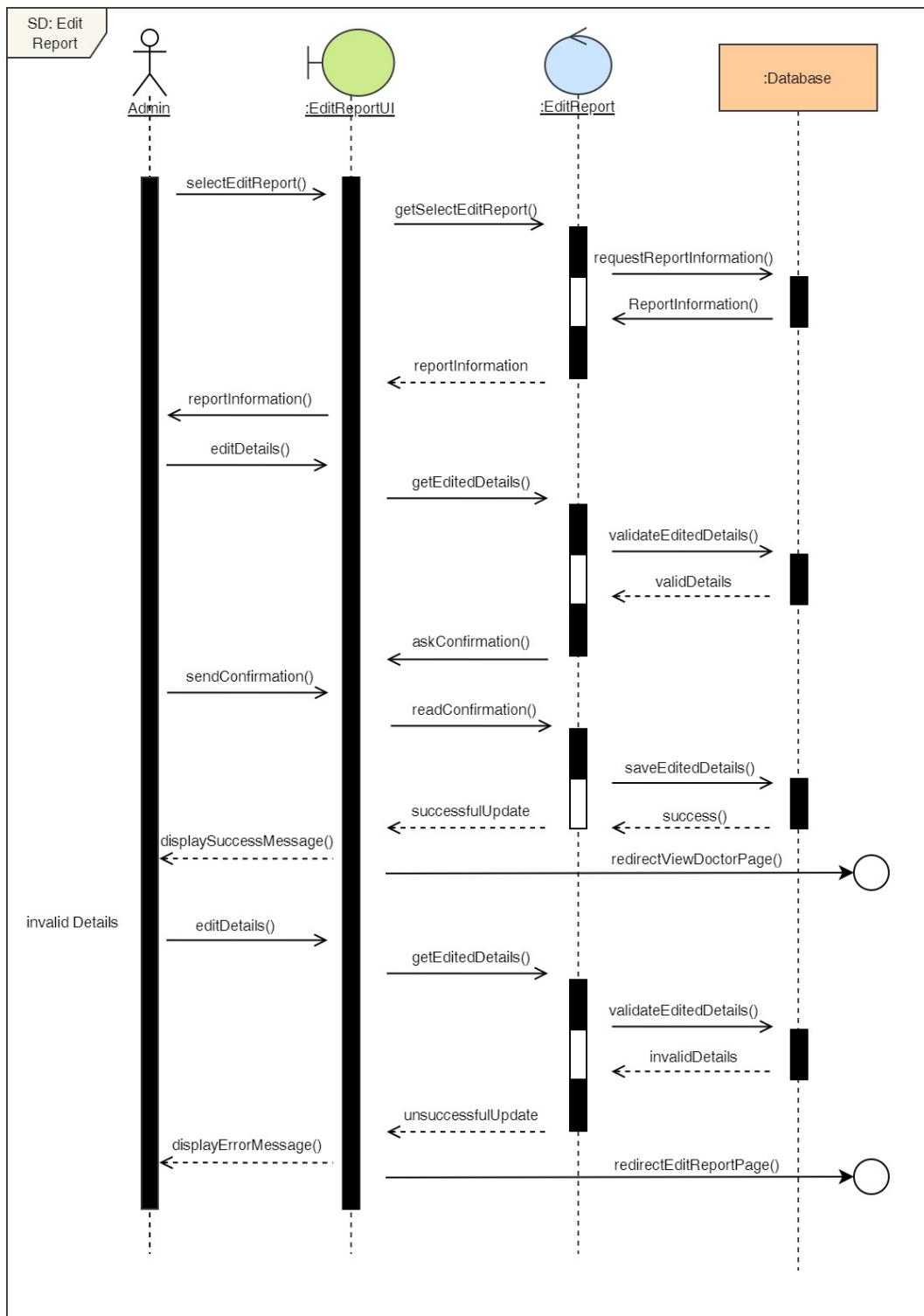


Figure 334: Sequence Diagram – Admin Edit Report

21. View all reports

High Level Use Case Description

Use Case:	View Reports
Actor:	Admin
Description:	Admin can view all the reports of all the users stored in the system.

Table 50: High Level Use Case – Admin View Reports

Use Case Diagram

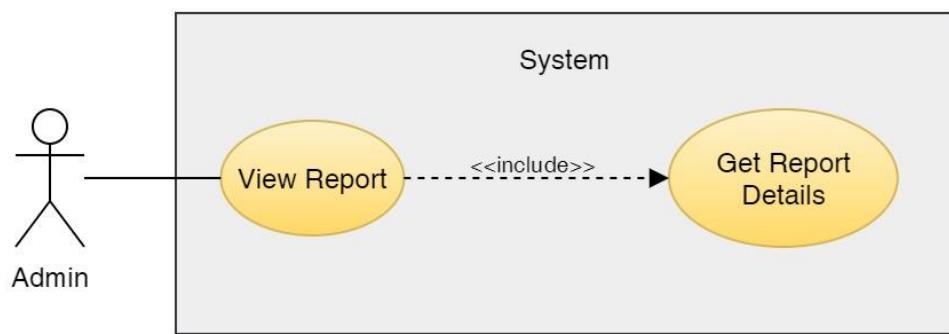


Figure 335: Use Case Diagram – Admin View Reports

Collaboration Diagram

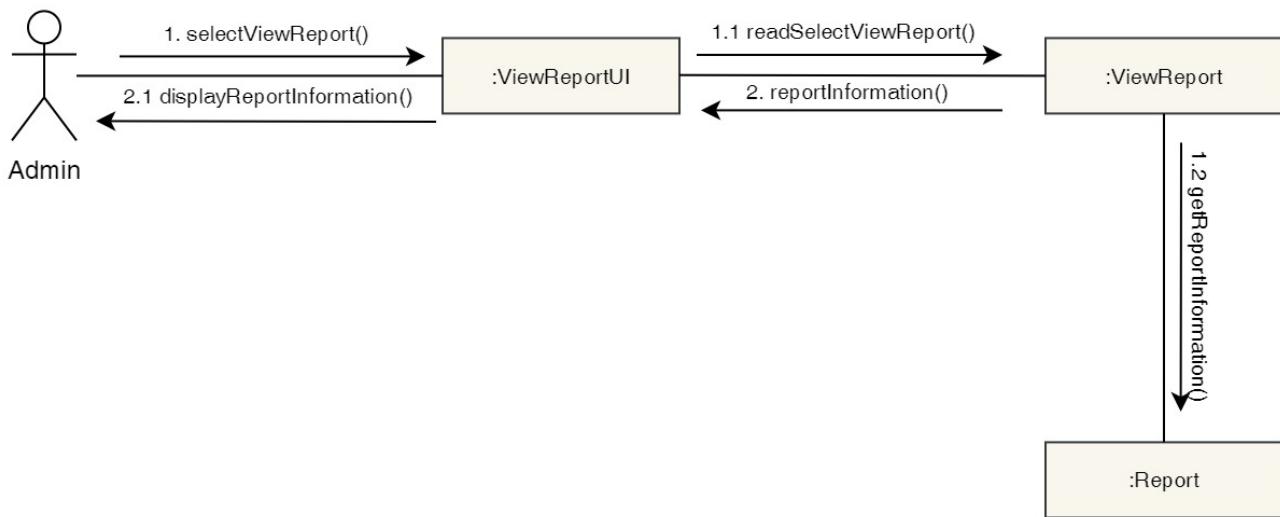


Figure 336: Collaboration Diagram – Admin View Reports

Activity Diagram

Figure 337: Activity Diagram – Admin View Reports

Sequence Diagram

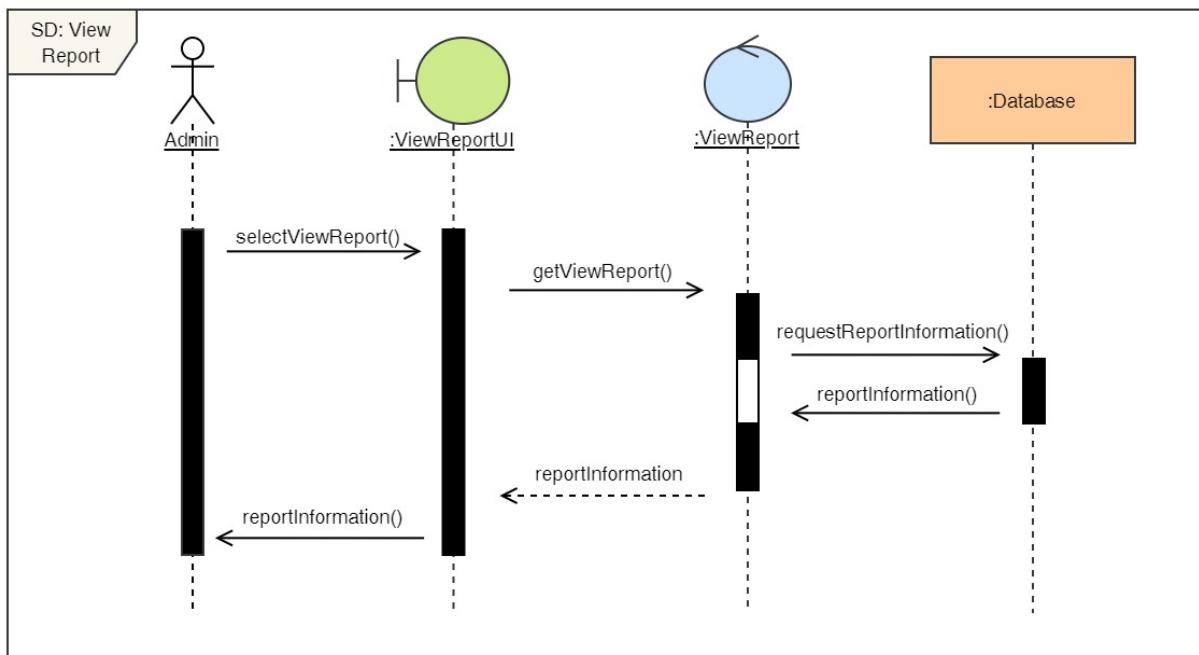


Figure 338: Sequence Diagram – Admin View Reports

22. Book a Dental Appointment

High Level Use Case Description

Use Case:	Book Appointment
Actor:	Admin
Description:	Admin can book appointment for any user and any dental services by selecting the desired date, time, doctor and service.

Table 51: High Level Use Case – Admin Book Appointment

Use Case Diagram

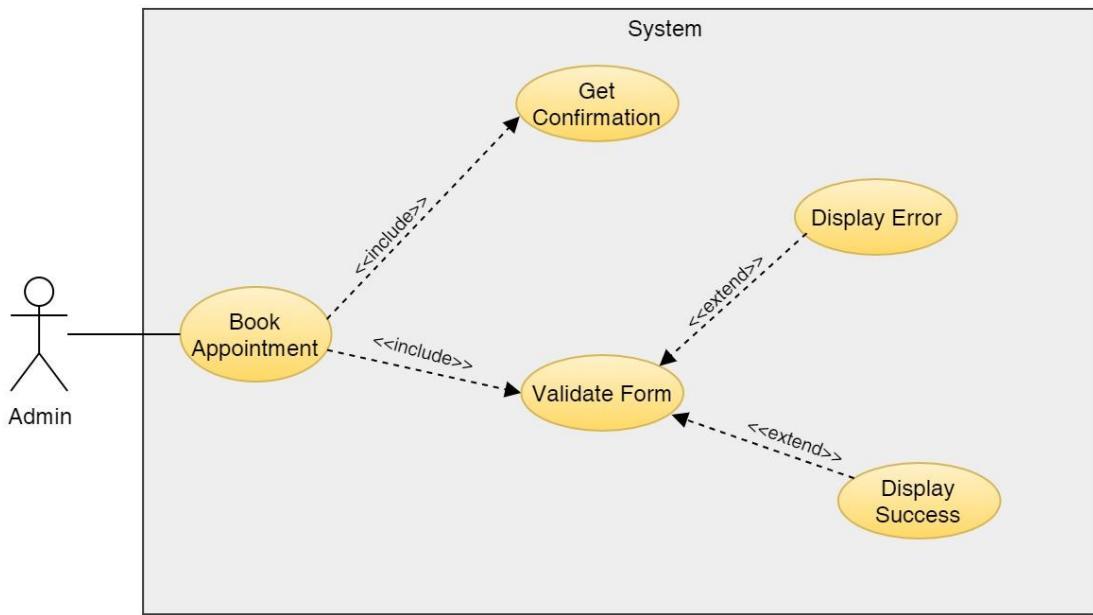


Figure 339: Use Case Diagram – Admin Book Appointment

Collaboration Diagram

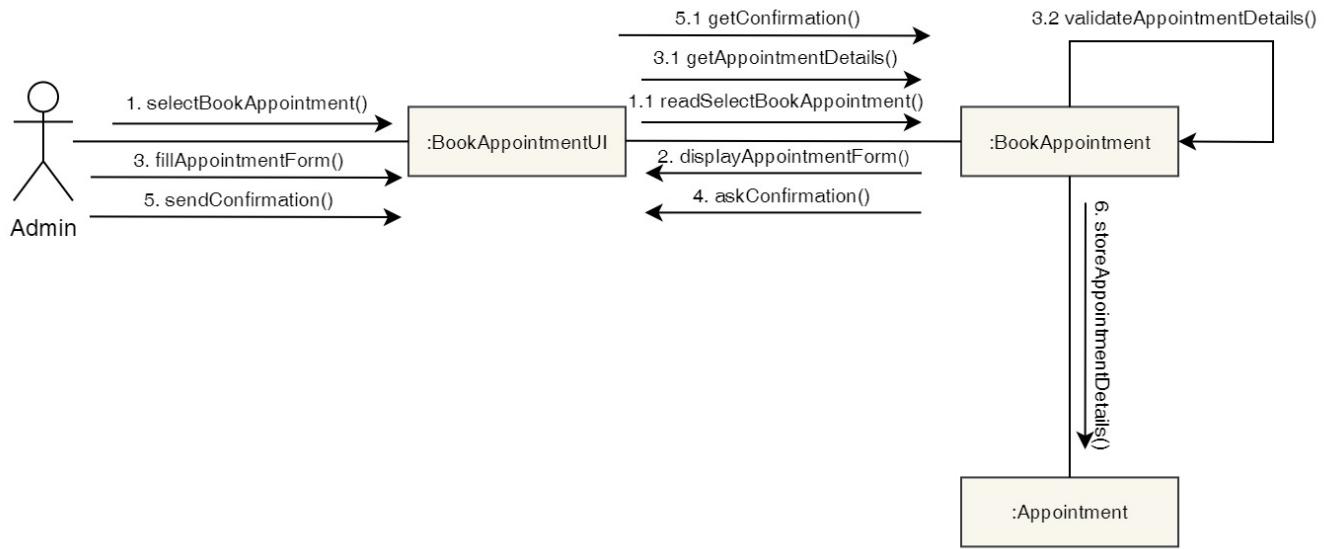
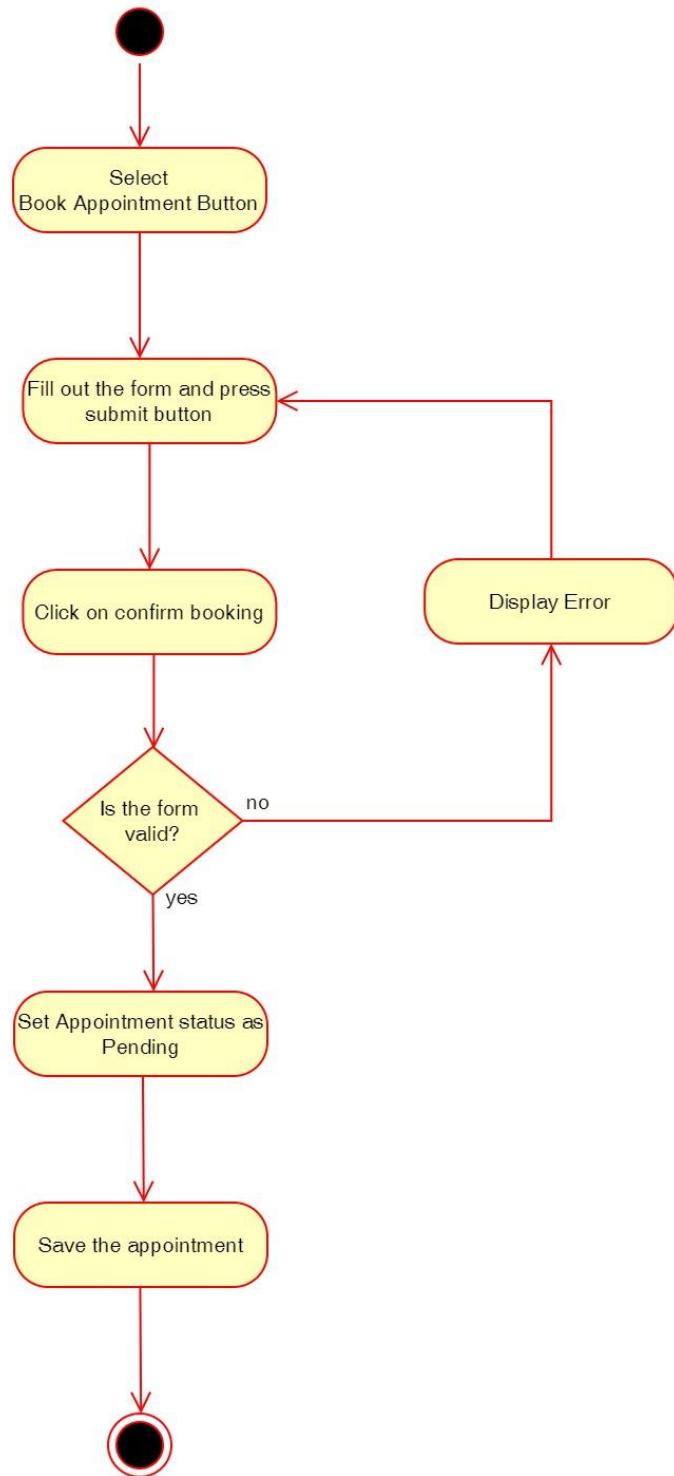


Figure 340: Collaboration Diagram – Admin Book Appointment

Activity Diagram*Figure 341: Activity Diagram – Admin Book Appointment*

Sequence Diagram

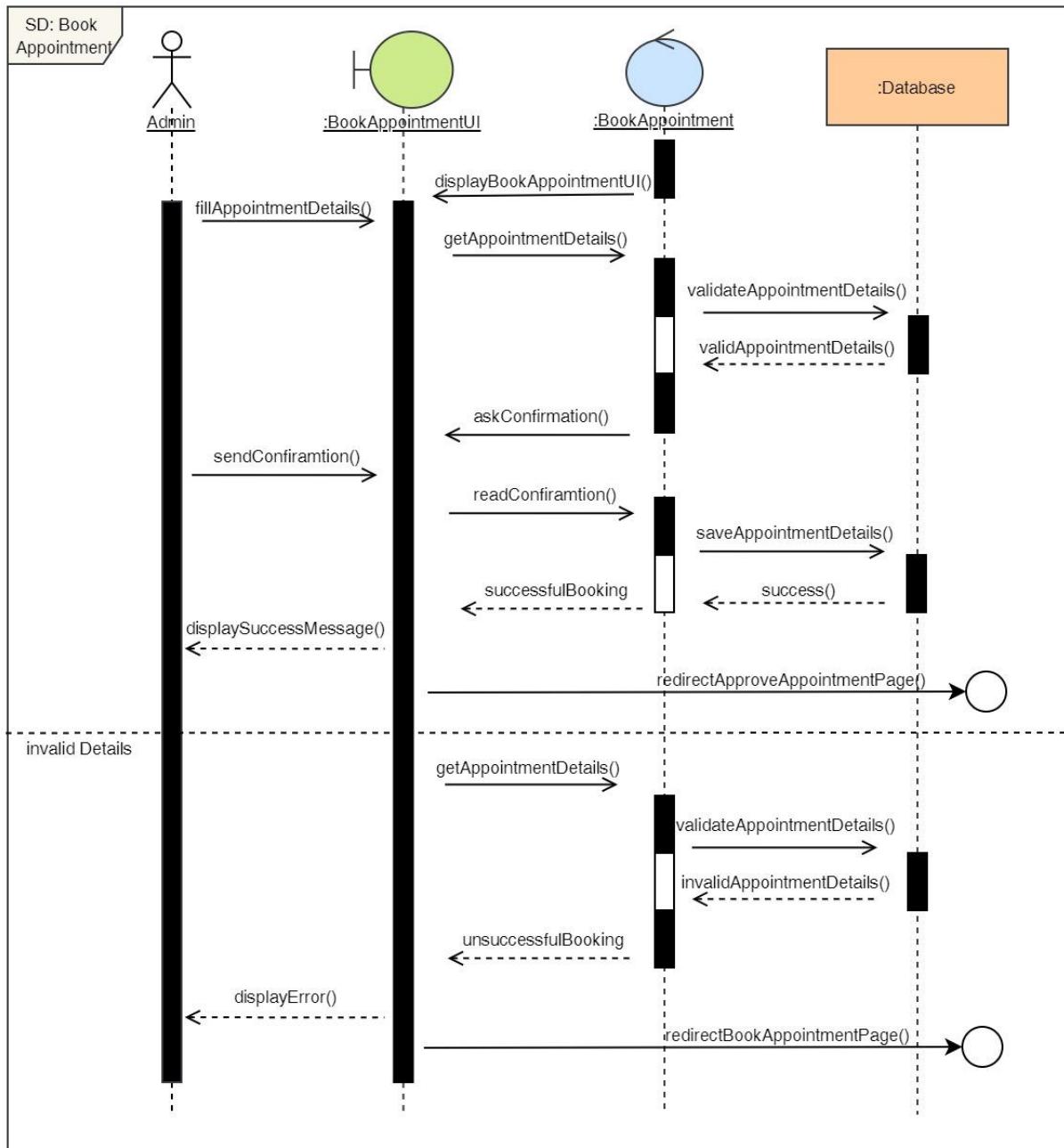


Figure 342: Sequence Diagram – Admin Book Appointment

23. View all appointments

High Level Use Case Description

Use Case:	View all appointments
Actor:	Admin
Description:	Admin can view all the appointments the users have made from the system that is not pending.

Table 52: High Level Use Case – Admin View Appointments

Use Case Diagram

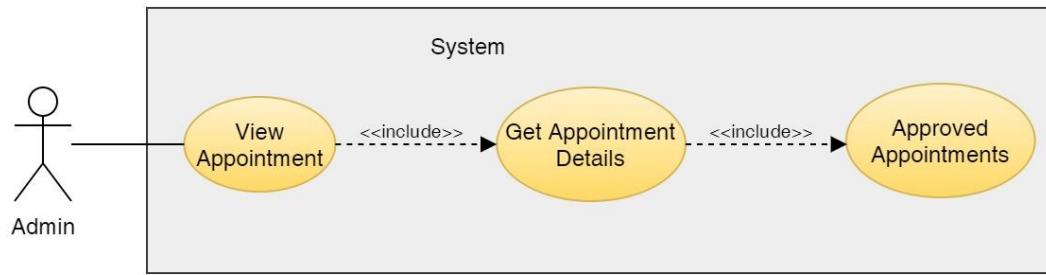


Figure 343: Use Case Diagram – Admin View Appointments

Collaboration Diagram

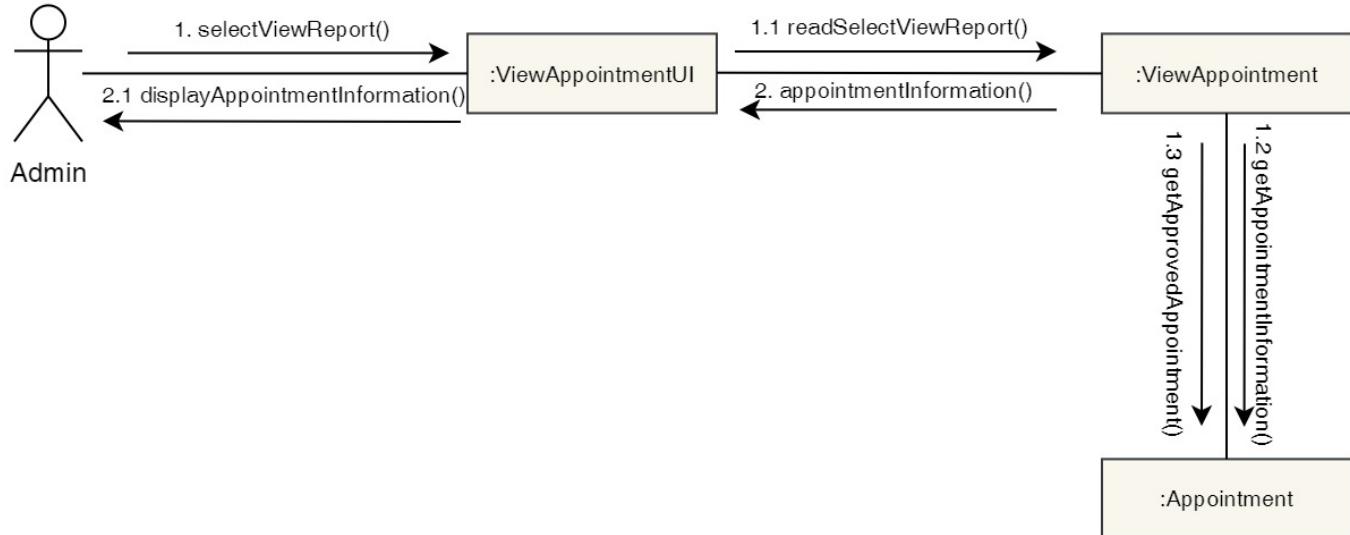


Figure 344: Collaboration Diagram – Admin View Appointments

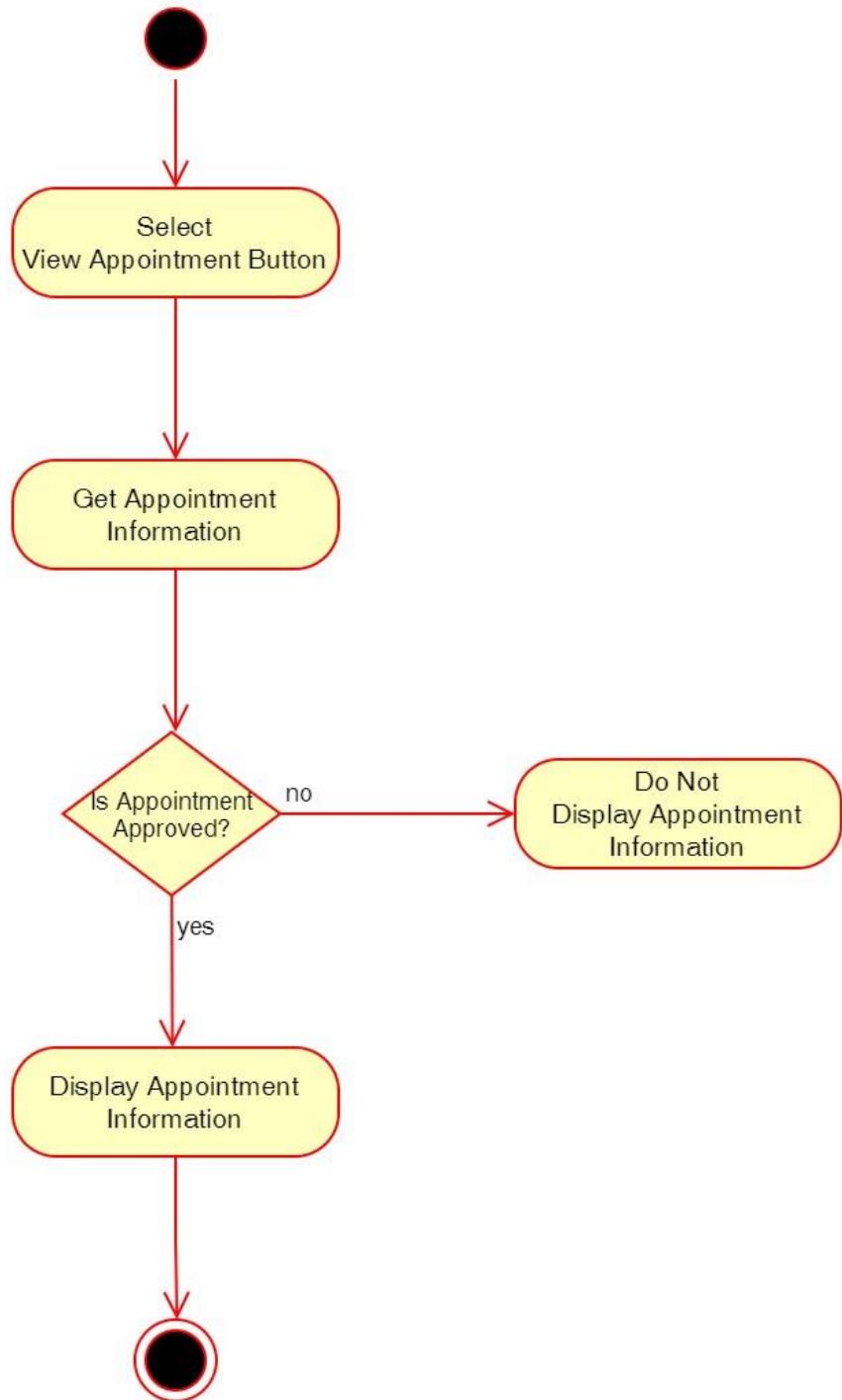
Activity Diagram

Figure 345: Activity Diagram – Admin View Appointments

Sequence Diagram

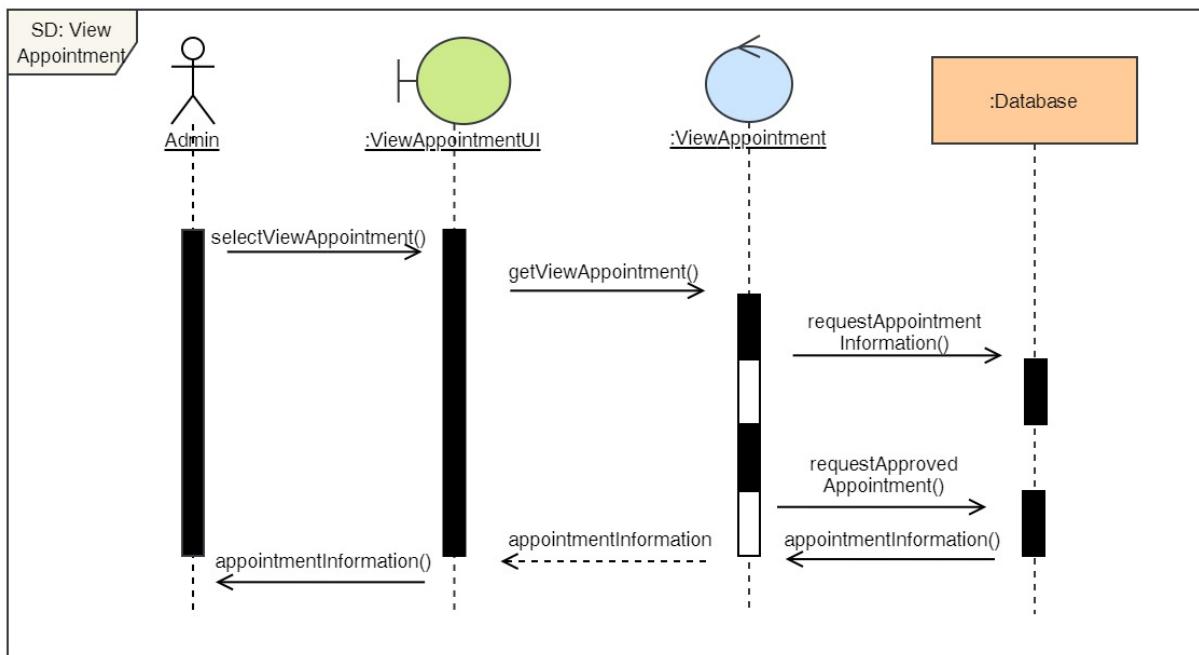


Figure 346: Sequence Diagram – Admin View Appointments

24. Cancel Appointment

High Level Use Case Description

Use Case:	Cancel Appointment
Actor:	Admin
Description:	The admin can cancel appointment of the user's that has already been confirmed.

Table 53: High Level Use Case – Admin Cancel Appointment

Use Case Diagram

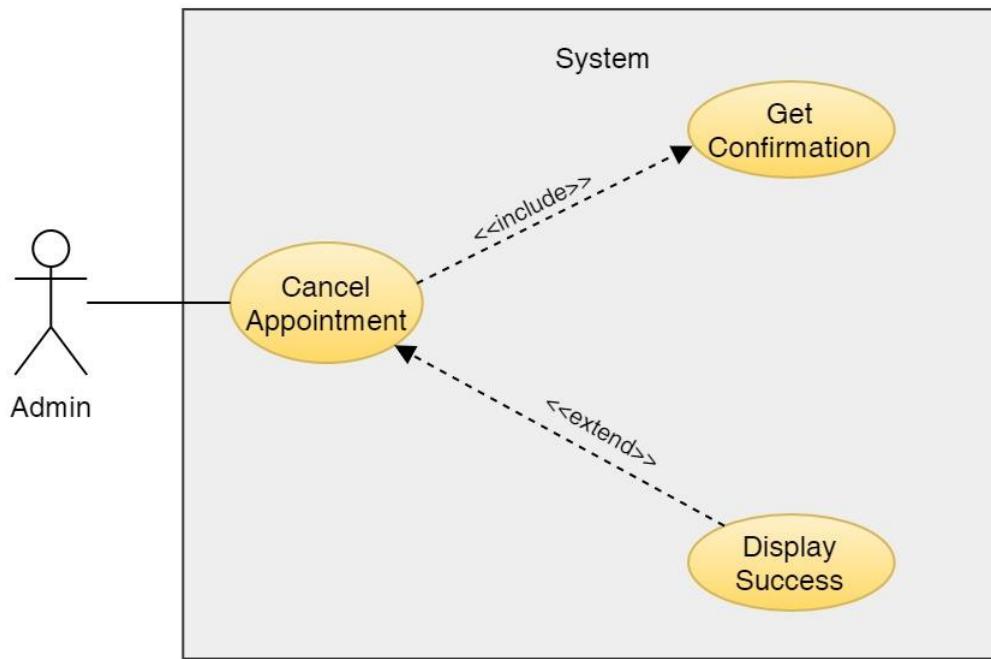


Figure 347: Use Case Diagram – Admin Cancel Appointment

Collaboration Diagram

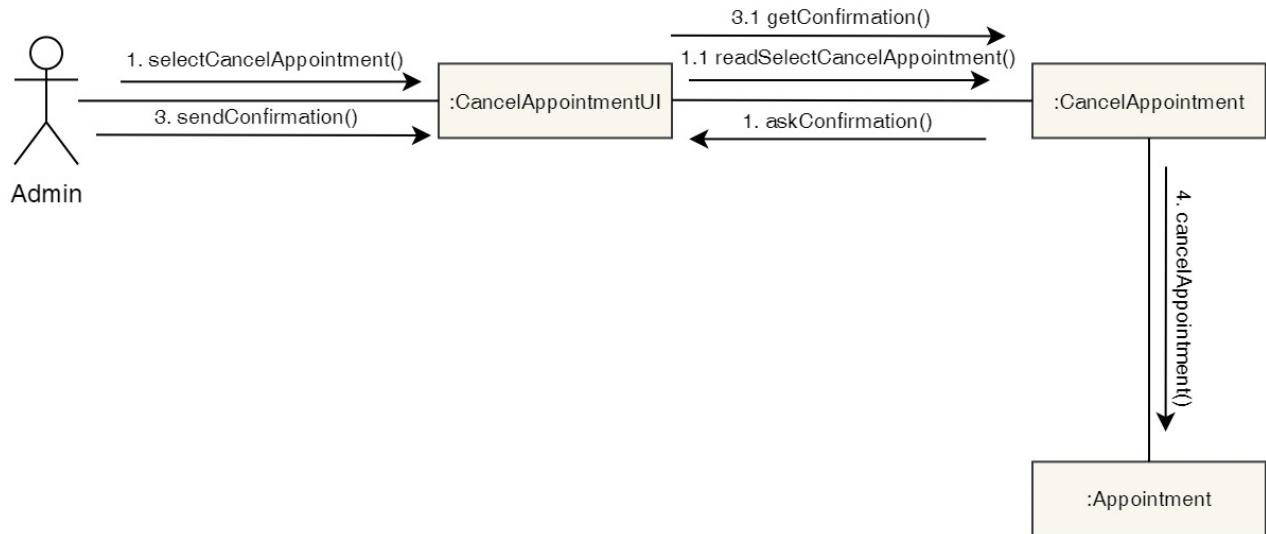


Figure 348: Collaboration Diagram – Admin Cancel Appointment

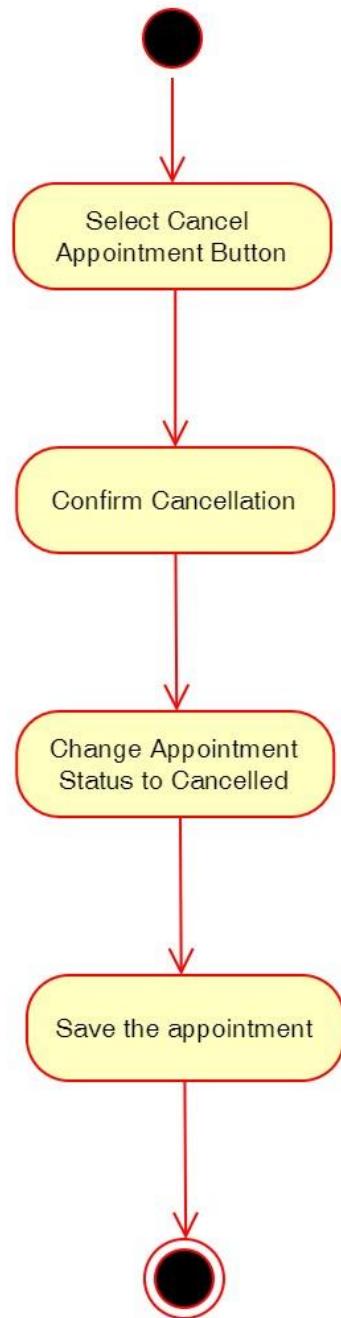
Activity Diagram

Figure 349: Activity Diagram – Admin Cancel Appointment

Sequence Diagram

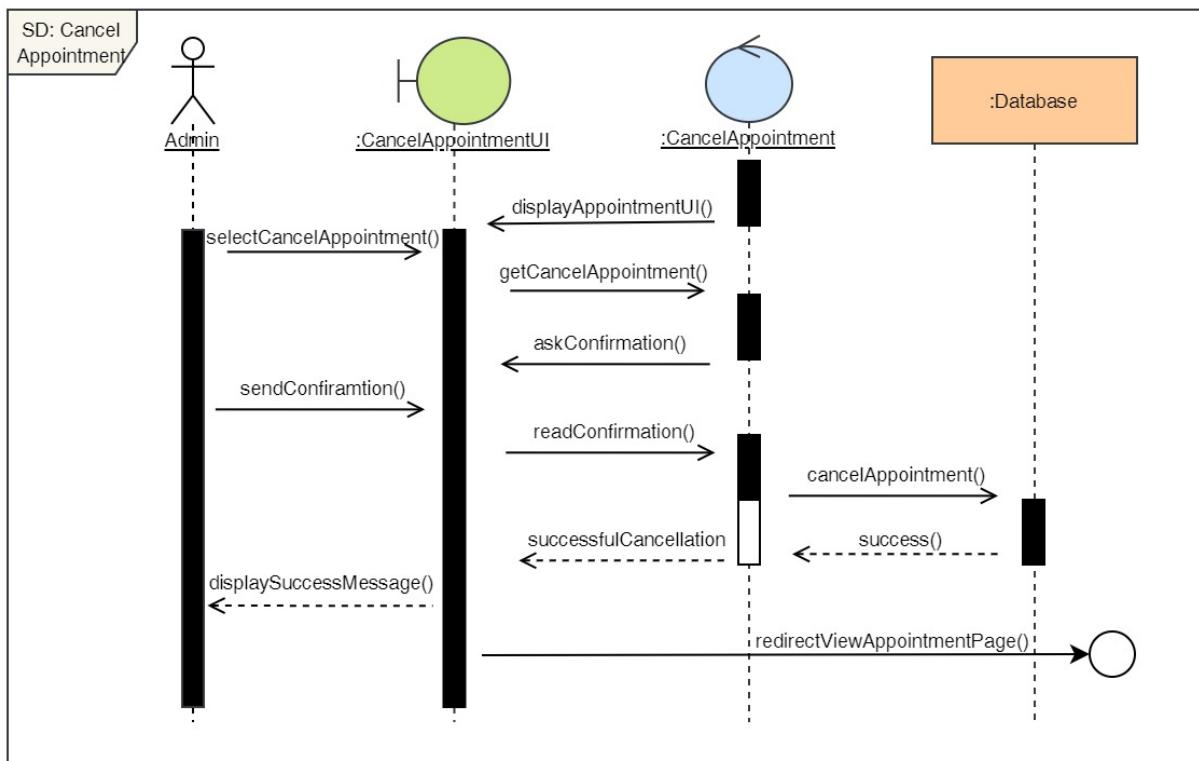


Figure 350: Sequence Diagram – Admin Cancel Appointment

25. Approve Appointment

High Level Use Case Description

Use Case:	Approve Appointment
Actor:	Admin
Description:	When a user books an appointment, it gets in pending stage. The admin then approves the pending appointment and changes the status of the appointment from pending to confirmed.

Table 54: High Level Use Case – Admin Approve Appointment

Use Case Diagram

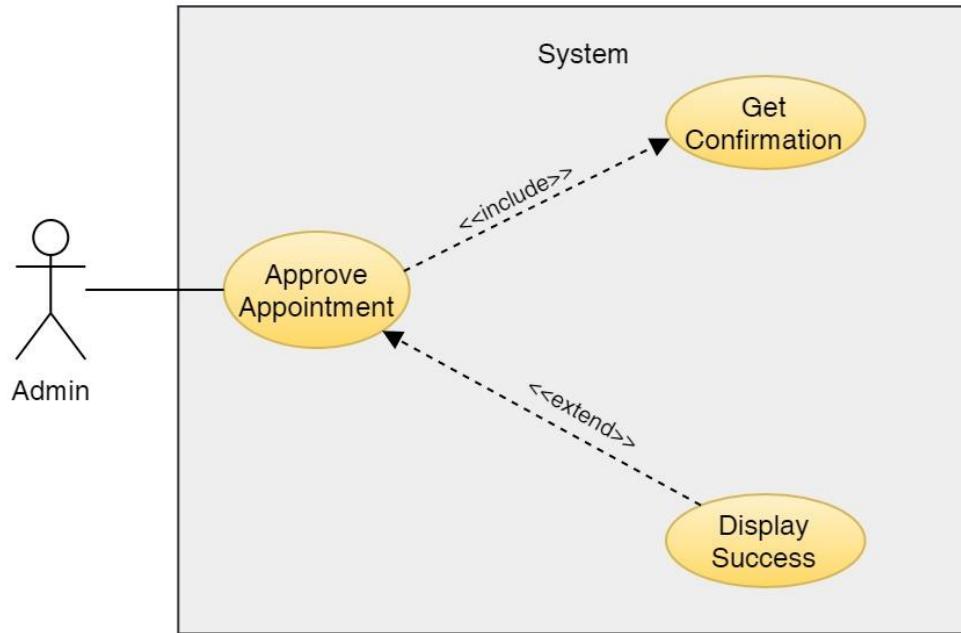


Figure 351: Use Case Diagram – Admin Approve Appointment

Collaboration Diagram

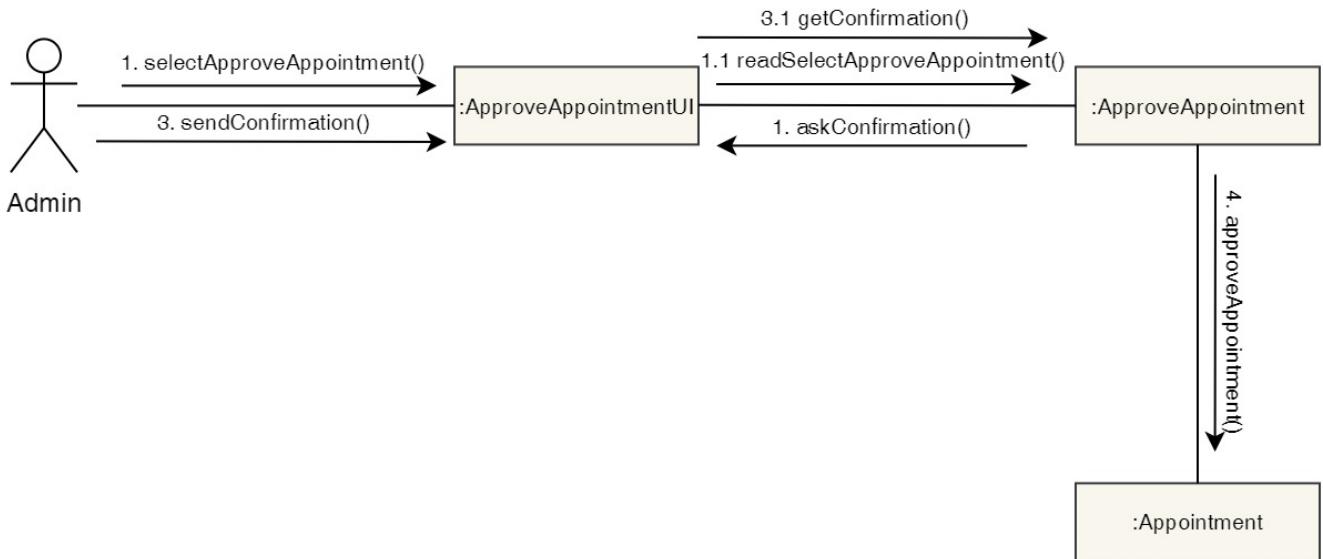


Figure 352: Collaboration Diagram – Admin Approve Appointment

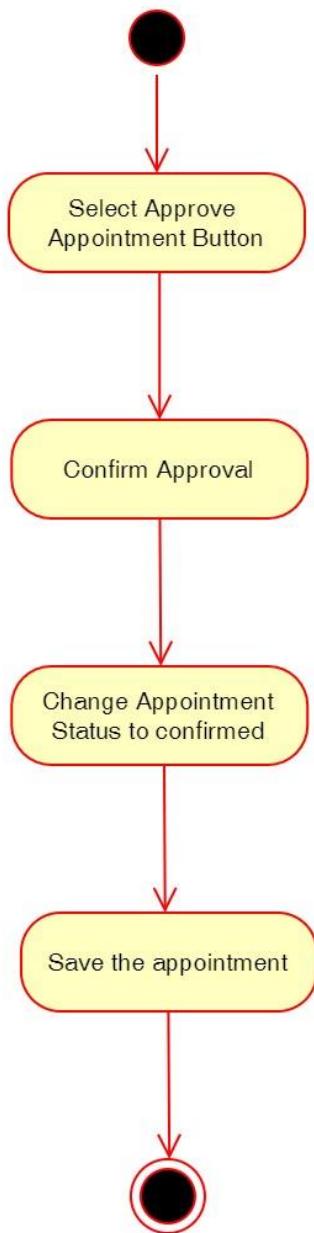
Activity Diagram

Figure 353: Activity Diagram – Admin Approve Appointment

Sequence Diagram

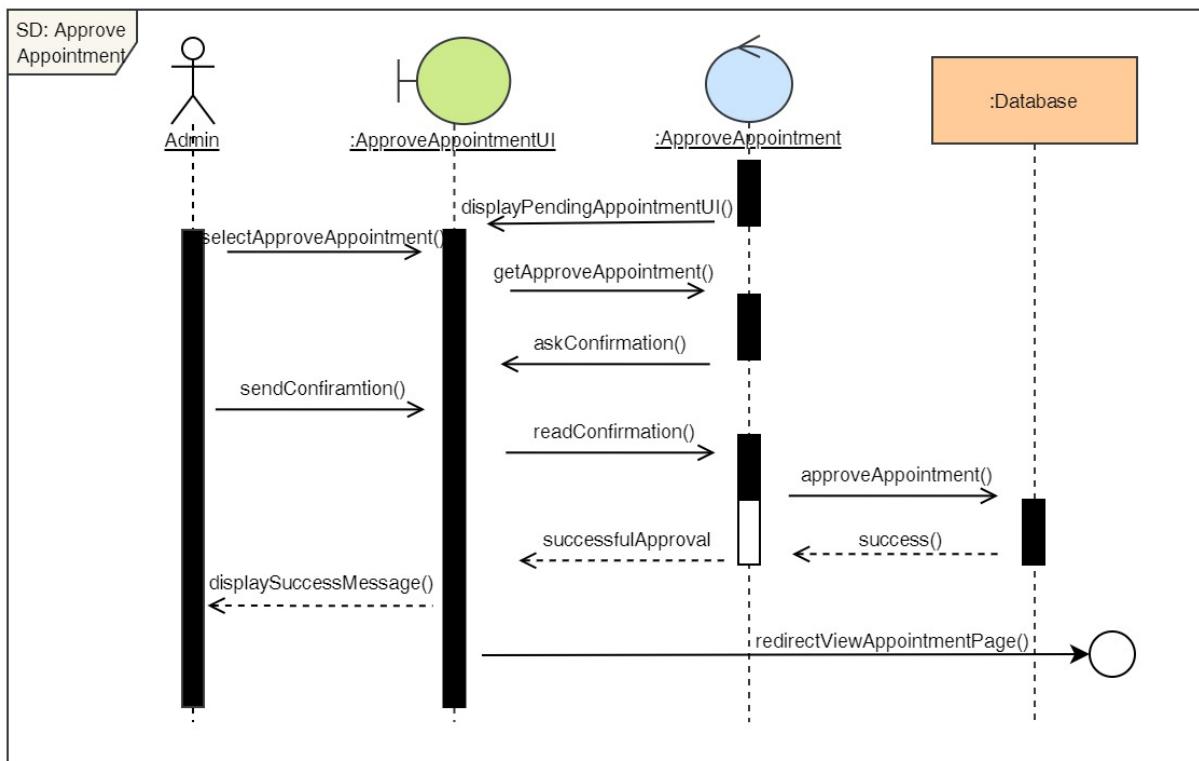


Figure 354: Sequence Diagram – Admin Approve Appointment

26. Reject Appointment

High Level Use Case Description

Use Case:	Reject Appointment
Actor:	Admin
Description:	When a user books an appointment, it gets in pending stage. The admin then rejects the pending appointment if the appointment does not have enough information and changes the status of the appointment from pending to rejected.

Table 55: High Level Use Case – Admin Reject Appointment

Use Case Diagram

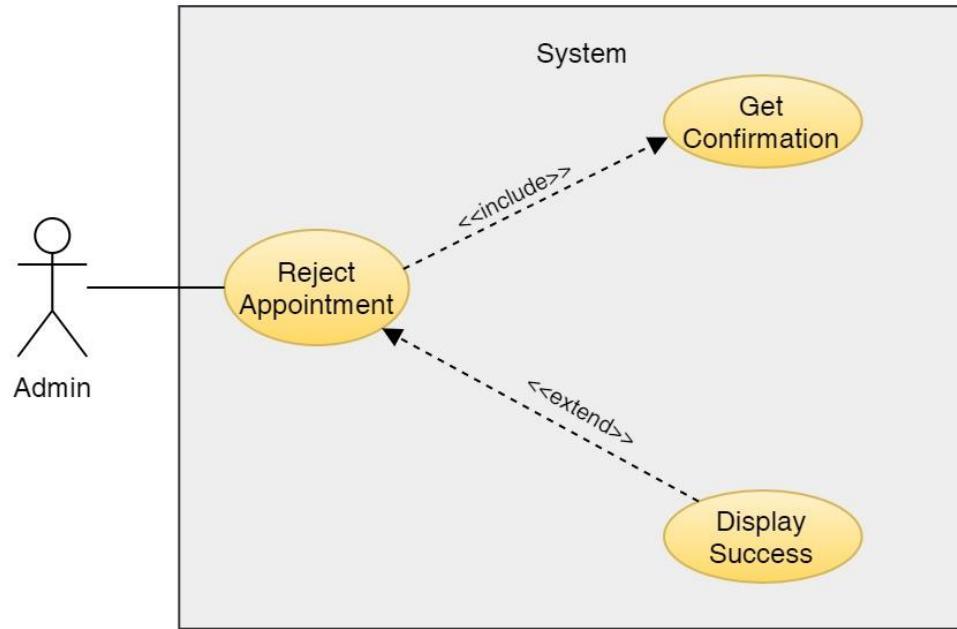


Figure 355: Use Case Diagram – Admin Reject Appointment

Collaboration Diagram

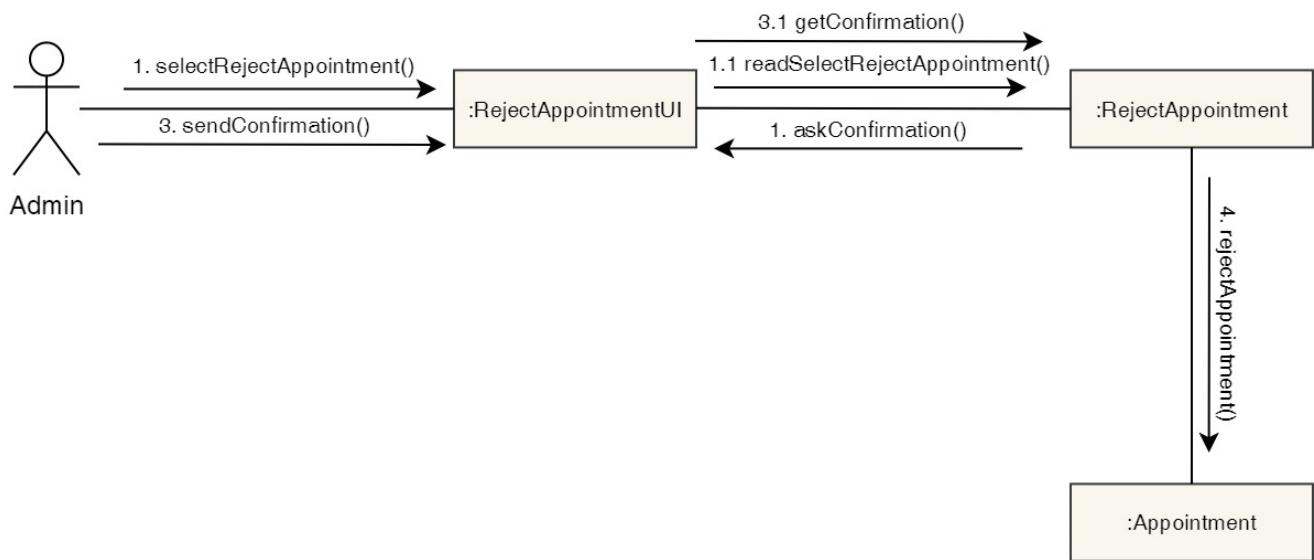


Figure 356: Collaboration Diagram – Admin Reject Appointment

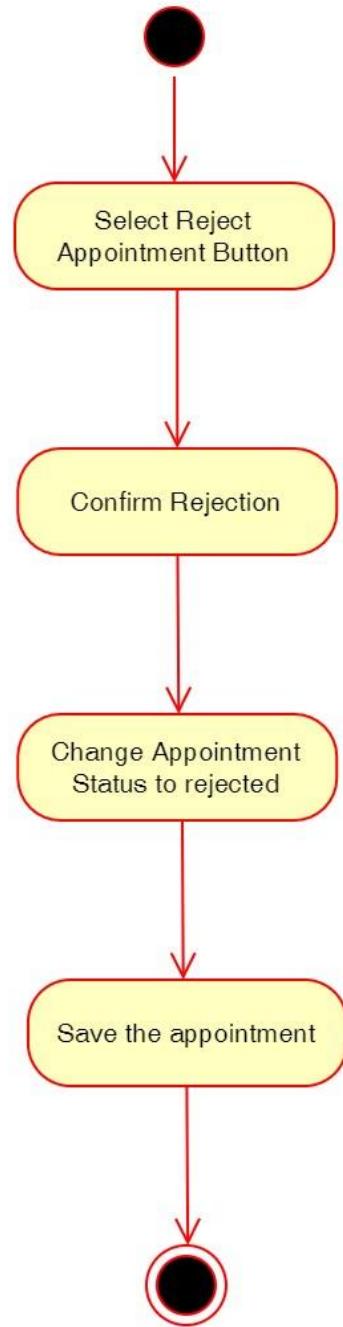
Activity Diagram

Figure 357: Activity Diagram – Admin Reject Appointment

Sequence Diagram

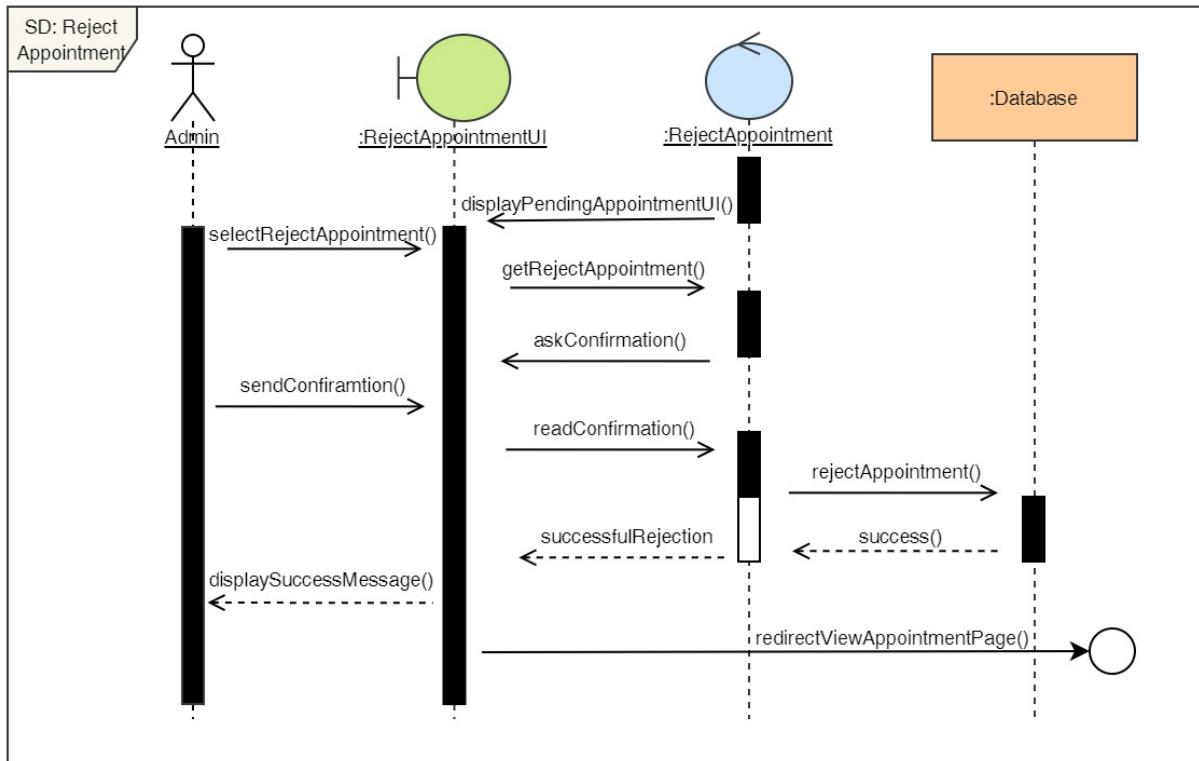


Figure 358: Sequence Diagram – Admin Reject Appointment

27. Finish Appointment

High Level Use Case Description

Use Case:	Finish Appointment
Actor:	Admin
Description:	After the appointment is confirmed, it is in confirmed stage. After the user visits the clinic for their appointment and the appointment is completed, the admin changes the appointment status from confirmed to finished.

Table 56: High Level Use Case – Admin Finish Appointment

Use Case Diagram

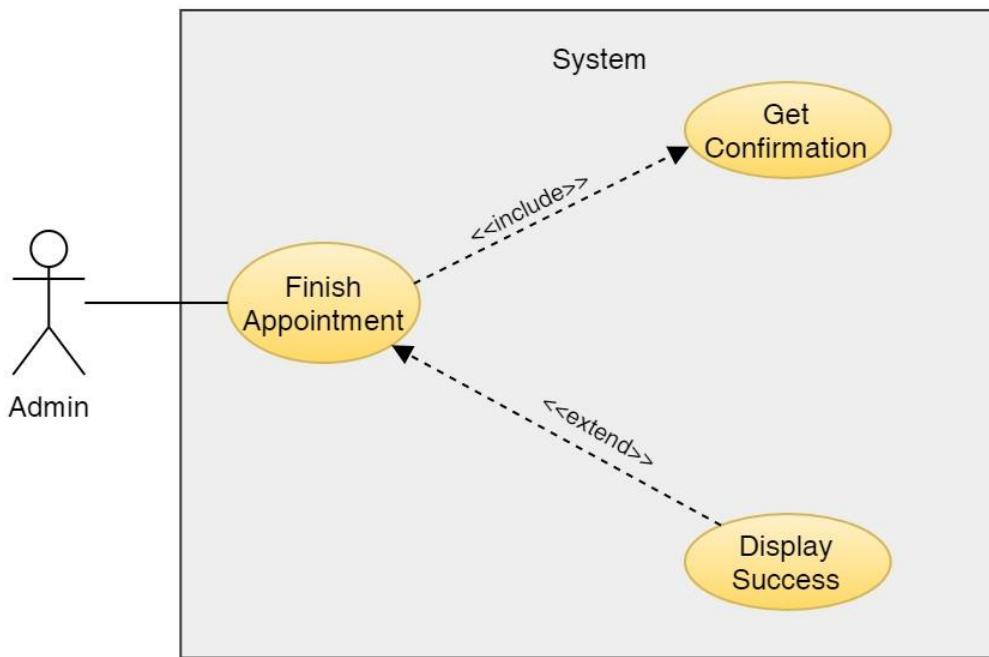


Figure 359: Use Case Diagram – Admin Finish Appointment

Collaboration Diagram

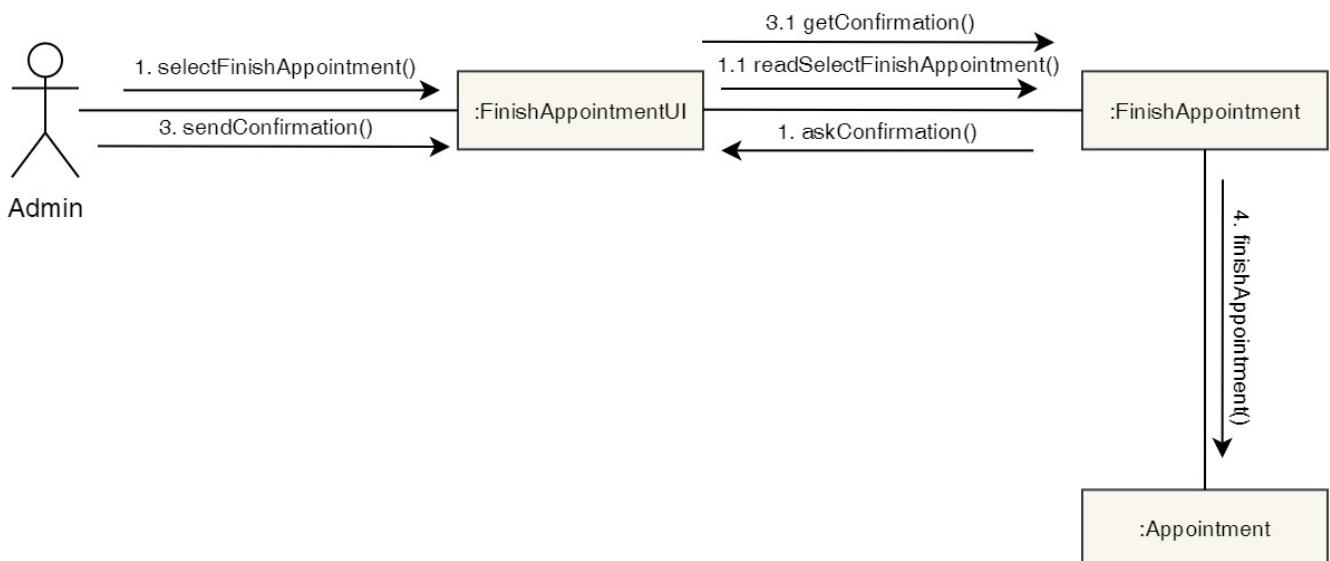


Figure 360: Collaboration Diagram – Admin Finish Appointment

Activity Diagram

Figure 361: Activity Diagram – Admin Finish Appointment

Sequence Diagram

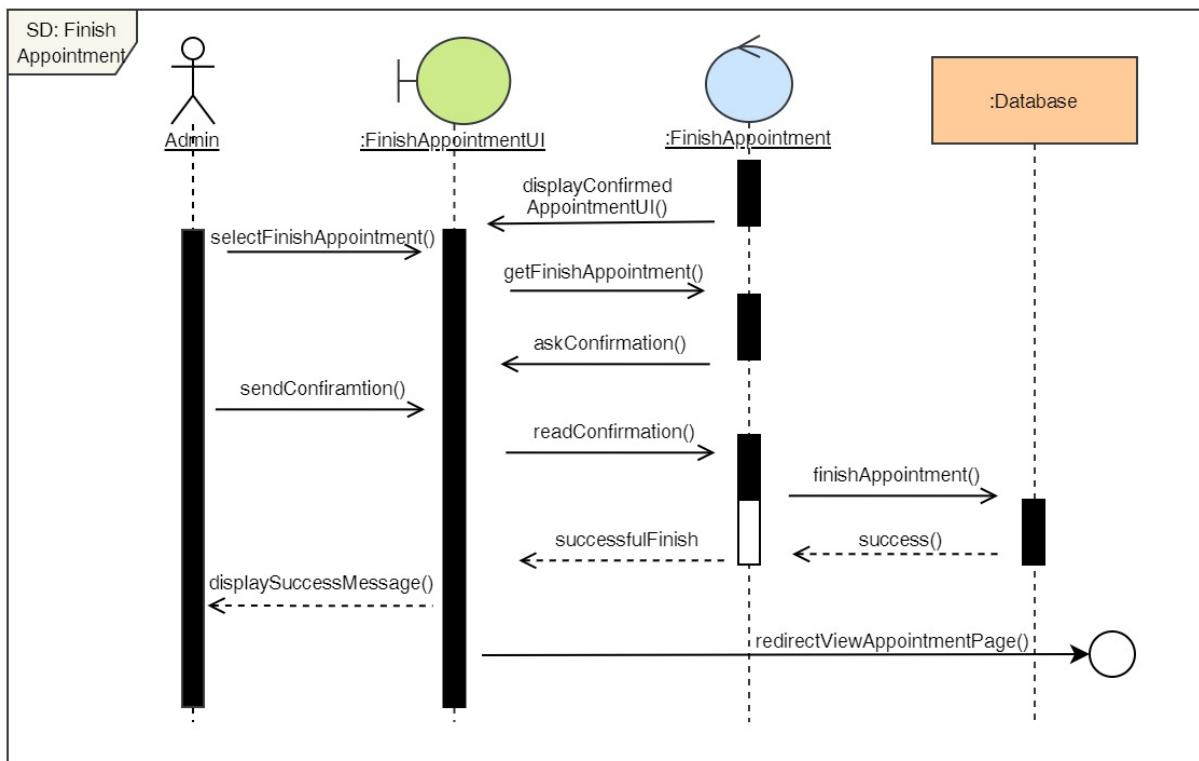


Figure 362: Sequence Diagram – Admin Finish Appointment

28. View all services

High Level Use Case Description

Use Case:	View Services
Actor:	Admin
Description:	Admin can view the list of all the services available along with the cost of it and description.

Table 57: High Level Use Case – Admin View Services

Use Case Diagram

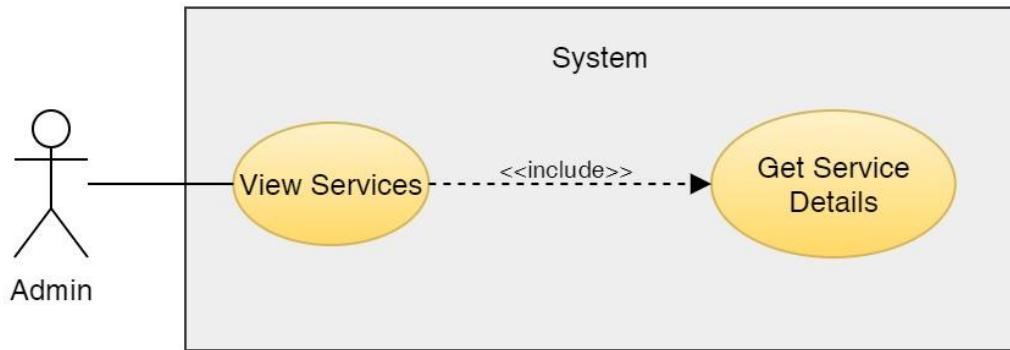


Figure 363: Use Case Diagram – Admin View Services

Collaboration Diagram

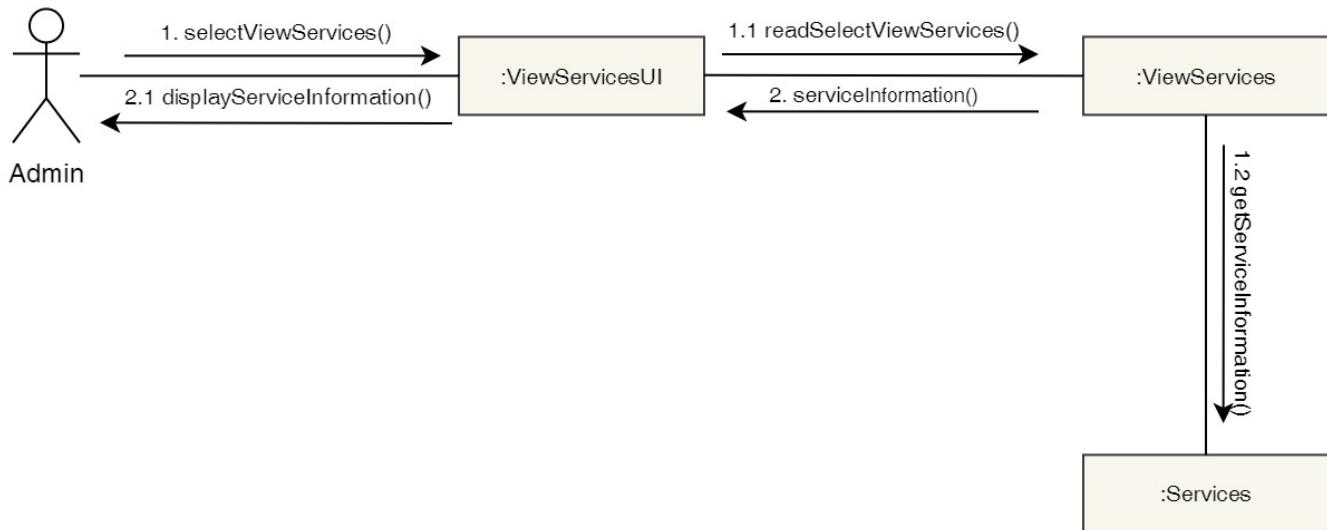


Figure 364: Collaboration Diagram – Admin View Services

Activity Diagram

Figure 365: Activity Diagram – Admin View Services

Sequence Diagram

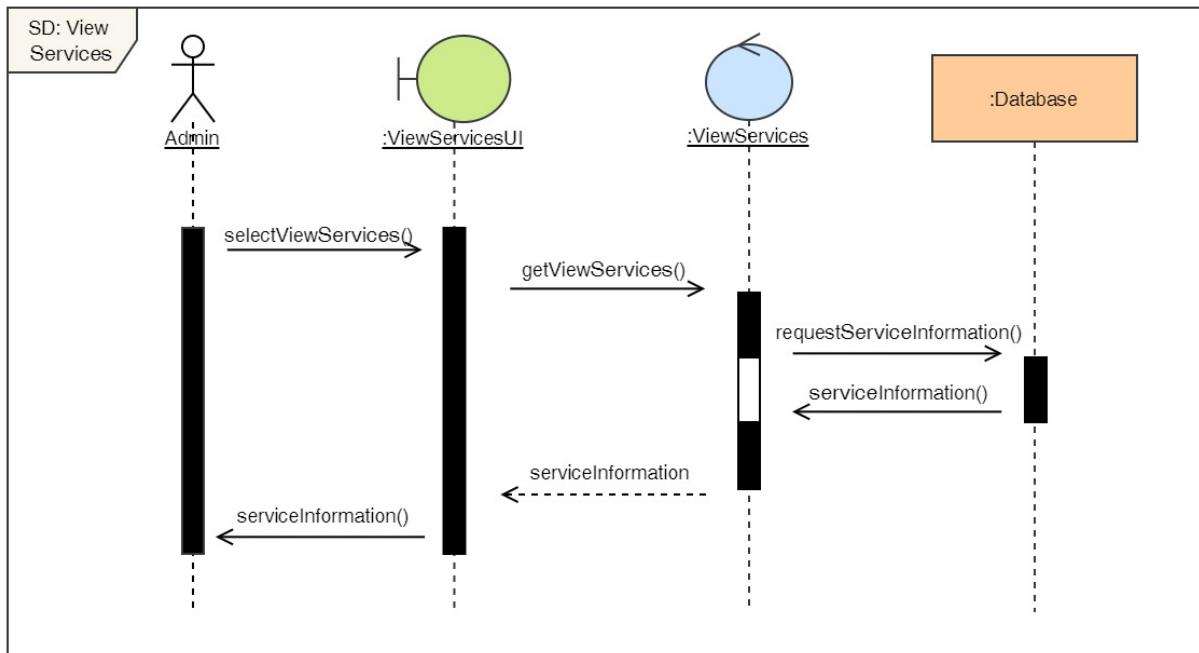


Figure 366: Sequence Diagram – Admin View Services

29. Add services

High Level Use Case Description

Use Case:	Add services
Actor:	Admin
Description:	Admin can add new services in the system.

Table 58: High Level Use Case – Admin Add Services

Use Case Diagram

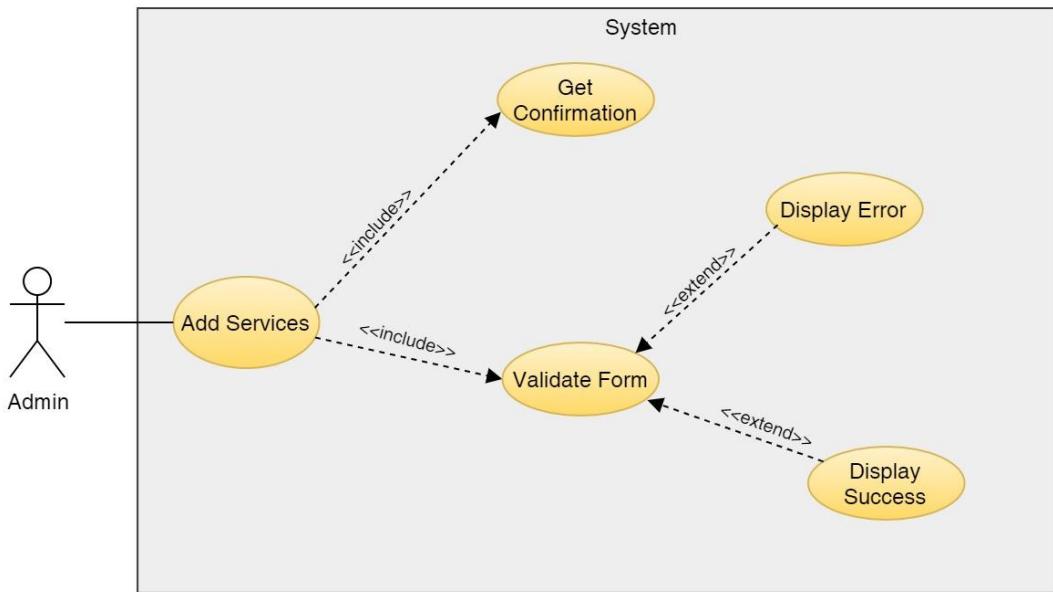


Figure 367: Use Case Diagram – Admin Add Services

Collaboration Diagram

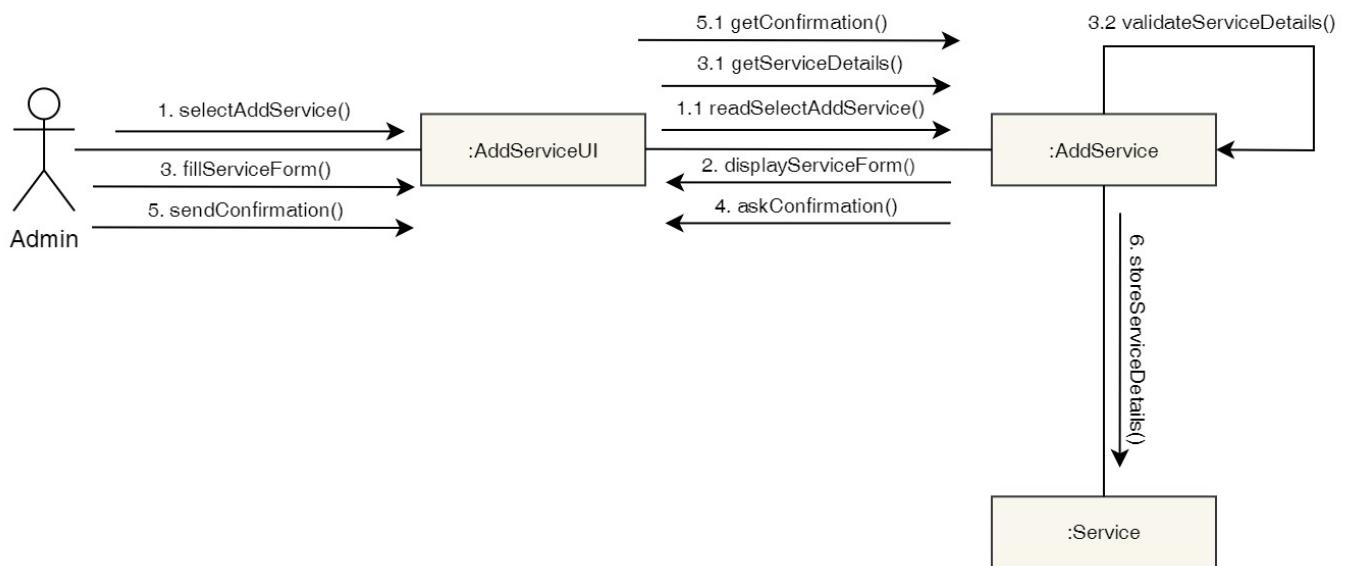
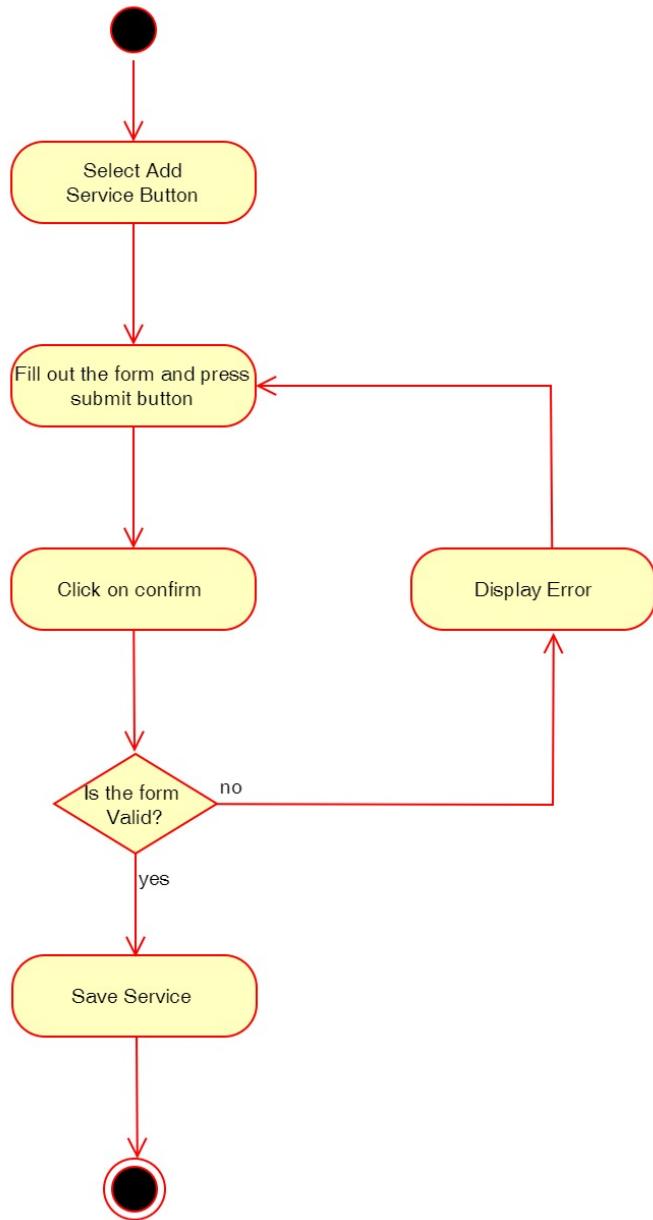


Figure 368: Collaboration Diagram – Admin Add Services

Activity Diagram*Figure 369: Activity Diagram – Admin Add Services*

Sequence Diagram

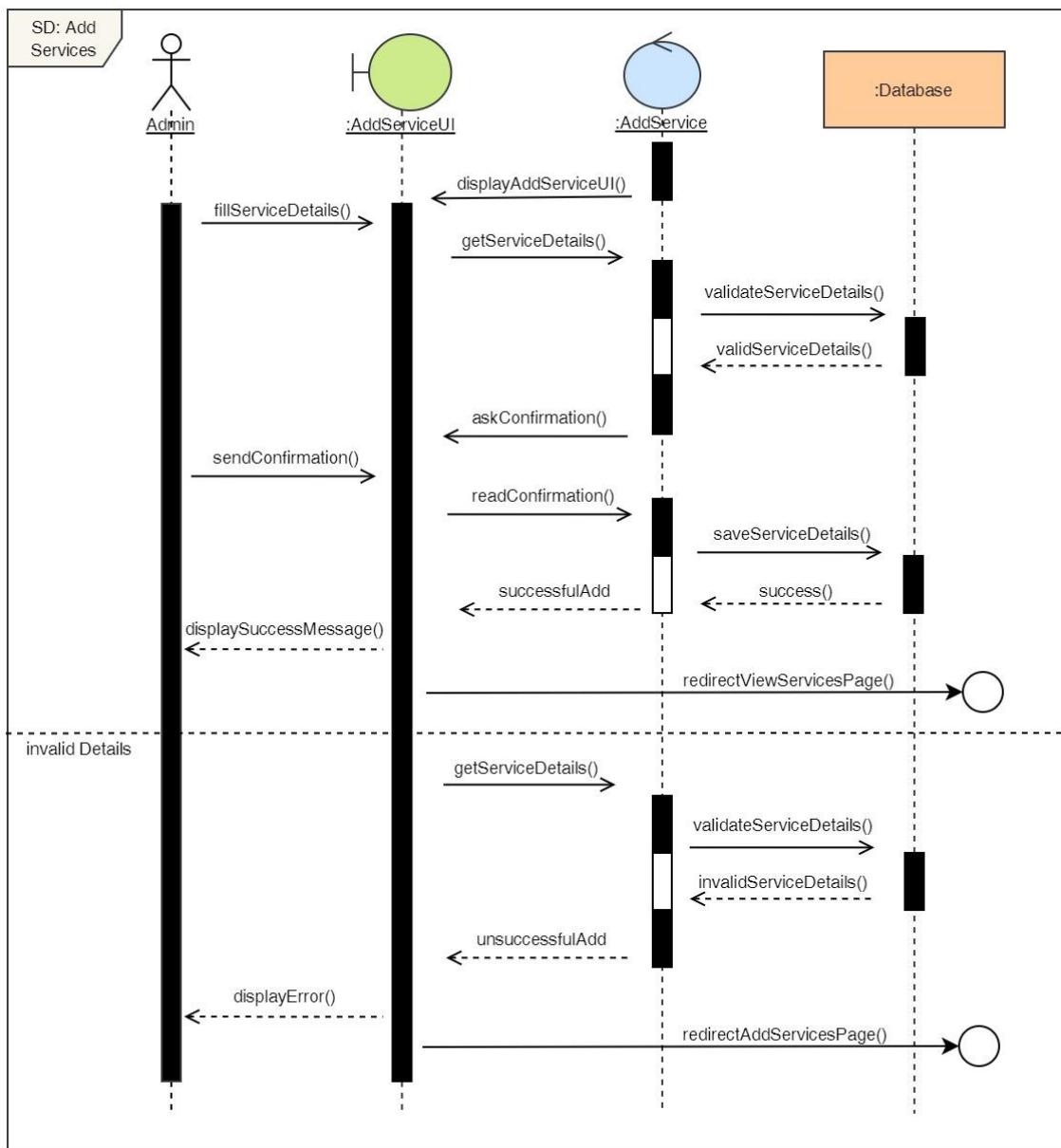
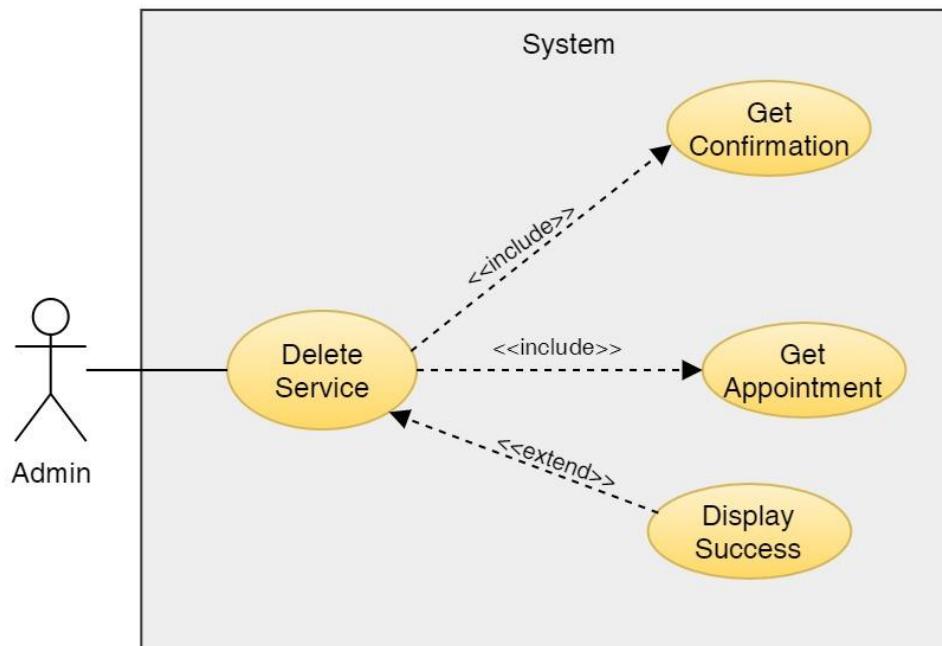


Figure 370: Sequence Diagram – Admin Add Services

30. Delete services

High Level Use Case Description

Use Case:	Delete services
Actor:	Admin
Description:	Admin can add remove existing services from the system.

*Table 59: High Level Use Case – Admin Delete Services***Use Case Diagram***Figure 371: Use Case Diagram – Admin Delete Services*

Collaboration Diagram

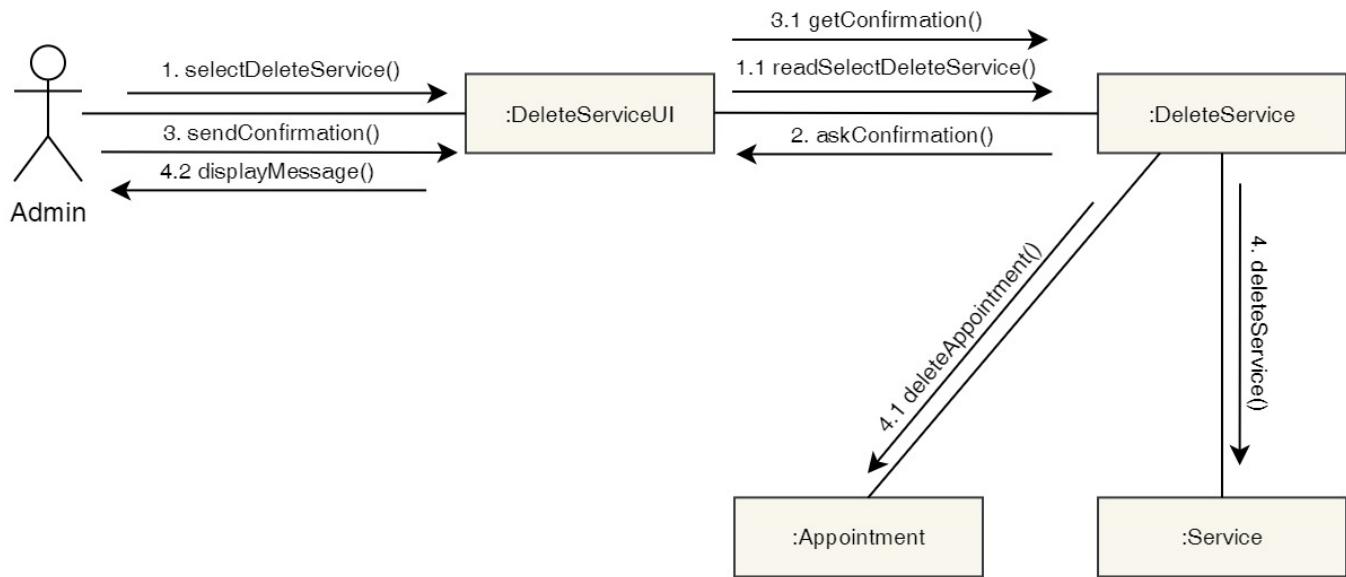


Figure 372: Collaboration Diagram – Admin Delete Services

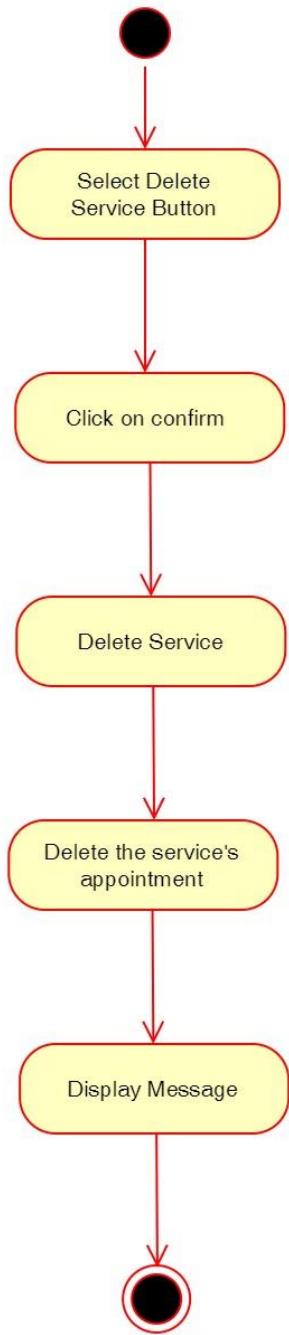
Activity Diagram

Figure 373: Activity Diagram – Admin Delete Services

Sequence Diagram

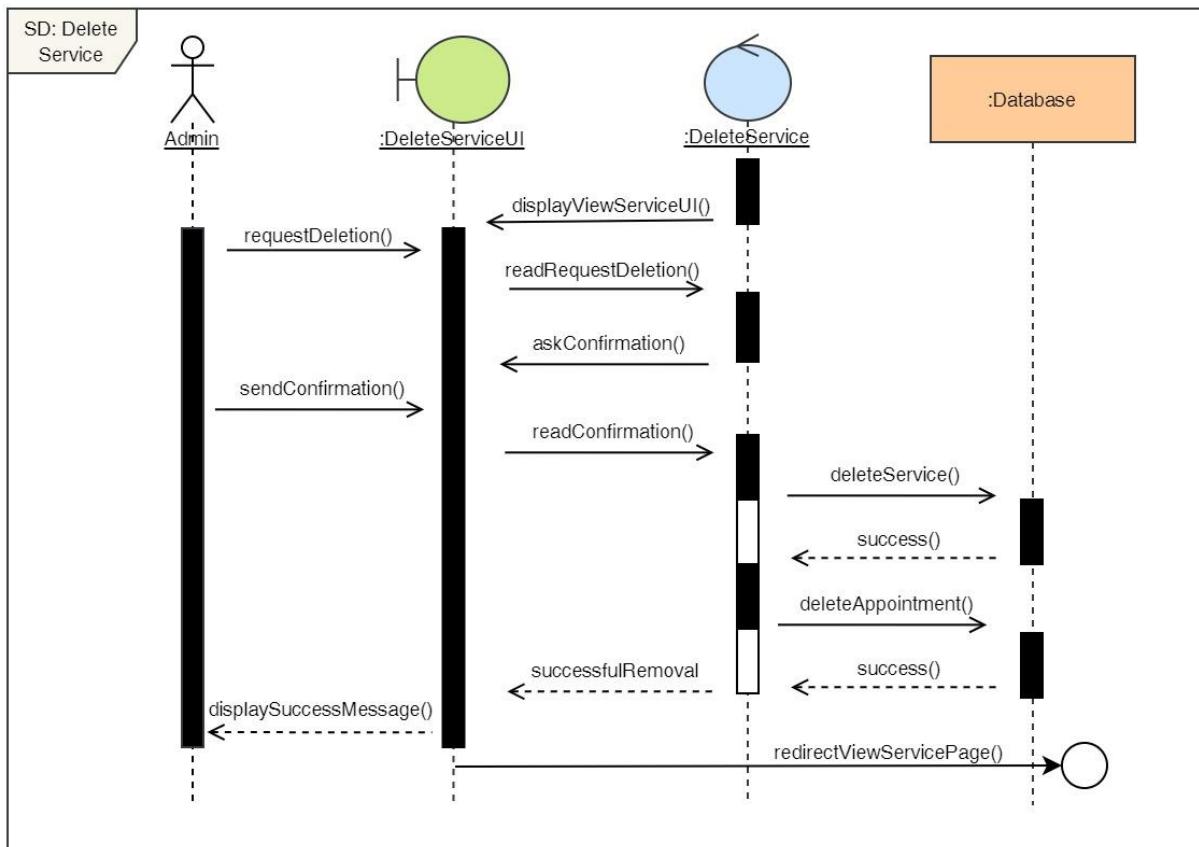


Figure 374: Sequence Diagram – Admin Delete Services

31. Edit services

High Level Use Case Description

Use Case:	Edit services
Actor:	Admin
Description:	Admin can update the details of the existing services in the system.

Table 60: High Level Use Case – Admin Edit Services

Use Case Diagram

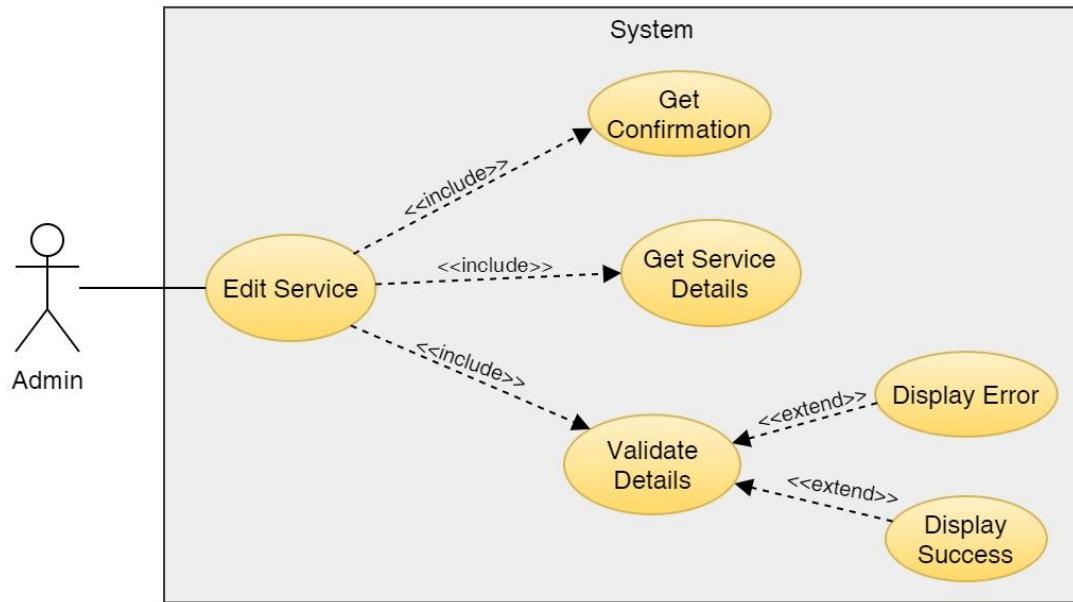


Figure 375: Use Case Diagram – Admin Edit Service

Collaboration Diagram

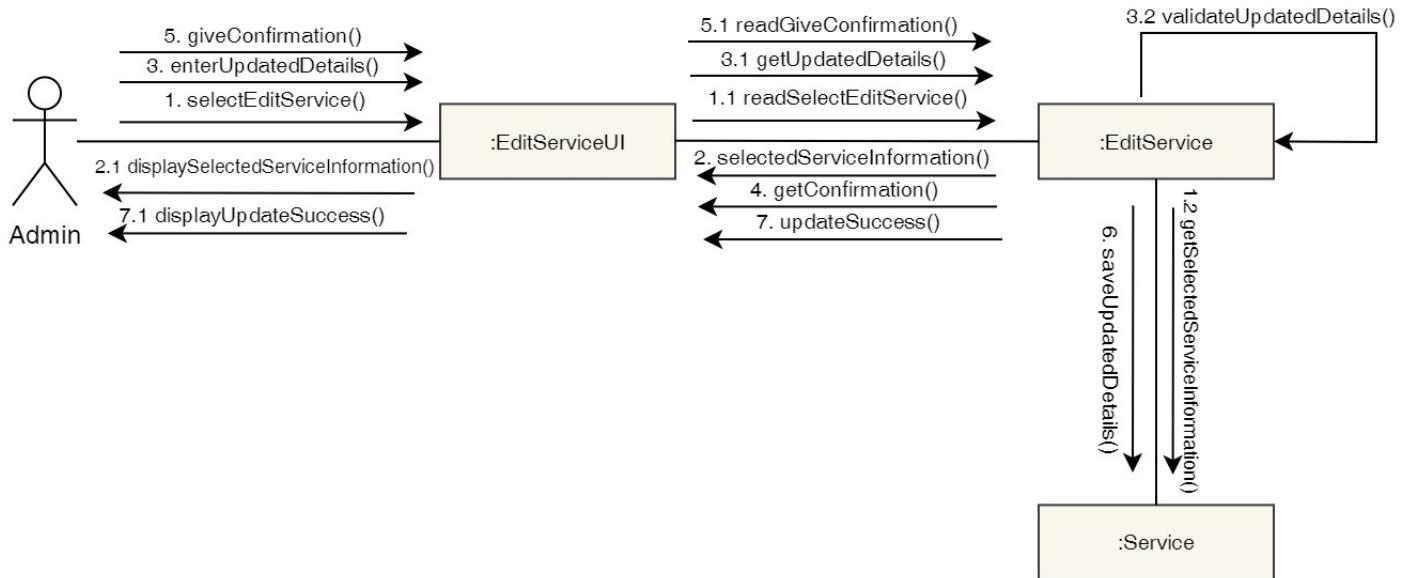


Figure 376: Collaboration Diagram – Admin Edit Service

Activity Diagram

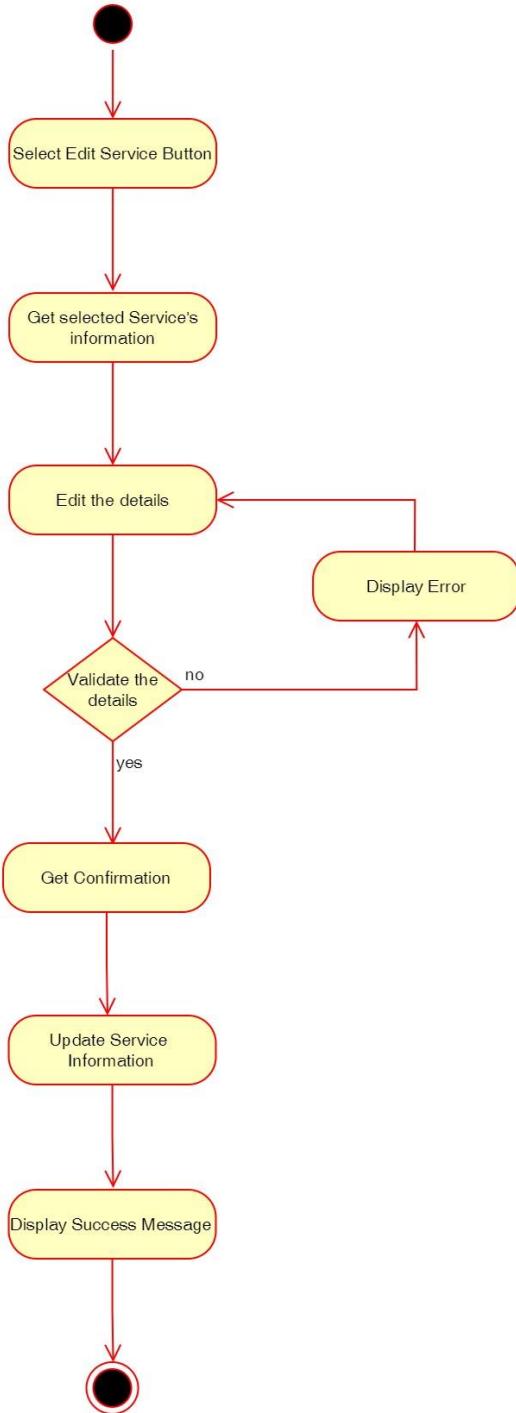


Figure 377: Activity Diagram – Admin Edit Service

Sequence Diagram

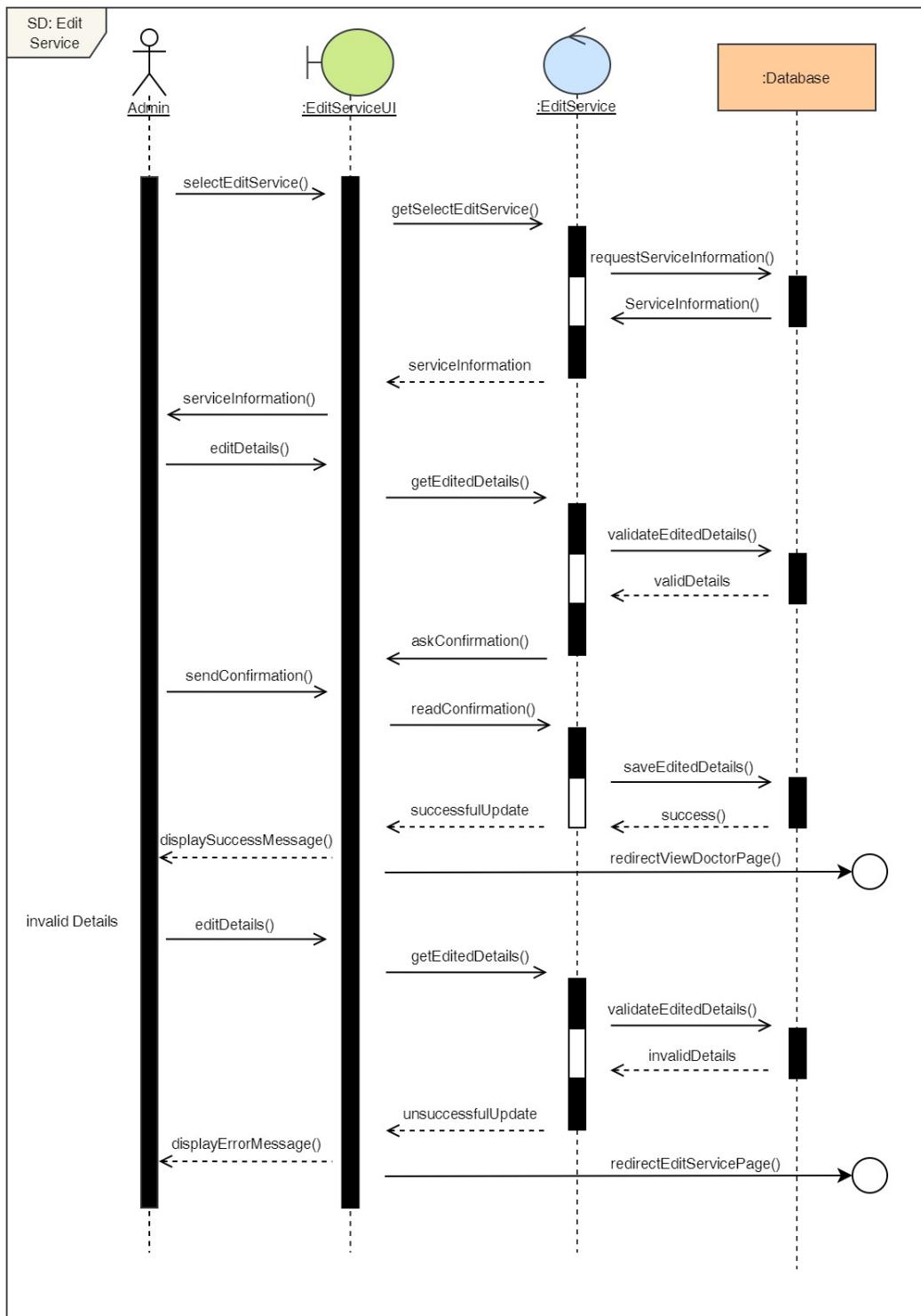


Figure 378: Sequence Diagram – Admin Edit Service

[Click here to view the user side functional UML Diagrams.](#)

[Click here to view the overall Use Case Diagram.](#)

7.6.2 Gantt Chart

A Gantt chart, commonly used in project management, is one of the most popular and useful ways of showing activities (tasks or events) displayed against time. (Gantt.com, 2022) The revised Gantt chart for the following Dental Appointment System is given below:

Dental Appointment System
Read-only view, generated on 26 Apr 2022

 Instagantt

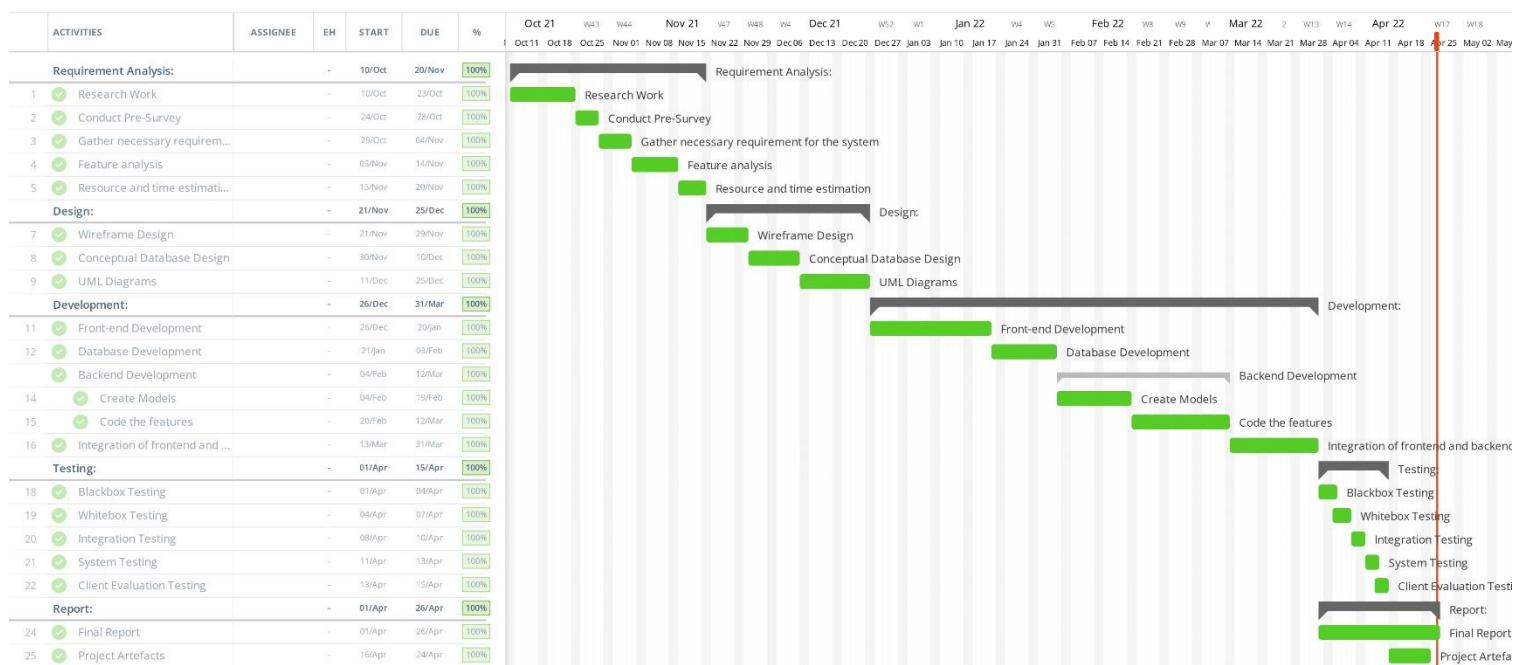


Figure 379: Gantt chart

Here, we can see the tasks according to the methodology. The Requirement Analysis, Design, Development and Testing are the four phases of the incremental methodology. The development portion has already been completed and the report part is still ongoing while the Gantt chart was created.

7.6.3 Work Breakdown Structure

A work breakdown structure, or WBS, is a project management tool that takes a step-by-step approach to complete large projects with several moving pieces. (Organ, 2022) A Work Breakdown Structure entails breaking down a huge, complicated project into smaller, more manageable, and self-contained jobs. The Project name is used to designate the root of this tree (structure). Each node is recursively divided into smaller sub-activities for the purpose of developing a work breakdown structure, until the activities become undividable and autonomous at the leaf level. It follows a Top-Down strategy. (Raman, 2019) The Work Breakdown Structure for the Dental Appointment System is given below:

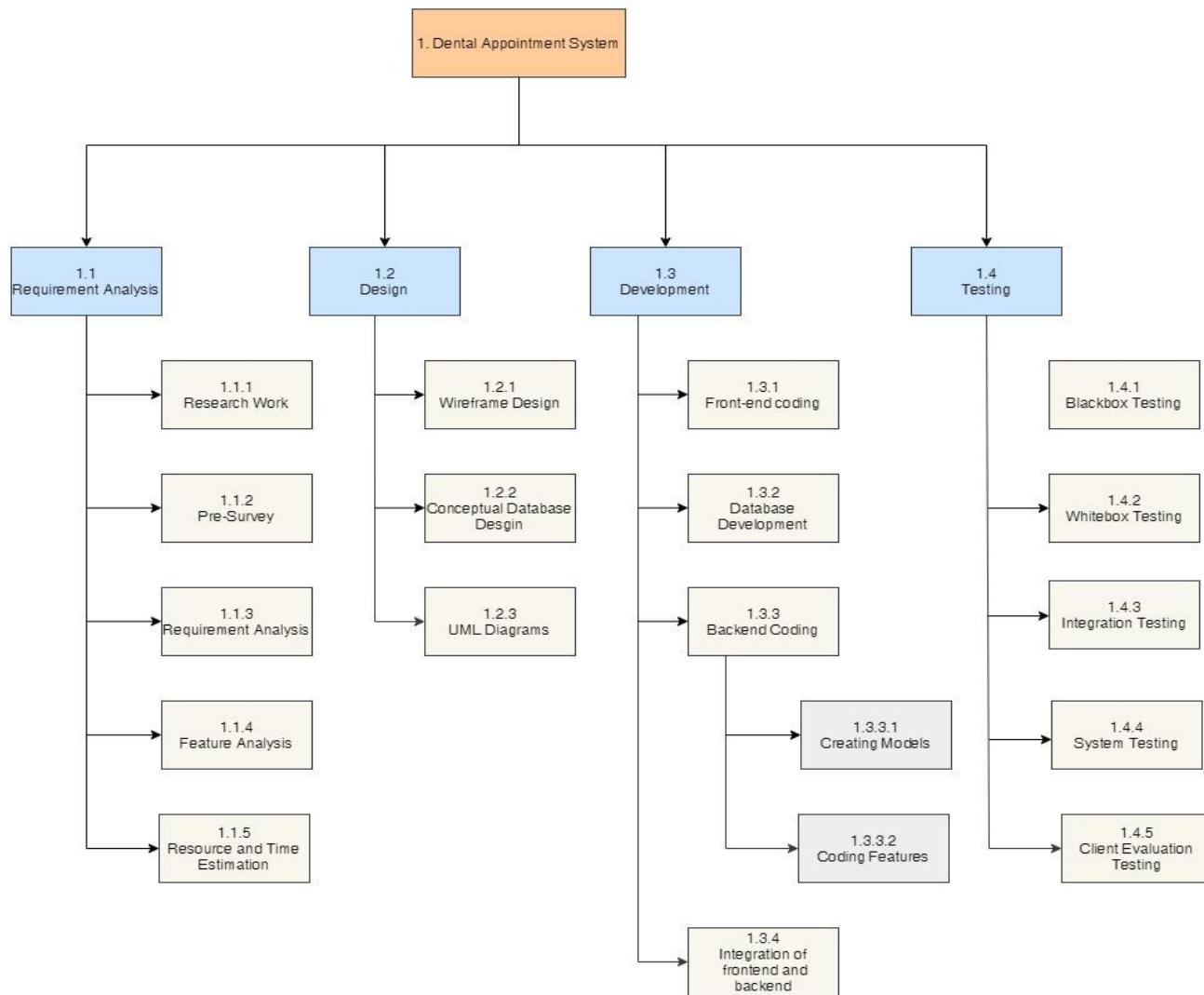


Figure 380: Work Breakdown Structure

7.6.4 Algorithms and Flowcharts

7.6.4.1 Algorithm

In computer programming terms, an algorithm is a set of well-defined instructions to solve a particular problem. It takes a set of input and produces a desired output. (Programiz, 2022) In this application, the bubble sort algorithm has been used to sort the data that has been used to display in the table. The algorithm was used to sort the details of the doctor, appointment, report, services and users data to display.

Bubble sort is a very simple algorithm that used for sorting. This sorting algorithm is comparison-based algorithm in which each pair of adjacent elements is compared and the elements are swapped if they are not in order. (TutorialsPoint, 2022)

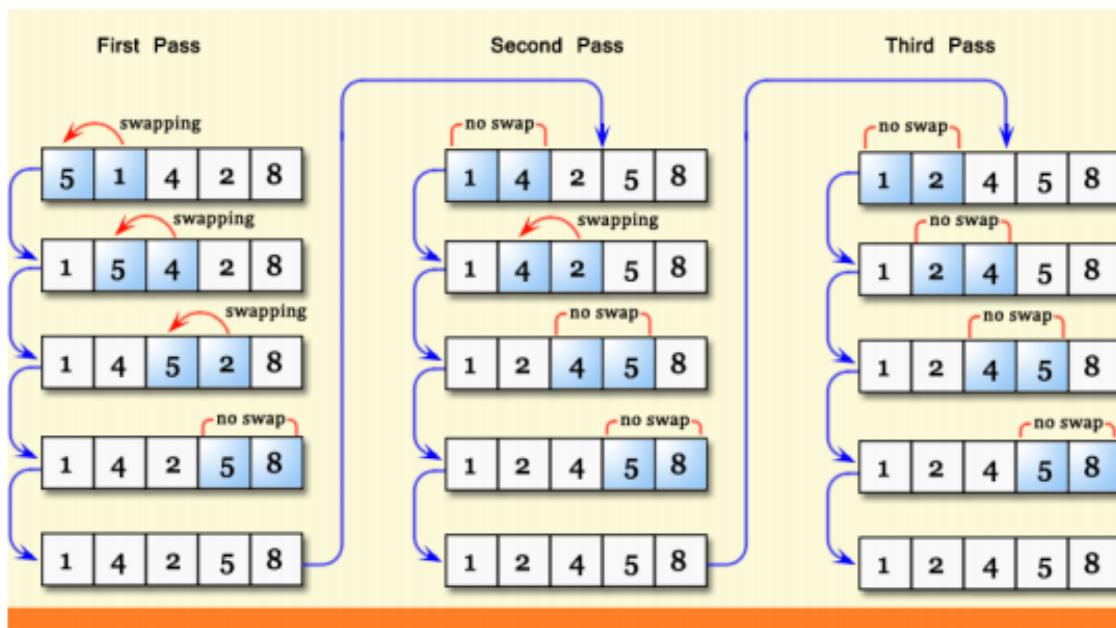


Figure 381: Bubble Sort Algorithm

The image above shows how the bubble sort algorithm works. It repeatedly swaps the adjacent elements if they are in wrong order and keeps on doing it until the elements are in correct order.

The pseudocode for the bubble sort algorithm is given below:

```
procedure bubbleSort( list : array of items )
```

```
    loop = list.count;
```

```
    for i = 0 to loop-1 do:
```

```
        swapped = false
```

```
        for j = 0 to loop-1 do:
```

```
            /* compare the adjacent elements */
```

```
            if list[j] > list[j+1] then
```

```
                /* swap them */
```

```
                swap( list[j], list[j+1] )
```

```
                swapped = true
```

```
            end if
```

```
        end for
```

```
        /*if no number was swapped that means
```

```
        array is sorted now, break the loop.*/
```

```
        if(not swapped) then
```

```
            break
```

```
        end if
```

```
    end for
```

```
end procedure return list
```

7.6.4.2 Flowchart

Flowcharts are simply graphical representations of data or algorithms for a better visual comprehension of the code. It shows how to solve an issue, algorithm, or procedure step by step. It is a pictorial way of representing steps that are preferred by most beginner-level programmers to understand algorithms of computer science, thus it contributes to troubleshooting the issues in the algorithm. (Mukherjee, 2021)

The overall flowchart diagram based only on the user side experience is given below:

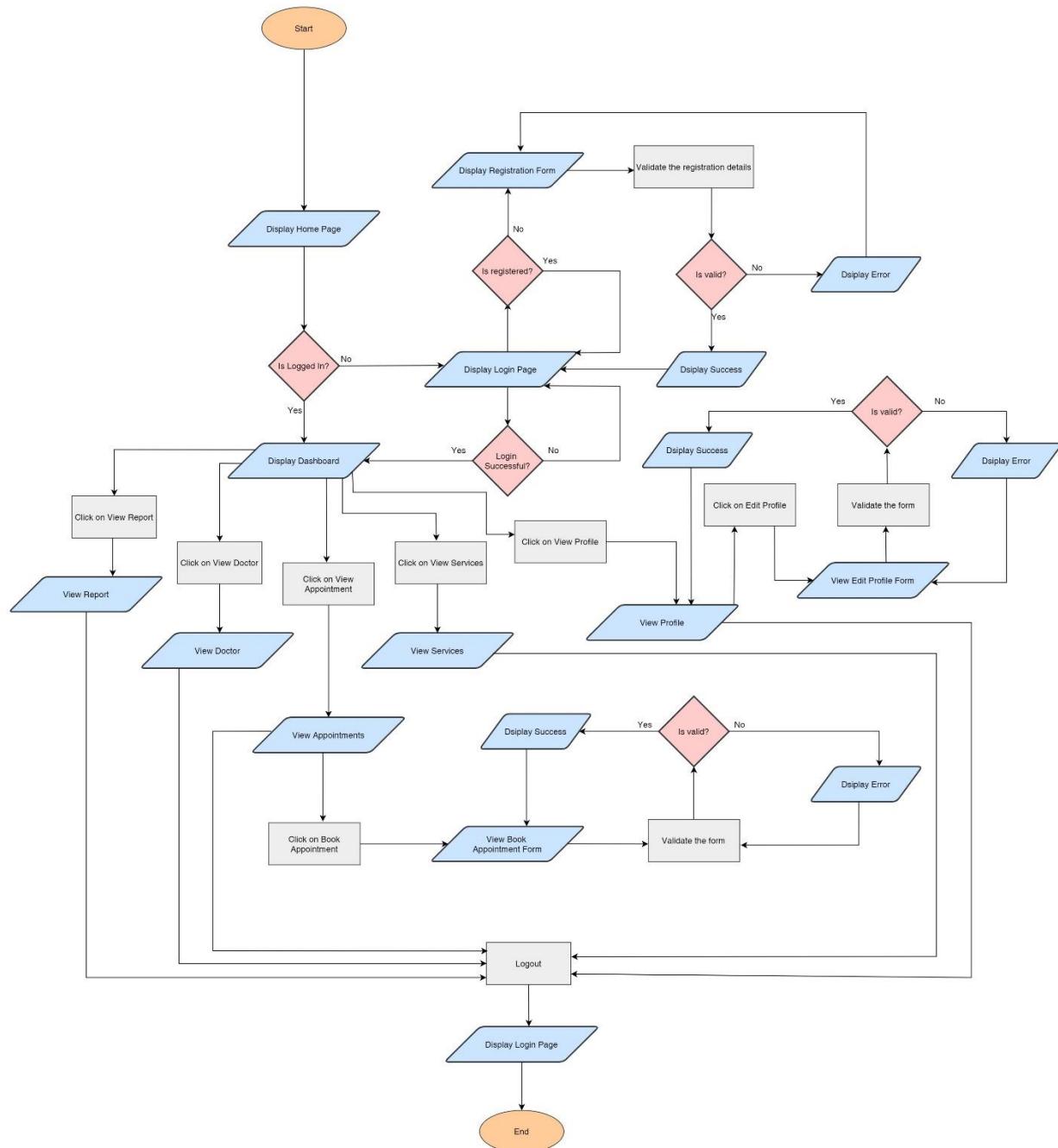


Figure 382: Flowchart of the overall system

7.6.5 ER Diagram Creation Process

The ERD creation process is elaborated below:

1. Possible Entities

The possible entities for Dental Appointment System as:

- i. User – To store information about the users/admin for authentication.
- ii. patientUser – To store additional information about the users.
- iii. Appointments – To store the details about the appointment done by user.
- iv. Services – To store the details about the services available.
- v. Doctors – To store the details about the doctor.
- vi. Report – To store the report of the appointment.

2. Relationship Between Entities

i. User and patientUser:

All the patientUser must be a User but a User can either be a patientUser or not. They can also be an admin who does not have extra fields. So, the relation between User and patientUser will be one mandatory to one optional. The entity along with relation is shown below:



Figure 383: User-patientUser Relation

ii. patientUser and Appointment:

A user can have multiple appointment but an appointment can only belong to one patientUser. So the relation between patientUser and Appointment is One Mandatory to Many.



Figure 384: patientUser-Appointment Relation

iii. Report and Appointment:

A report can only belong to one appointment and an appointment can only have one report. So, the relation between report and appointment will be one mandatory to one mandatory.



Figure 385: Report-Appointment Relation

iv. Doctor and Appointment:

A doctor can oversee many appointments but an appointment can only have one doctor. So, the relation between Doctor and Appointment will be one mandatory to many.

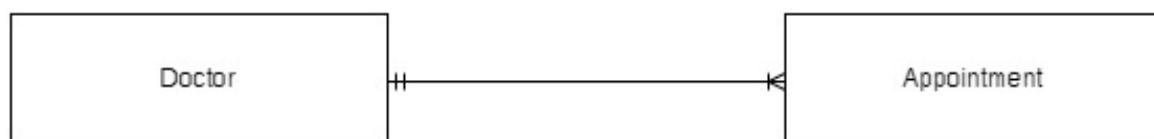


Figure 386: Doctor-Appointment Relation

v. Service and Appointment

An appointment can only be made for one service but a service can have multiple appointments. So, the relation between Services and Appointment will also be one mandatory to many.



Figure 387: Service-Appointment Relation

3. Attributes

i. User

Attributes	Data Type	Others
ID	Integer	Primary Key, Not Null, Auto Increment
First_name	Varchar(150)	Not Null
Last_name	Varchar(150)	Not Null
Email	Varchar(254)	Not Null, Unique
Username	Varchar(150)	Not Null, Unique
Password	Varchar(128)	Not Null

Table 61: User Attributes

ii. patientUser

Attributes	Data Type	Others
ID	Integer	Primary Key, Not Null, Auto Increment
Date_of_birth	Date	Not Null
Phone_number	Varchar(15)	Not Null
Profile_picture	Varchar(100)	Not Null, Default="img/defaultuser.png"
User_id	Integer	Foreign Key, Unique

Table 62: patientUser Attributes

iii. Doctor

Attributes	Data Type	Others
ID	Integer	Primary Key, Not Null, Auto Increment
Full_name	Varchar(100)	Not Null
Qualification	Varchar(50)	Not Null
Date_of_birth	Date	Not Null
Email	Varchar(100)	Not Null
Phone_number	Varchar(15)	Not Null
Address	Varchar(50)	Not Null
Experience	Varchar(10)	Not Null, Default="1 year"
Profile_picture	Varchar(100)	

Table 63: Doctor Attribute

iv. Service

Attributes	Data Type	Others
ID	Integer	Primary Key, Not Null, Auto Increment
Service_name	Varchar(50)	Not Null
description	Text	Not Null
Cost	Integer	Not Null

Table 64: Service Attribute

v. Appointment

Attributes	Data Type	Others
ID	Integer	Primary Key, Not Null, Auto Increment
Date	Date	Not Null
Time	Time	Not Null
Description	Text	Not Null
Doctor_id	BIGINT	Foreign Key, Not Null
patient_ser_id	BIGINT	Foreign Key, Not Null
Service_id	BIGINT	Foreign Key, Not Null
Status	Integer	Not Null, Default="1"

Table 65: Appointment attribute

vi. Report

Attributes	Data Type	Others
ID	Integer	Primary Key, Not Null, Auto Increment
Problems	Text	Not Null
Prescription	Text	Not Null
Other_details	Text	Not Null
Appointment_id	BIGINT	Foreign Key, Not Null, Unique

Table 66: Report attribute

Final ERD

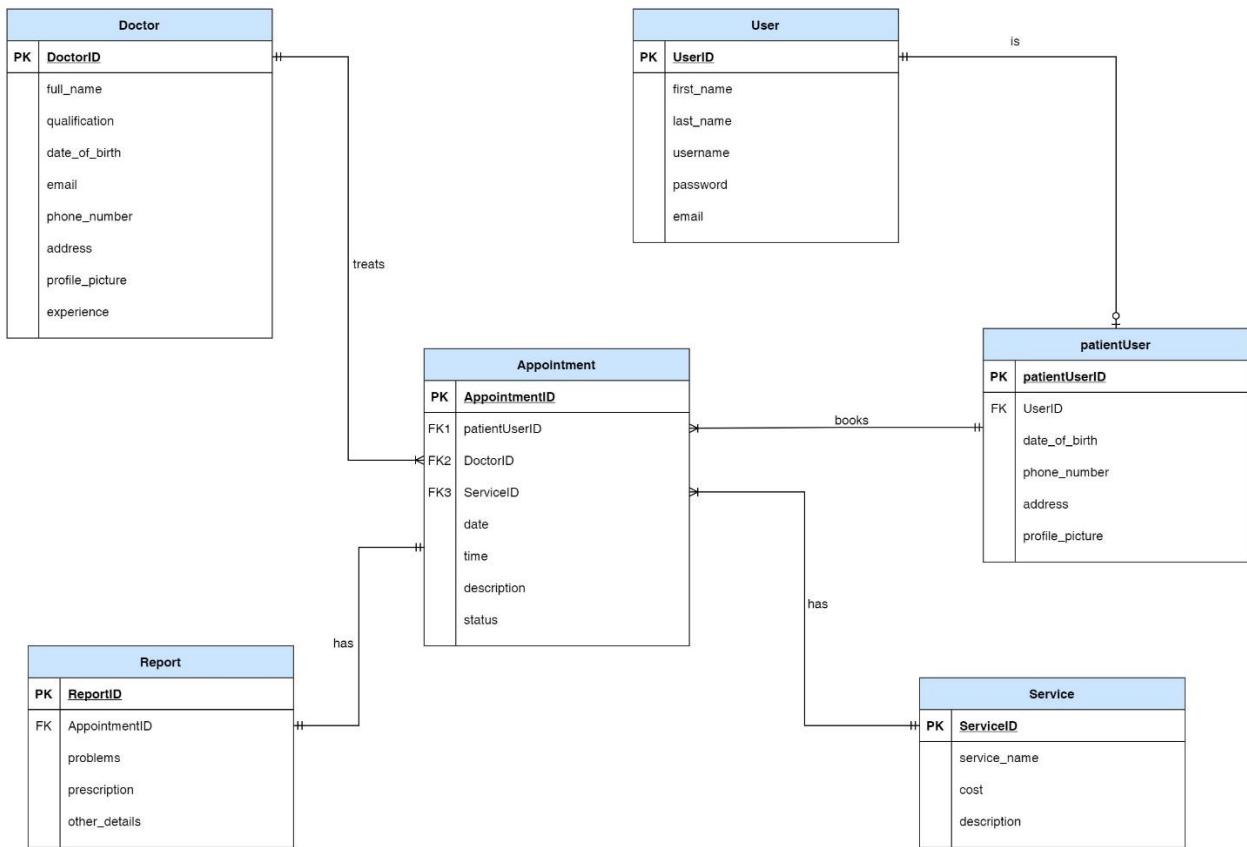


Figure 388: Final ERD:

Here, we can see User is a default database table of Django which has limited fields. So, to extend it, we created a new table called patientUser i.e. all patientUser must be a user but a user can either be a patientUser or an admin. There is Doctor and services table. The appointment table takes patientUser, services and doctor as a foreign key. A doctor treats more than one appointment but an appointment can only belong to one doctor. Similarly, a patientUser can book many appointment but an appointment can only belong to one patientUser. An appointment also has one service but a service can have multiple appointment. The report then takes appointment as a foreign key. A report can belong to only one appointment and an appointment can only have one report.

7.6.6 Creation of Use Case Diagram

Use case diagram shows the functionality and the actors which uses the functionality in a system. When creating a use case, an active verb or noun is used. When creating a use case diagram, the actor and the use case that the actor requires are defined.

- **System**

The project that we plan to build is called System. The method is defined by a rectangular shape in the use case diagram. It establishes the overall system's reach. The name of the system is always shown at the top.



Figure 389: Rectangle which represents System

- **Actors**

In the use case diagram, the actor is portrayed by a human stick picture. The device users of the software system are referred to as actors. Actors are always kept outside the system. Primary and Secondary actors are the two groups of actors. The system's primary actors are the system's creators. It makes direct requests to the machine. Whereas secondary actors communicate less with the system and only responds to the primary actors.

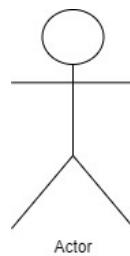


Figure 390: Human stick figure which represents Actor

- **Use Case**

It is the list of tasks that are performed by the system. It is usually done in logical order. It is represented by oval.

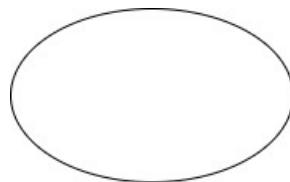


Figure 391: Oval which represents Use Case

- **Relation between actor and use case**

The relation between actor and use case is demonstrated by a solid line. It is called association relationship.

Figure 392: Solid line which represents the relation between actor and use case

- **Relationships**

1. Include Relation

It is the relation which is performed immediately. Use case that is auto executed with base use is include relation. The arrow points to the use case that includes it. It is represented by the following diagram:



Figure 393: Include Relation

2. Extend Relation

It is used to indicate that one use case extends the functionality of another use case. The arrow points to the base use case. It is represented by the following diagram:



Figure 394: Extend Relation

3. Generalisation Relation

It is the inheritance relation. Generalisation relation is the relation between parent and child.

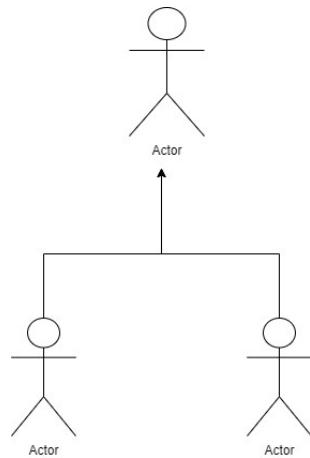


Figure 395: Generalisation Relation

[Click here to see the overall use case diagram of the system.](#)

7.6.7 Class Diagram

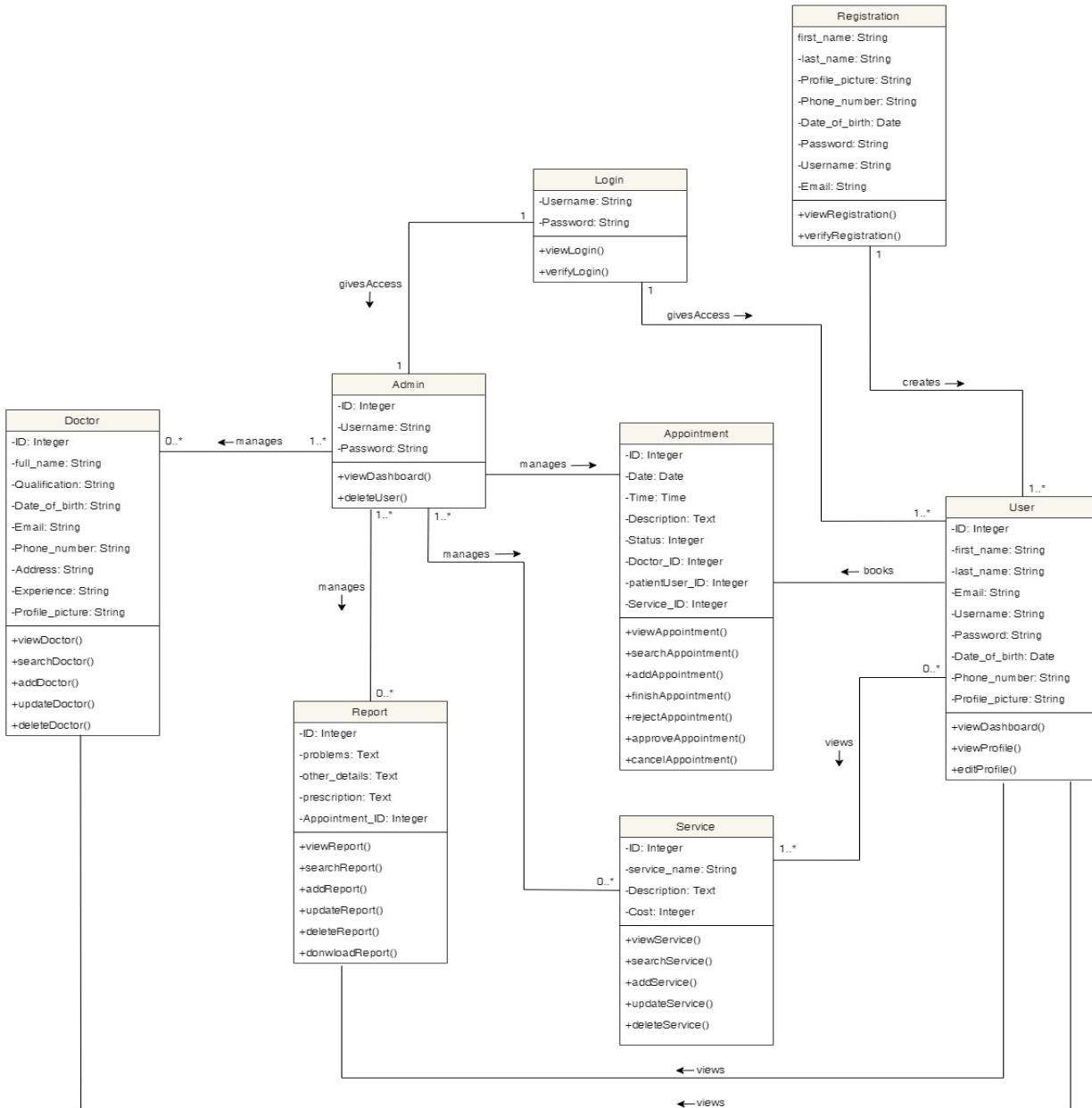


Figure 396: Class Diagram

7.6.8 Software Requirement Specification (SRS)

7.6.8.1 Introduction

Project: Dental Appointment System

Category: Web Application

7.6.8.1.1 Purpose

The primary purpose of this document is to provide a full description of the Dental Appointment System. It will explain the system's goal and characteristics, as well as the system's interfaces, what the system will perform, and the limitations under which it must operate. It provides a thorough explanation of the system's objectives, functionality, and user interface. It outlines how the application (system) will operate. This paper is designed for both system users and system developers.

7.6.8.1.2 Intended Audience

The intended audience for this system is the general public. Everyone has their dental needs that to be treated or cleaned. But not everyone has the time to go to the clinic and wait in line for hours. So, this system is intended for those people who do not wish to wait for hours for their check-up rather be more convenient and visit the clinic when they have the appointment time,

The SRS (Software requirement specification) is intended for:

- Client/User
- Developer
- Tester

7.6.8.1.3 Project Scope

The primary goal of this system is to manage users' appointments and to build a user-friendly application for booking appointments. This program assists users in connecting to a database server and logging into the system. Above all, we believe in providing excellent service at a reasonable cost. This application will help to schedule an appointment and manage the work efficiently. It is a booking system that the user uses to schedule an appointment. The user can also view the doctors and services available along with the costs of services and the details of the doctors. Furthermore, users can view their report after their visit to the clinic. This helps avoid the loss of report that will be useful for the next meeting. This application is intended for a particular clinic only i.e. Phulchowki Dental Clinic.

The terms present in the documentations are:

- Admin: The person who is responsible for managing the whole system.
- User: A person who is using the application with limited privilege.
- Database: A server for the storing information of this system.

7.6.8.2 Overall Description

7.6.8.2.1 System Perspective

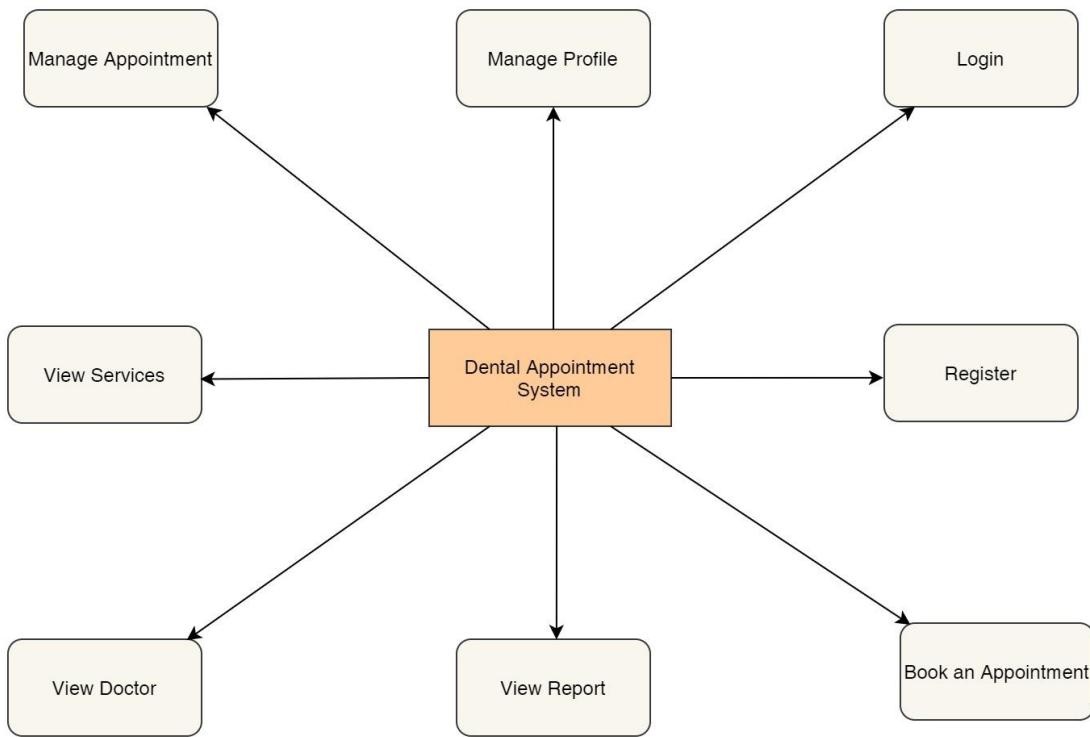


Figure 397: System Perspective Diagram

The features of the web based application are as mentioned below:

- SP1: Register
- SP2: Login
- SP3: Book an appointment
- SP4: View Report
- SP5: View Doctor
- SP6: View Services
- SP7: Manage Appointment
- SP8: Manage Profile

Use Case of the Overall System

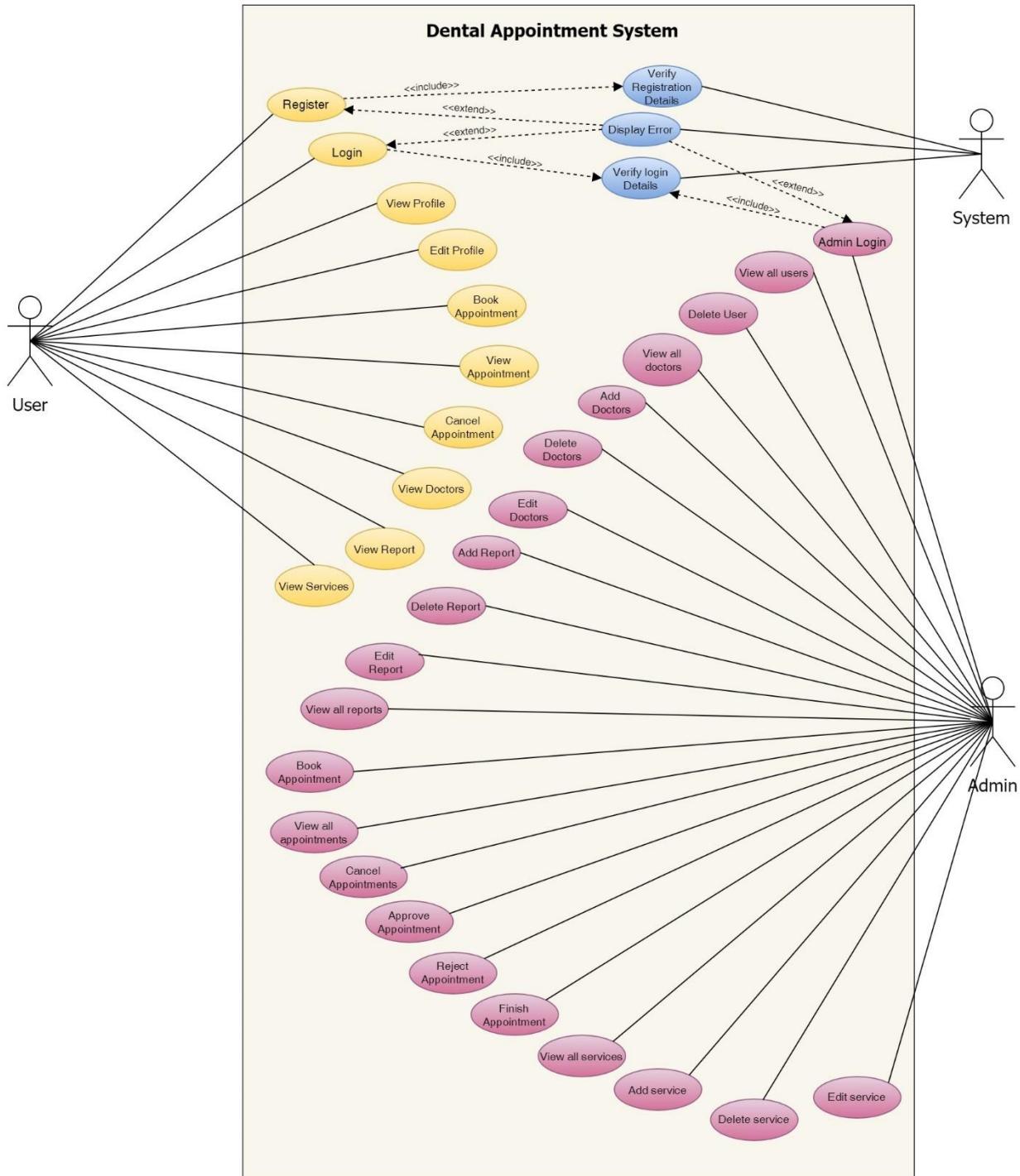


Figure 398: Overall Use Case of the System

7.6.8.2.2 System Features

SF1: Register

Description: New users who have not yet been registered to the system registers into it using the web browser. The registration details entered by the user is then verified by the system.

SF2: Login

Description: Registered Users can proceed to login to the system to use the features using the web browser. The login details entered by the customer is then verified by the system.

SF3: Book an Appointment

Description: Users can book an appointment for their dental requirements. The appointment is then reviewed by the admin.

SF4: View Report

Description: Users can view the report for their appointment only after it has been finished.

SF5: View Doctor

Description: Users can view the list of doctors available in the clinic along with their description and select the one they want to book an appointment with.

SF6: View Services

Description: Users can view the list of services available in the clinic along with the cost and description of it.

SF7: Manage Appointment

Description: Users can view the list of their appointments and cancel their pending as well confirmed appointment if the need arises to.

SF8: Manage Profile

Description: Users can view their profile and edit their details that they previously entered while registering. However, they cannot change their username.

7.4.5.2.3 User Class and Characteristics

The following are the classes for users who will be using the Dental Appointment System application with their roles:

UC1: Patient User

UC1.1: Book an appointment for their dental requirements

UC1.2: Browse doctors available in the clinic

UC1.3: View services available in the clinic

UC1.4: Manage their appointment

UC1.5: Manage their profile

UC1.6: View the reports of their appointment visit

UC2: Admin User

UC2.1: Manage appointments booked by the patient users like approving, rejecting, finishing and cancelling the appointments of the users.

UC2.2: Manage the reports of the user's appointments like adding, updating and deleting.

UC2.3: Manage the doctors in the clinic like adding new doctors, updating and deleting the existing ones if the need arises.

UC2.4: Manage the services in the clinic like adding new services, updating and deleting the existing ones if the need arises.

UC2.5: Manage users and deleting them if needed.

UC3: Client

UC3.1: Requirement analysis and finalization

UC3.2: Check whether the requirements are met or not at the time of delivery

UC3.3: Provide funding, resources and further information if needed

UC4: Developer

UC4.1: Check for errors and fix the bugs

UC4.2: Add additional features

UC4.3: Provide progress to the client

UC4.4: Manage the databases

7.6.8.2.4 Operating Environment

OE1: Currently the dental appointment system web application is available in any web browser like Google Chrome, Microsoft Edge, Mozilla Firefox, etc. In the near future, the application will also run on other platforms such as Android and iOS.

7.6.8.2.5 Design and implementation constraints

CO1: Only registered users can book the appointment.

CO2: Reports can only be made of appointments that are marked as finished by the admin.

CO3: User can only view their appointment, report and profile.

CO4: User can view all the doctors and services.

CO5: User cannot access the admin's panel.

CO6: User can only cancel their pending and confirmed appointments.

CO7: Only admin can approve, reject and finish the user's appointments.

CO8: Only admin can manage the doctor, reports and services

CO8: The appointment can only be taken from 9am – 5pm any day of the week.

7.6.8.2.6 Assumptions and Dependencies

AS1: This application is target mainly for those who have to visit dental clinic for their dental check-ups.

AS2: The application will be connected with the database.

AS3: The documentation and reports of the system are only accessible to limited people like the client and developers.

AS4: The users can book any number of appointment they may like to. However, they may only choose one service at a time but can include other services at the same time by mentioning in the description.

7.6.8.3 Functional Requirement

7.6.8.3.1 Login Feature

Req.ID	Requirement Description		Priority	Complexity
FR1	All users will be able to access the login page		Most	High
	System Requirement			
	SR1	On opening the application, the user will be directed to the home/landing page.		
	SR2	When the user click the 'Login' button, they will be directed to the login page		
	SR3	The user needs to enter their login details i.e. username and password in order to log in to the application. If the credentials are correct, they will be redirected towards the user dashboard page.		

Table 67: Functional Requirement 1

7.6.8.3.2 Register Feature

Req.ID	Requirement Description		Priority	Complexity
FR2	All users will be able to access the register page		Most	High
	System Requirement			
SR4	When the user click the 'Register' button, they will be directed to the register page			
SR5	The user needs to fill the registration form. The details are then verified by the system and the user will be registered in to the system. The user can then log in using their username and password.			

Table 68: Functional Requirement 2

7.6.8.3.3 Booking an appointment and manage appointment

Req.ID	Requirement Description		Priority	Complexity							
FR3	All users who are registered and logged in will be able to use this feature.		Most	High							
	<table border="1"> <thead> <tr> <th colspan="2">System Requirement</th> </tr> </thead> <tbody> <tr> <td>SR6</td><td>After logging in, the users will be directed to their dashboard page.</td></tr> <tr> <td>SR7</td><td>When the user clicks the 'Book Appointment' button, they will be directed to the book appointment page.</td></tr> <tr> <td>SR8</td><td>The user needs to fill the form and click submit. If the details are valid, the appointment is then viewed by admin for approval.</td></tr> </tbody> </table>		System Requirement		SR6	After logging in, the users will be directed to their dashboard page.	SR7	When the user clicks the 'Book Appointment' button, they will be directed to the book appointment page.	SR8	The user needs to fill the form and click submit. If the details are valid, the appointment is then viewed by admin for approval.	
System Requirement											
SR6	After logging in, the users will be directed to their dashboard page.										
SR7	When the user clicks the 'Book Appointment' button, they will be directed to the book appointment page.										
SR8	The user needs to fill the form and click submit. If the details are valid, the appointment is then viewed by admin for approval.										
FR4	All users who have at least one appointment can use this feature.		Most	High							
	<table border="1"> <thead> <tr> <th colspan="2">System Requirement</th> </tr> </thead> <tbody> <tr> <td>SR9</td><td>When the user clicks the 'View Appointment' button, they will be directed to the view appointment page.</td></tr> <tr> <td>SR10</td><td>The user can cancel their pending or confirmed appointment by clicking the 'cancel' button.</td></tr> </tbody> </table>		System Requirement		SR9	When the user clicks the 'View Appointment' button, they will be directed to the view appointment page.	SR10	The user can cancel their pending or confirmed appointment by clicking the 'cancel' button.			
System Requirement											
SR9	When the user clicks the 'View Appointment' button, they will be directed to the view appointment page.										
SR10	The user can cancel their pending or confirmed appointment by clicking the 'cancel' button.										

Table 69: Functional Requirement 3

7.6.8.3.4 View Report

Req.ID	Requirement Description		Priority	Complexity								
FR5	<p>The user must have at least one appointment marked as finished to view the reports.</p> <table border="1" data-bbox="347 523 1082 1208"> <thead> <tr> <th colspan="2" data-bbox="347 523 1082 587">System Requirement</th></tr> </thead> <tbody> <tr> <td data-bbox="347 587 470 745">SR11</td><td data-bbox="470 587 1082 745">After logging in, the users will be directed to their dashboard page.</td></tr> <tr> <td data-bbox="347 745 470 946">SR12</td><td data-bbox="470 745 1082 946">When the user clicks the 'View Report' button, they will be directed to the view report page.</td></tr> <tr> <td data-bbox="347 946 470 1208">SR13</td><td data-bbox="470 946 1082 1208">The user will be able to view the report details of the appointment that they have finished. The admin upload the report and the user can view it.</td></tr> </tbody> </table>	System Requirement		SR11	After logging in, the users will be directed to their dashboard page.	SR12	When the user clicks the 'View Report' button, they will be directed to the view report page.	SR13	The user will be able to view the report details of the appointment that they have finished. The admin upload the report and the user can view it.	Most		High
System Requirement												
SR11	After logging in, the users will be directed to their dashboard page.											
SR12	When the user clicks the 'View Report' button, they will be directed to the view report page.											
SR13	The user will be able to view the report details of the appointment that they have finished. The admin upload the report and the user can view it.											

Table 70: Functional Requirement 4

7.6.8.3.5 View Doctor

Req.ID	Requirement Description		Priority	Complexity								
FR6	<p>The user must be registered and logged in to view this feature.</p> <table border="1" data-bbox="347 523 1082 1072"> <thead> <tr> <th colspan="2" data-bbox="347 523 1082 587">System Requirement</th></tr> </thead> <tbody> <tr> <td data-bbox="347 587 470 745">SR14</td><td data-bbox="470 587 1082 745">After logging in, the users will be directed to their dashboard page.</td></tr> <tr> <td data-bbox="347 745 470 946">SR15</td><td data-bbox="470 745 1082 946">When the user clicks the 'View Doctor' button, they will be directed to the view doctor page.</td></tr> <tr> <td data-bbox="347 946 470 1106">SR16</td><td data-bbox="470 946 1082 1106">The user will be able to view the doctors along with their details.</td></tr> </tbody> </table>	System Requirement		SR14	After logging in, the users will be directed to their dashboard page.	SR15	When the user clicks the 'View Doctor' button, they will be directed to the view doctor page.	SR16	The user will be able to view the doctors along with their details.	Most		Normal
System Requirement												
SR14	After logging in, the users will be directed to their dashboard page.											
SR15	When the user clicks the 'View Doctor' button, they will be directed to the view doctor page.											
SR16	The user will be able to view the doctors along with their details.											

Table 71: Functional Requirement 5

7.6.8.3.6 View Services

Req.ID	Requirement Description		Priority	Complexity								
FR7	<p>The user must be registered and logged in to view this feature.</p> <table border="1" data-bbox="347 523 1082 1108"> <thead> <tr> <th colspan="2" data-bbox="347 523 1082 593">System Requirement</th></tr> </thead> <tbody> <tr> <td data-bbox="347 593 470 741">SR17</td><td data-bbox="470 593 1082 741">After logging in, the users will be directed to their dashboard page.</td></tr> <tr> <td data-bbox="347 741 470 931">SR18</td><td data-bbox="470 741 1082 931">When the user clicks the 'View Services' button, they will be directed to the view services page.</td></tr> <tr> <td data-bbox="347 931 470 1108">SR19</td><td data-bbox="470 931 1082 1108">The user will be able to view the services along with their details.</td></tr> </tbody> </table>	System Requirement		SR17	After logging in, the users will be directed to their dashboard page.	SR18	When the user clicks the 'View Services' button, they will be directed to the view services page.	SR19	The user will be able to view the services along with their details.	Most		Normal
System Requirement												
SR17	After logging in, the users will be directed to their dashboard page.											
SR18	When the user clicks the 'View Services' button, they will be directed to the view services page.											
SR19	The user will be able to view the services along with their details.											

Table 72: Functional Requirement 6

7.6.8.3.7 Manage Profile

Req.ID	Requirement Description		Priority	Complexity
FR8	The user must be registered and logged in to view this feature.		Most	Medium
	System Requirement			
	SR20	After logging in, the users will be directed to their dashboard page.		
	SR21	When the user clicks the 'View Profile' button, they will be directed to the view profile page.		
	SR22	The user will be able to view their details.		
FR9	The user must be registered and logged in to view this feature.		Most	High
	System Requirement			
	SR23	When the user clicks the 'Edit Profile' button, they will be directed to the view services page.		
	SR24	When the user clicks the 'View Profile' button, they will be directed to the edit profile page.		
	SR25	The user will be able to update their details and if the details are valid, the details of the users will be updated.		

Table 73: Functional Requirement 7

7.6.8.4 External Interface Requirement

7.6.8.4.1 User Interfaces

The project's user interface will be intended to be clean and simple to use. Simple phrases will be used, and page titles will be assigned based on the objective of the page. The application will make use of appropriate colour schemes and fonts. Similarly, the photos in the program will be retained in order to deliver a relevant message to the users. User testing will be done on a regular basis to ensure maximum effort and happiness.

The home page was made as a one page website. The screenshot of the home page UI is shown below:

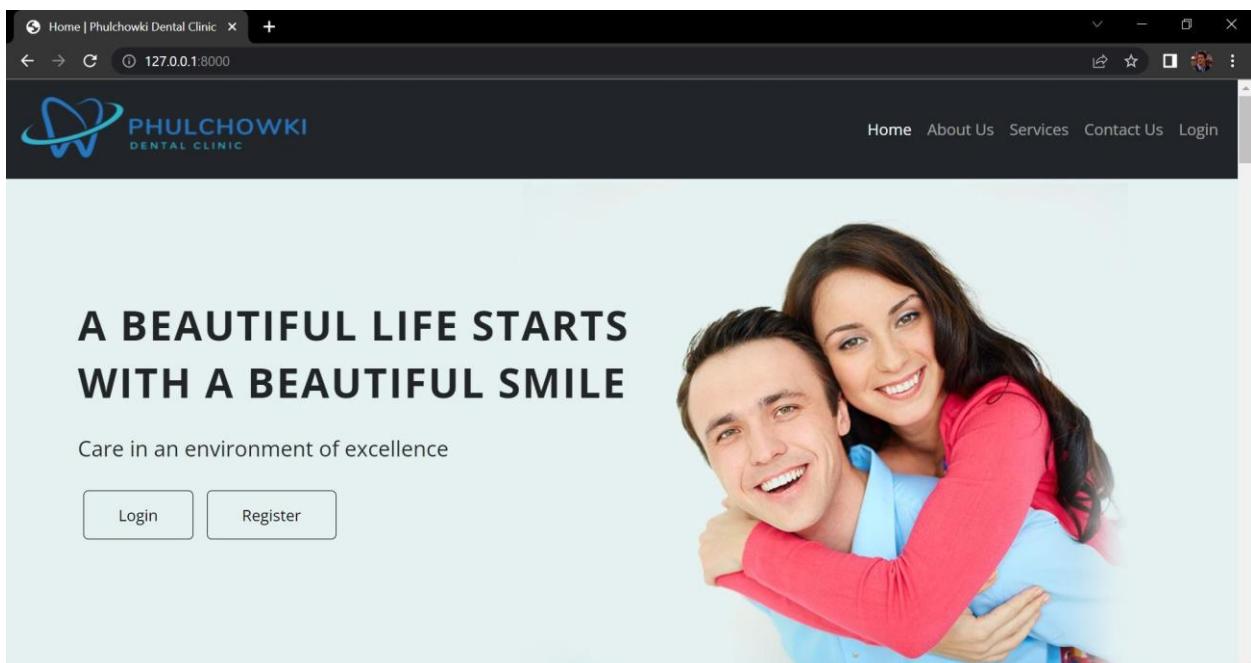


Figure 399: UI of Home Page 1

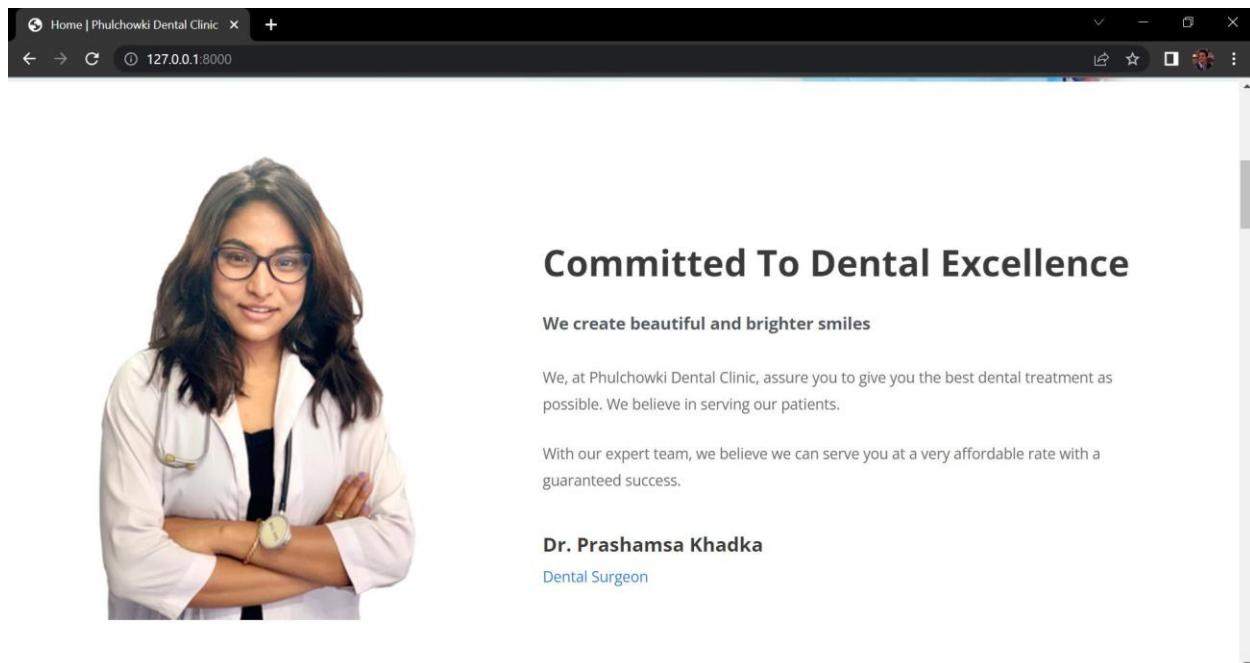


Figure 400: UI of Home Page 2

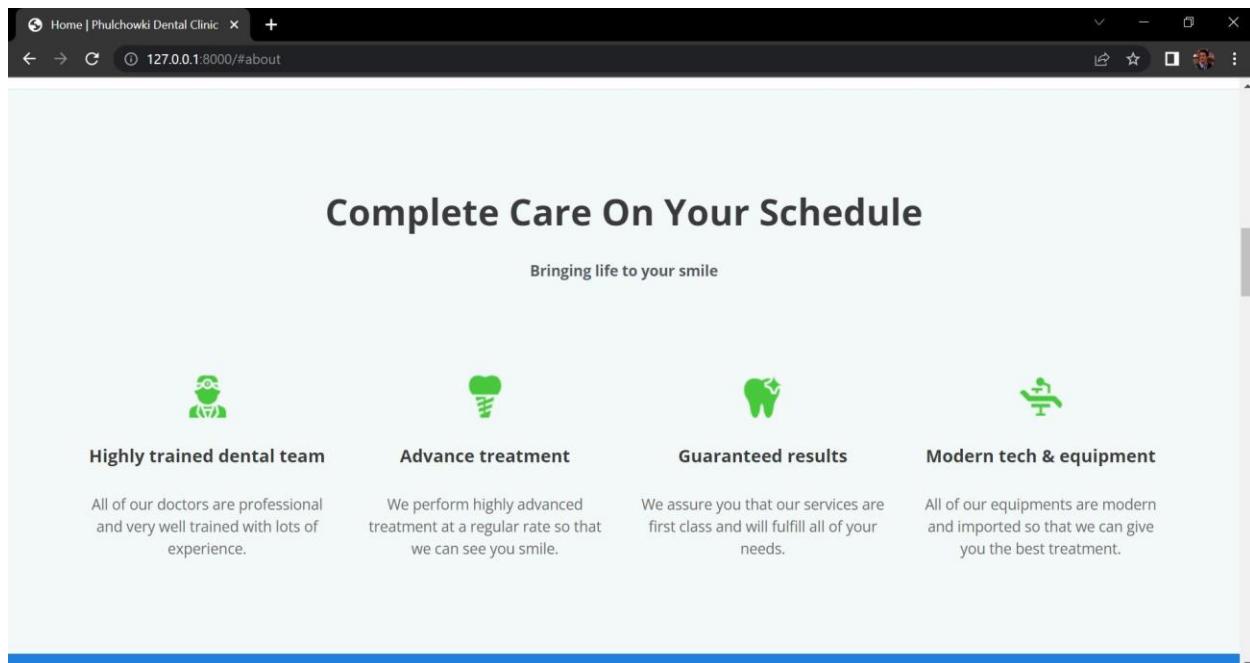


Figure 401: UI of Home Page 3

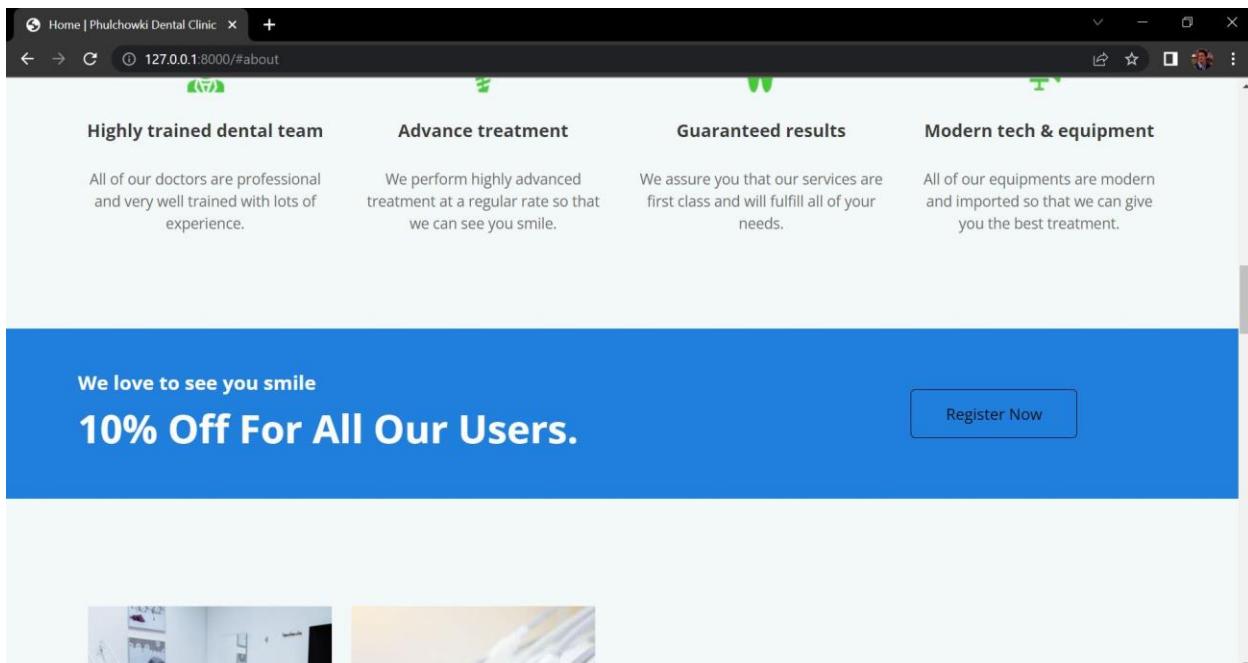


Figure 402: UI of Home Page 4

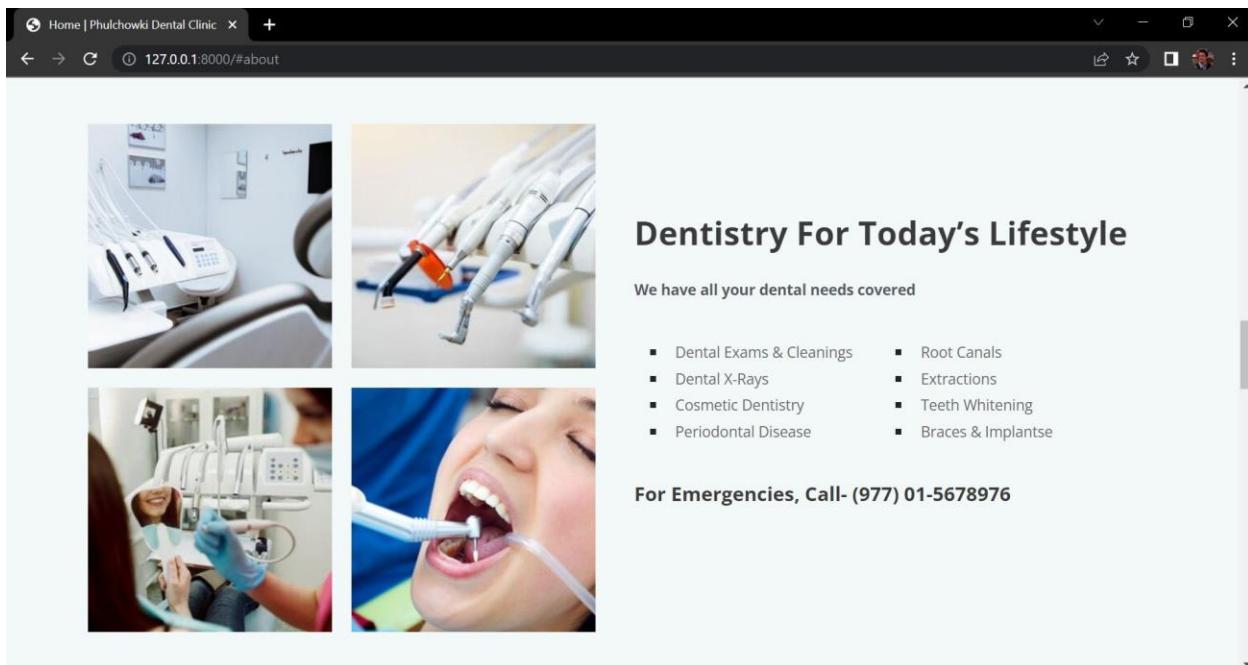


Figure 403: UI of Home Page 5

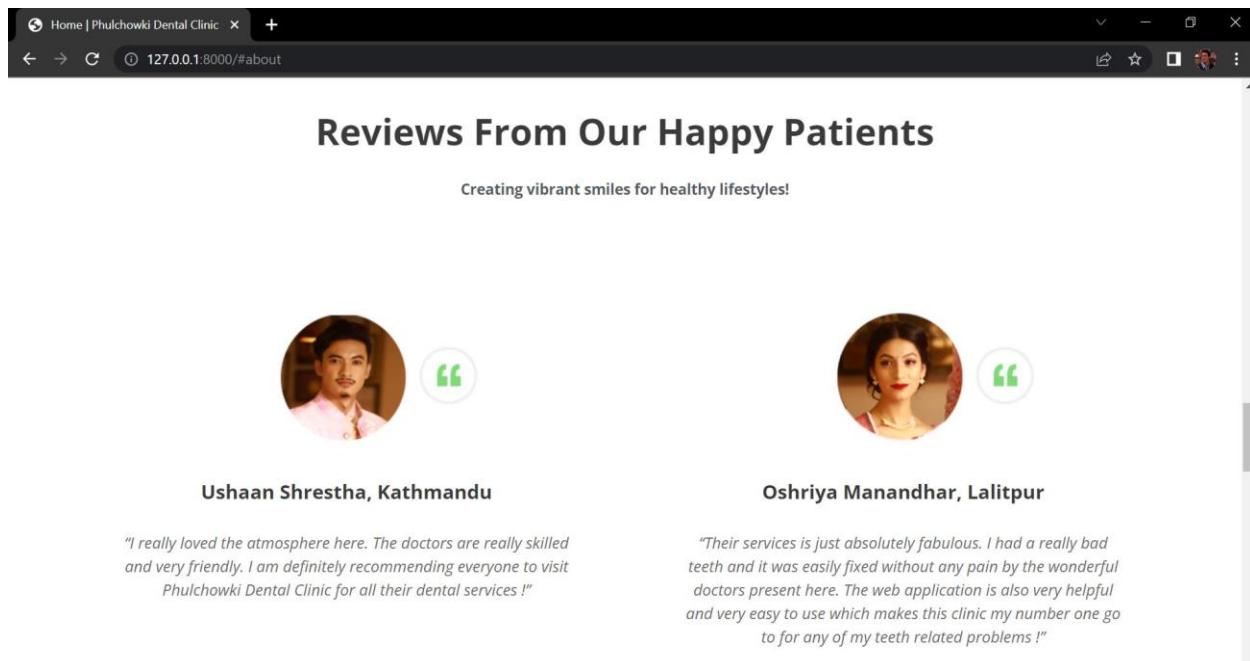


Figure 404: UI of Home Page 6

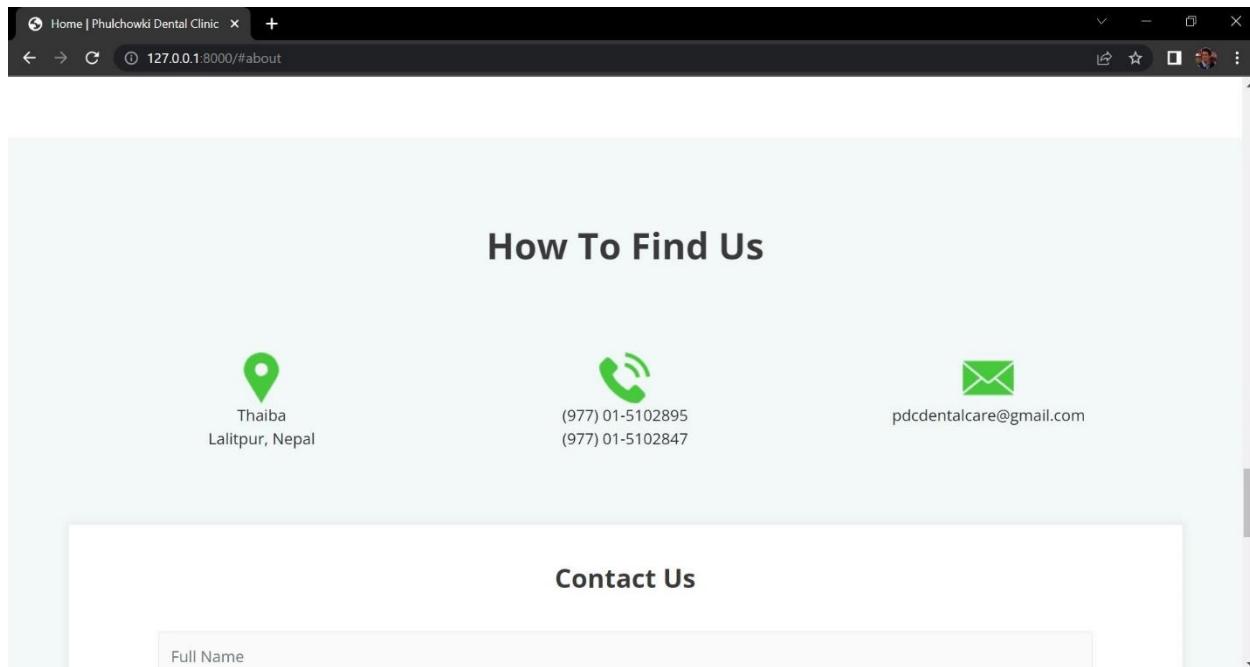


Figure 405: UI of Home Page 7

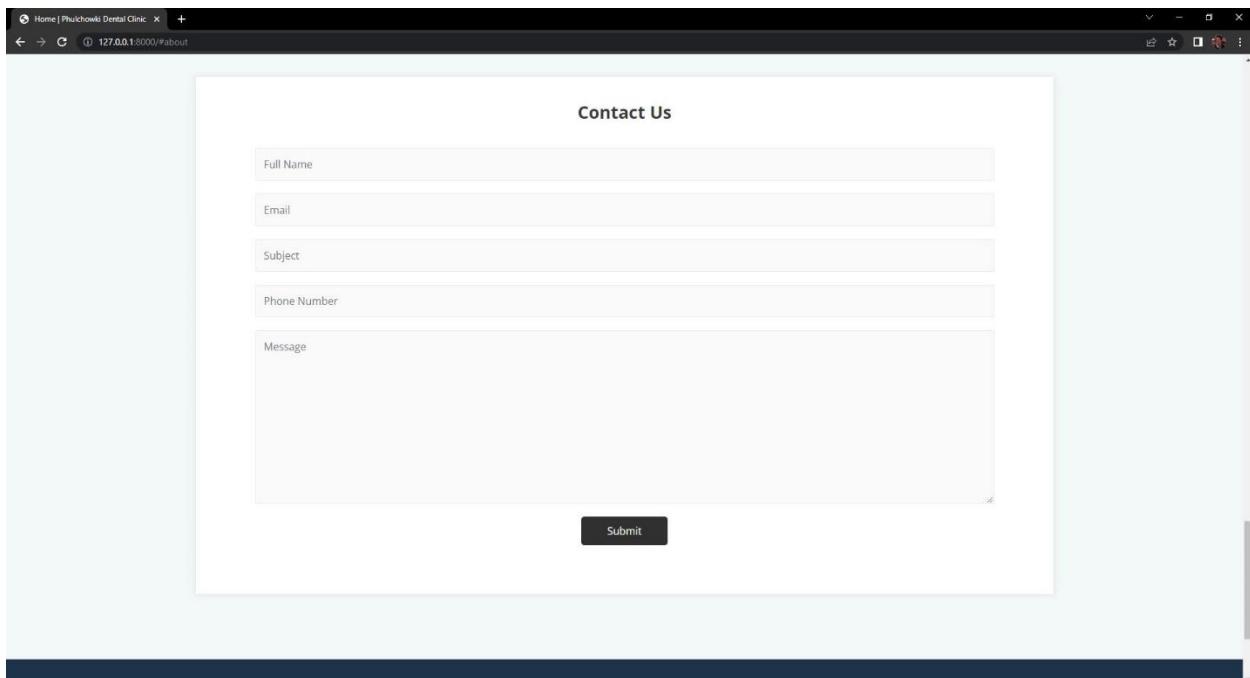


Figure 406: UI of Home Page 8

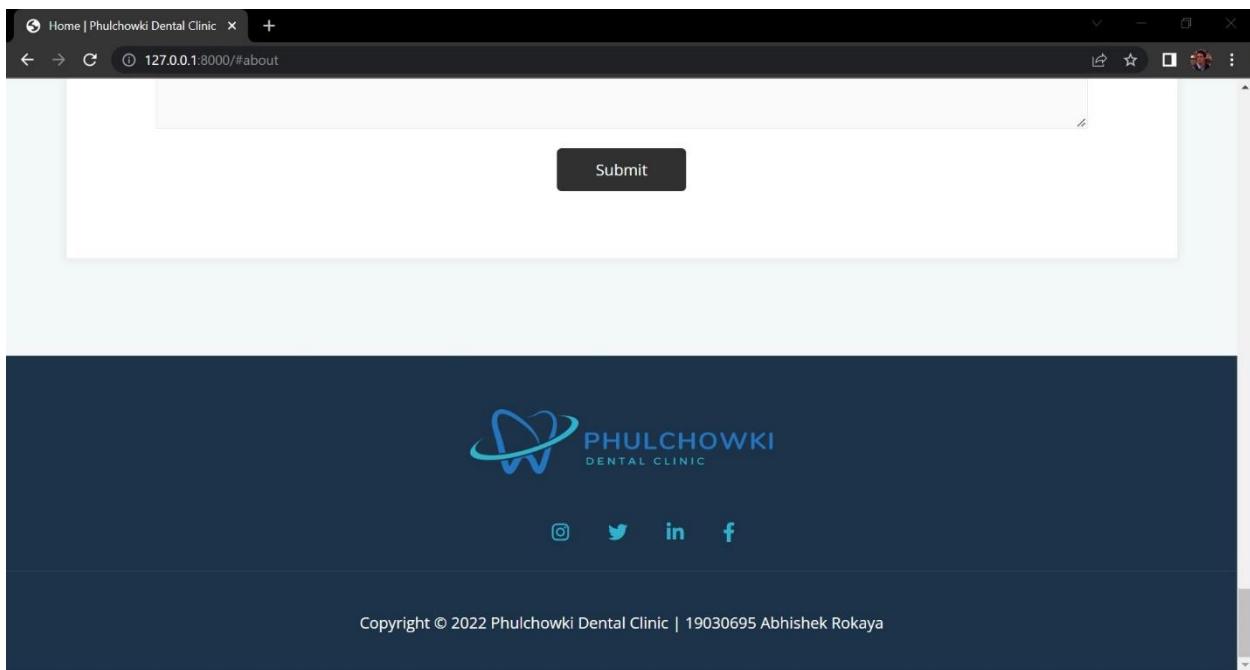


Figure 407: UI of Home Page 9

[Click here to view the rest of the UI.](#)

7.6.8.4.2 Hardware Interfaces

Since this is a web application, there are no such hardware requirements. This application can be executed on any functional machine with an internet connection.

7.6.8.4.3 Software Interfaces

The software interface for the application is as mentioned below:

- **Programming Language:** Python
- **Framework:** Django
- **IDE:** Visual Studio Code
- **Database:** SQLite3

7.6.8.4.4 Communication Interfaces

This system's communication interface will use the HTTP protocol for internet communication and the TCP/IP protocol suite for intranet communication.

7.6.8.5 Non-Functional Requirements

Non-functional requirements are those that are used to assess the application's quality. This section contains a comprehensive description of the application's non-functional needs.

7.4.5.5.1 Performance Requirements

Req. ID	Requirement Description	Priority	Complexity
PR1	The system should be fast and easy to use.	Should	Medium
PR2	The system should be reliable meaning the function should work properly.	Should	Normal
PR3	The system should be user friendly.	Should	Normal

Table 74: Performance Requirement

7.6.8.5.2 Safety Requirements

Req. ID	Requirement Description	Priority	Complexity
SR1	The program shall not install any type of malware or spyware in the background without notifying the users.	Should	Medium
SR2	The system must keep track of the session to guarantee that the login and logout are secure.	Should	High
SR3	A user category must not have access to the profiles of other user categories, i.e. a user (patient) must not be able to access the admin panel.	Should	Medium

Table 75: Safety Requirement

7.6.8.5.3 Other Software Quality Attributes

Req. ID	Requirement Description	Priority	Complexity
SQA1	The system must be scalable, which means that it should be able to incorporate new features in the future.	Could	Medium
SQA2	All users must be able to quickly access the program.	Should	Normal
SQA3	The system must be user-friendly, and even non-technical users should be able to utilize it with ease.	Should	Normal
SQA4	The program must be interoperable, which means it must act consistently across browsers.	Could	Normal

Table 76: Other Software Quality Attributes

7.6.9 Wireframes

A wireframe is a schematic or blueprint that is useful for helping you, your programmers and designers think and communicate about the structure of the software or website you're building. (Guilizzoni, 2022) The wireframes are created using Balsamiq. It is a desktop application that is used to create various kinds of wireframes for different web as well as mobile based applications. Wireframes are needed in a software development. It creates a blueprint for the application we are developing. It helps in the development phase to create a design by looking at the wireframes.

The wireframes for the dental appointment system application are as follows:

1. Home Page (Landing Page)

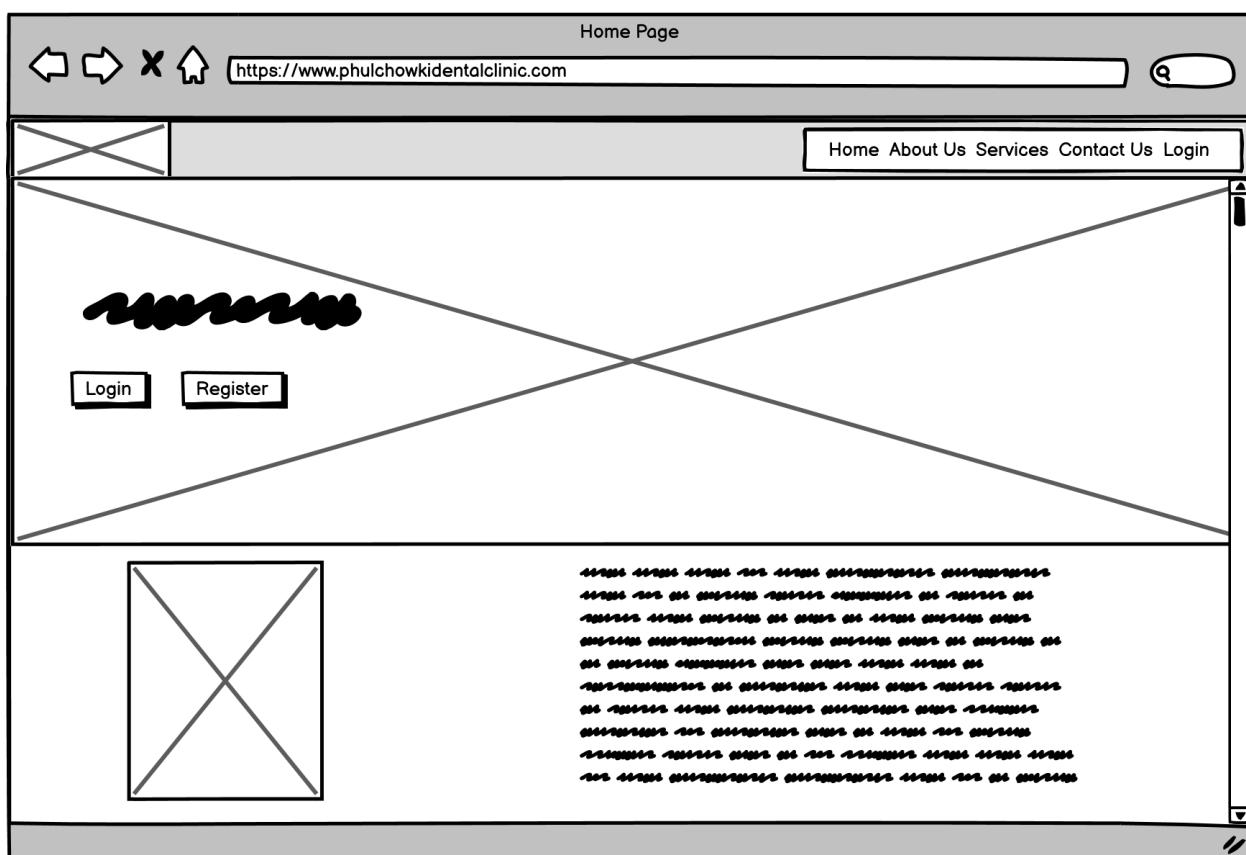


Figure 408: Wireframe - Home Page

2. User Registration Page

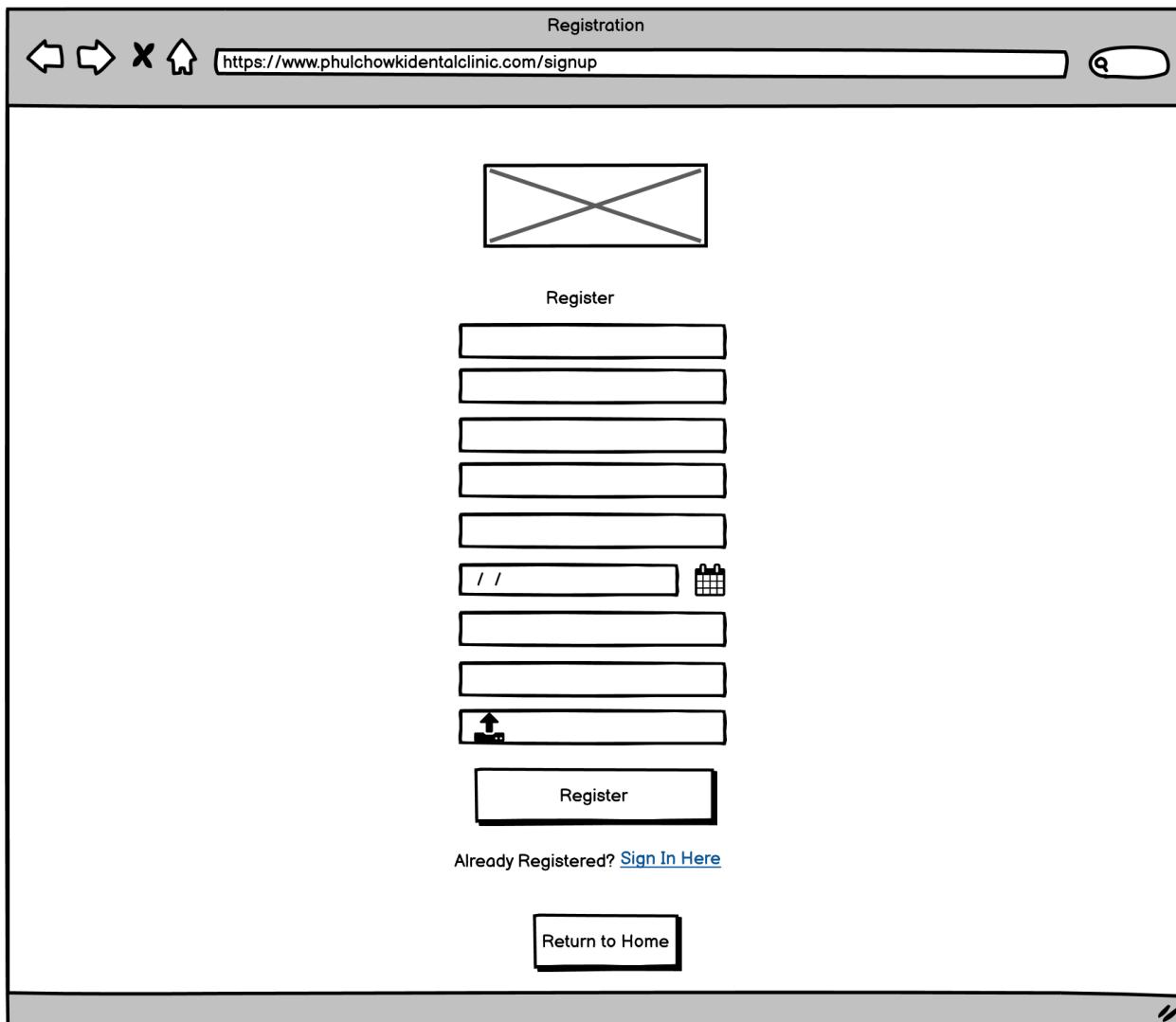


Figure 409: Wireframe - User Registration

3. User Login Page

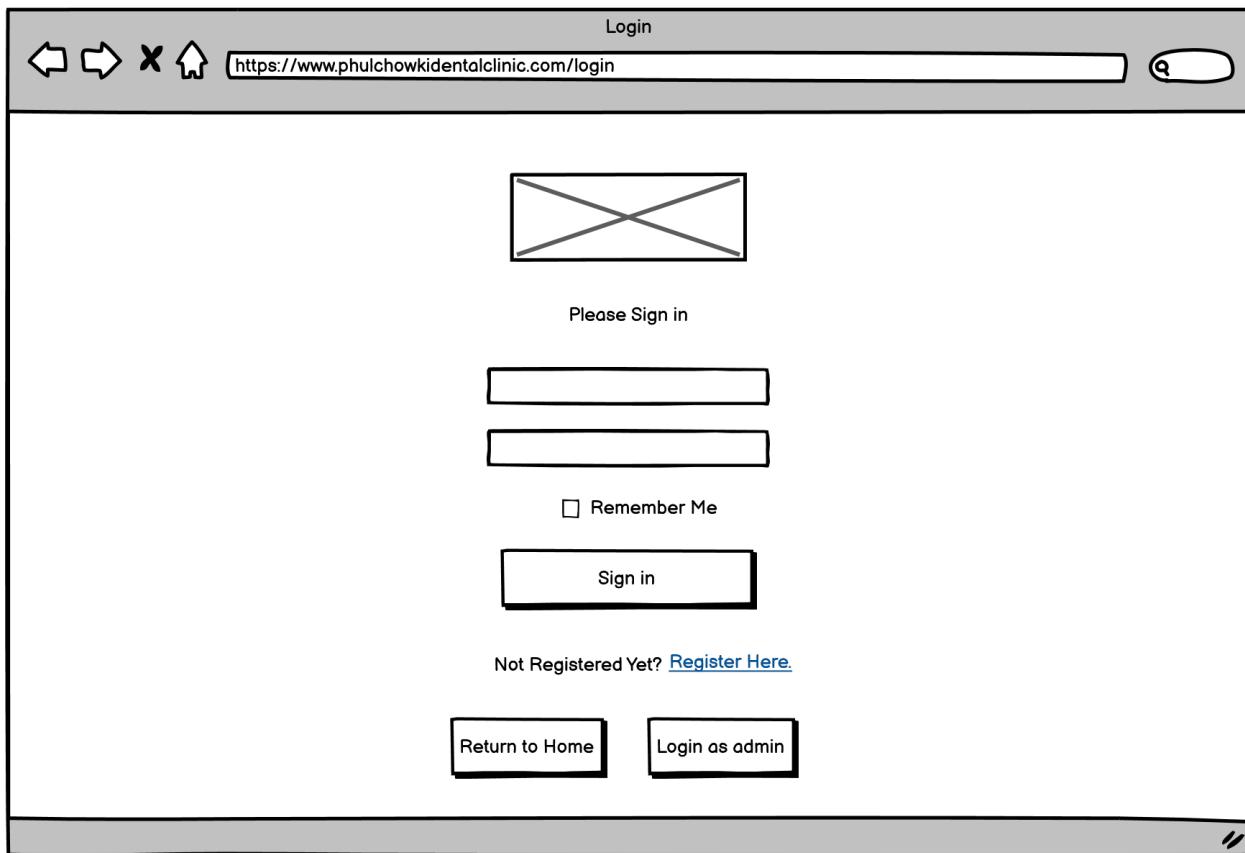


Figure 410: Wireframe - User Login

4. Admin Login Page

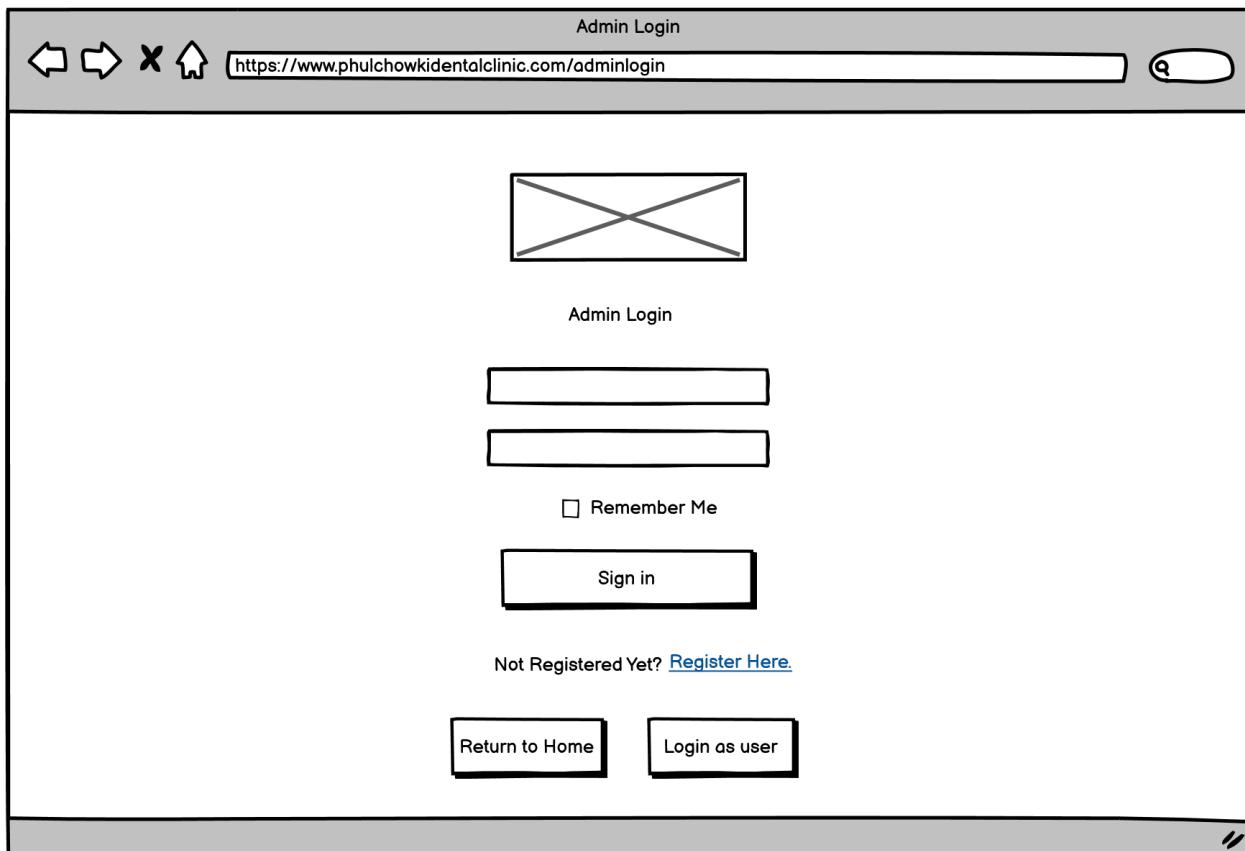


Figure 411: Wireframe - Admin Login

5. User Dashboard Page

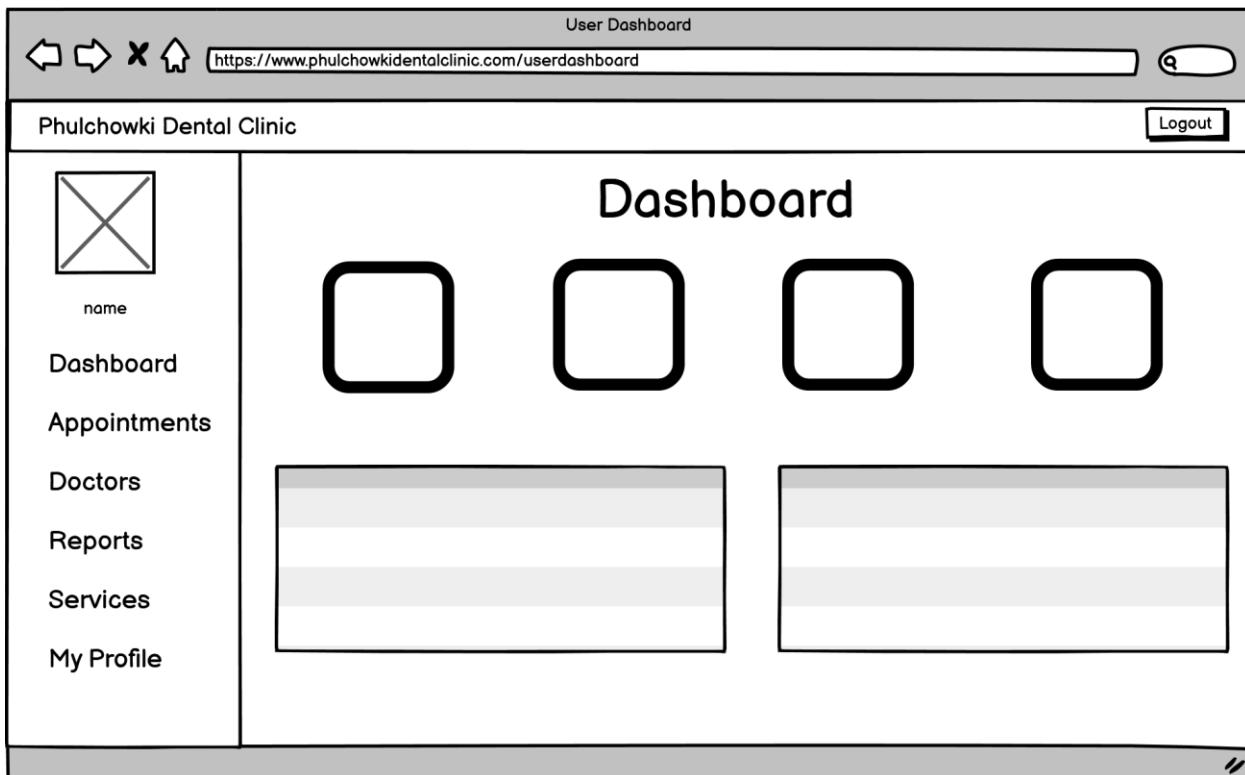


Figure 412: Wireframe - User Dashboard

6. User View Appointment Page

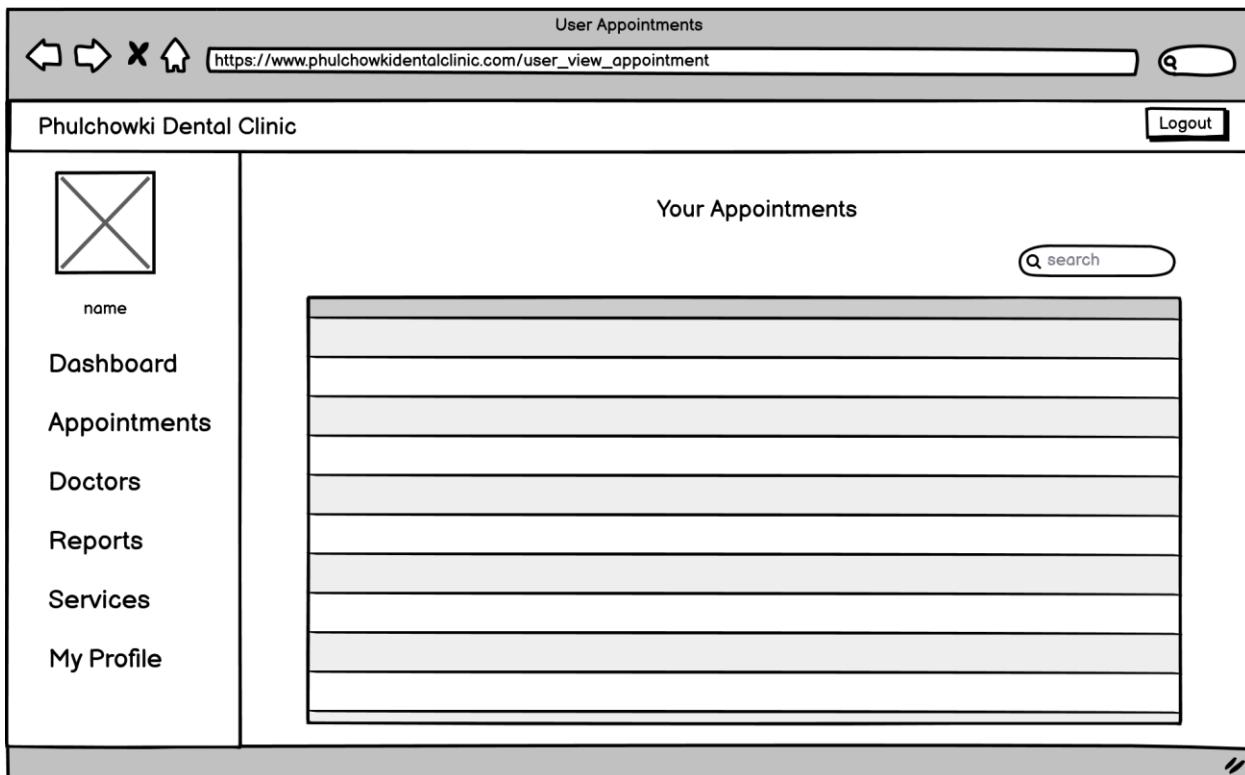


Figure 413: Wireframe - User View Appointment

7. User Book Appointment Page

The wireframe shows a web browser window titled "User Book Appointment" with the URL "https://www.phulchowkidentalclinic.com/user_book_appointment". The page header includes standard browser controls (back, forward, search, etc.) and the clinic's name "Phulchowki Dental Clinic" with a "Logout" button. On the left, a sidebar lists navigation links: Dashboard, Appointments, Doctors, Reports, Services, and My Profile. The main content area is titled "Book an Appointment" and contains several input fields: two dropdown menus labeled "ComboBox", a date input field with a calendar icon containing the text "/ /", and two large empty rectangular boxes for additional information. A "Submit" button is located at the bottom right of the form area.

Figure 414: Wireframe - User Book Appointment

8. User View Doctor Page

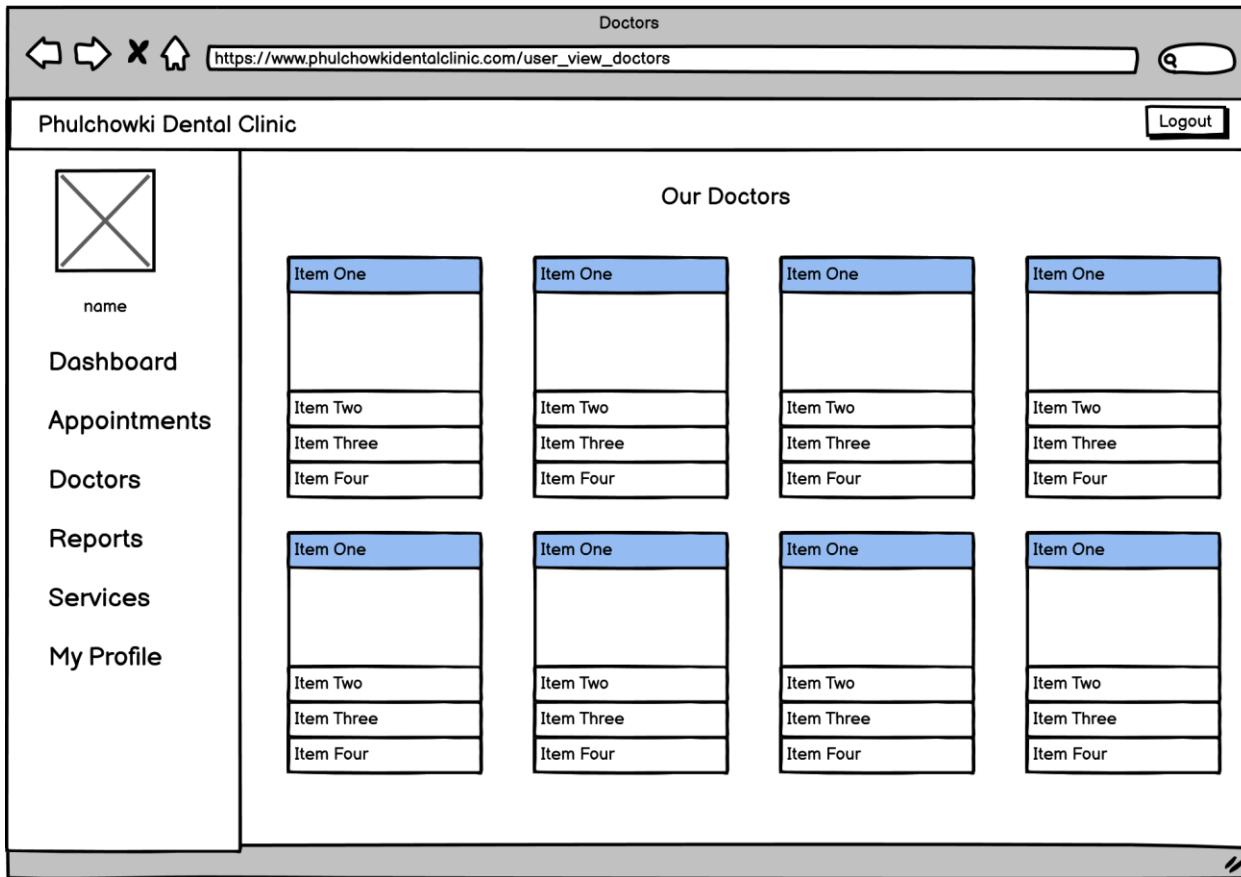


Figure 415: Wireframe - User View Doctor

9. User View Report Page

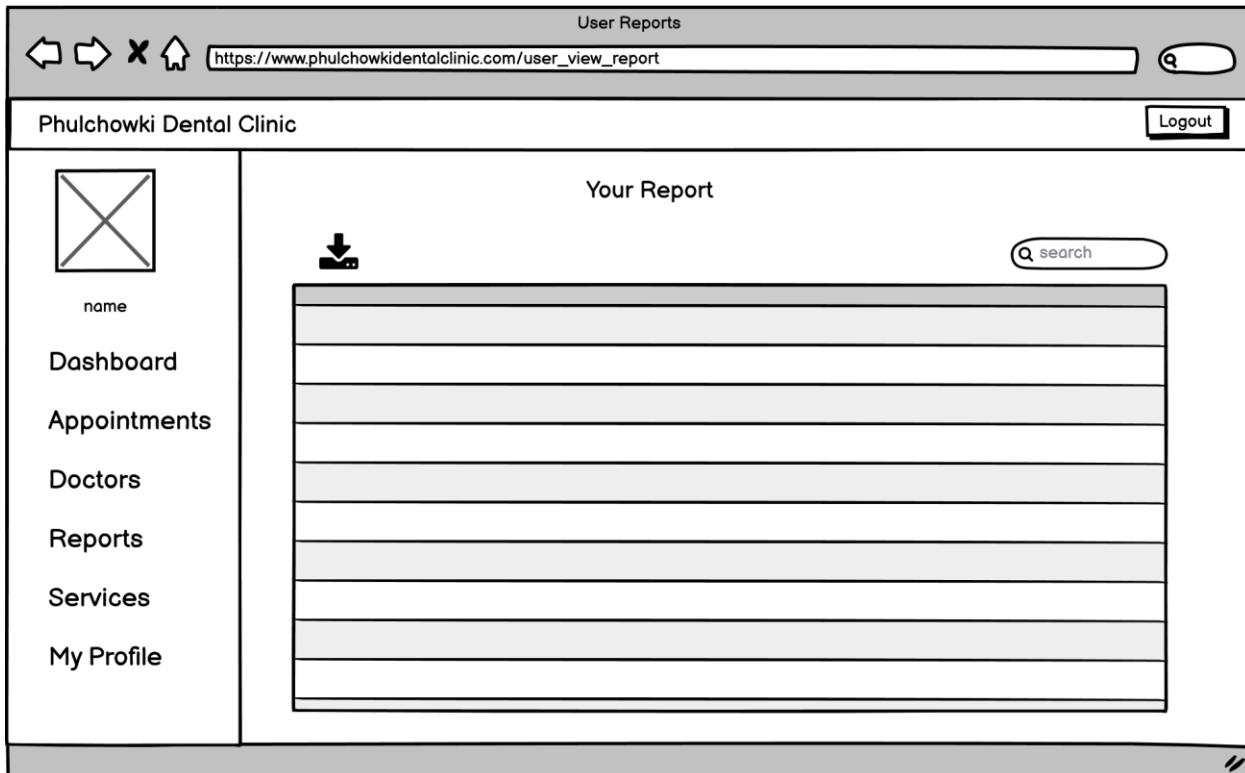


Figure 416: Wireframe - User View Report

10. User View Service Page

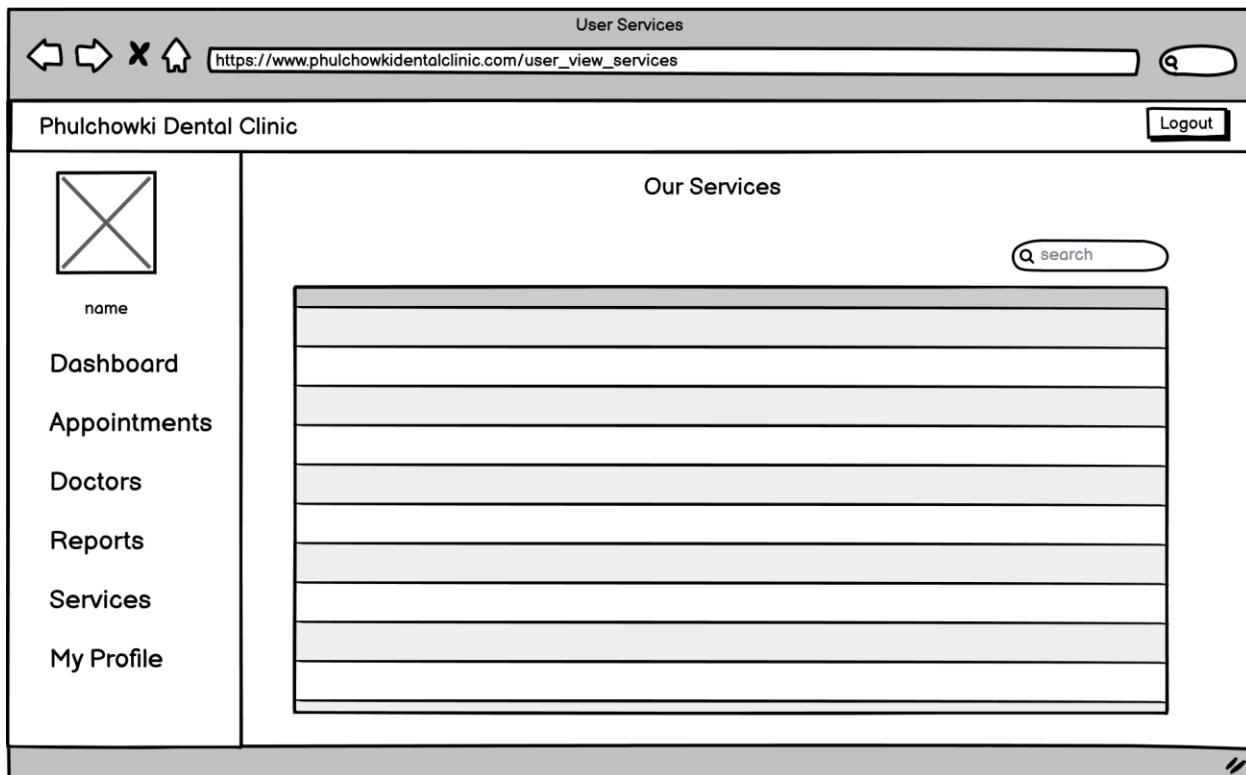


Figure 417: Wireframe - User View Service

11. User View Profile Page

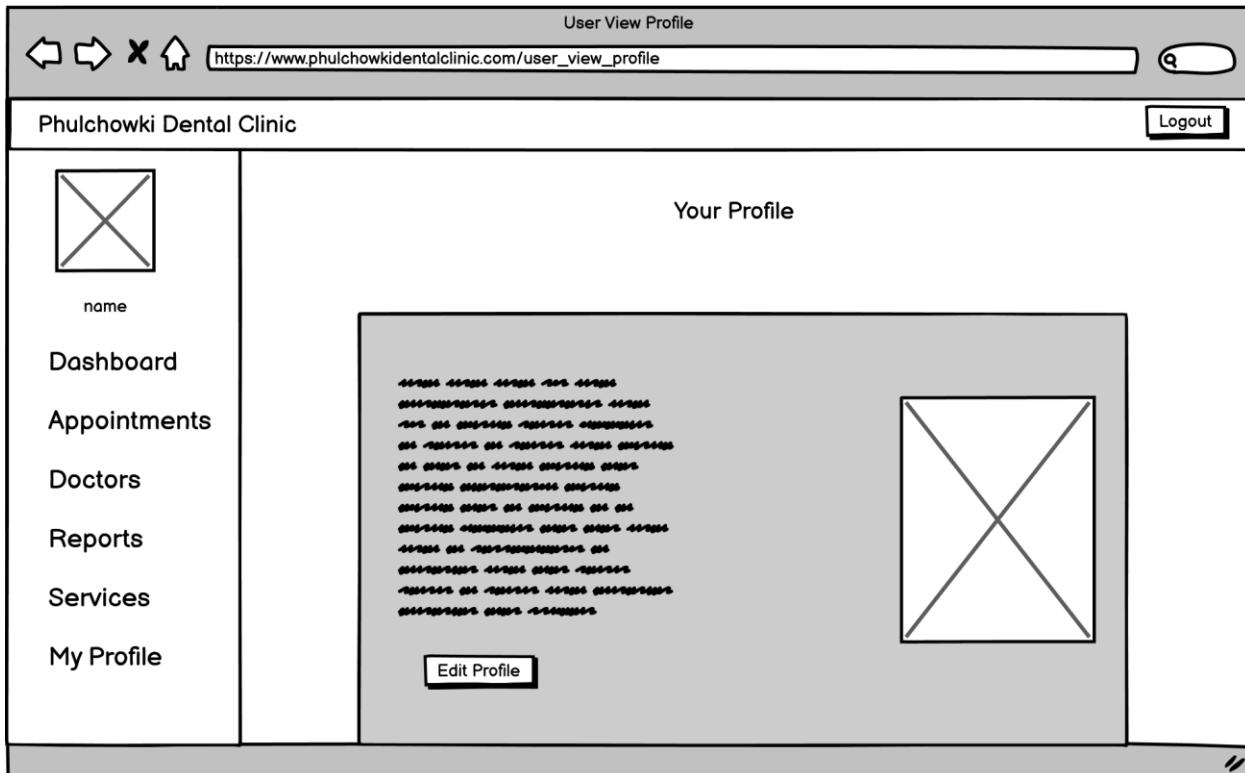


Figure 418: Wireframe - User View Profile

12. User Edit Profile Page

The wireframe shows a web browser window titled "User Edit Profile" with the URL "https://www.phulchowkidentalclinic.com/user_edit_profile". The page header includes standard navigation icons (back, forward, search, etc.) and the clinic's name "Phulchowki Dental Clinic" on the left, with a "Logout" button on the right. The main content area is titled "Update Your Profile" and contains seven input fields for profile information. To the left of these fields is a sidebar with links: "Dashboard", "Appointments", "Doctors", "Reports", "Services", and "My Profile". An icon of a person with a large X over it is positioned above the "name" link. A central "Update" button is located at the bottom of the form.

Figure 419: Wireframe - User Edit Profile

13. Admin Dashboard Page

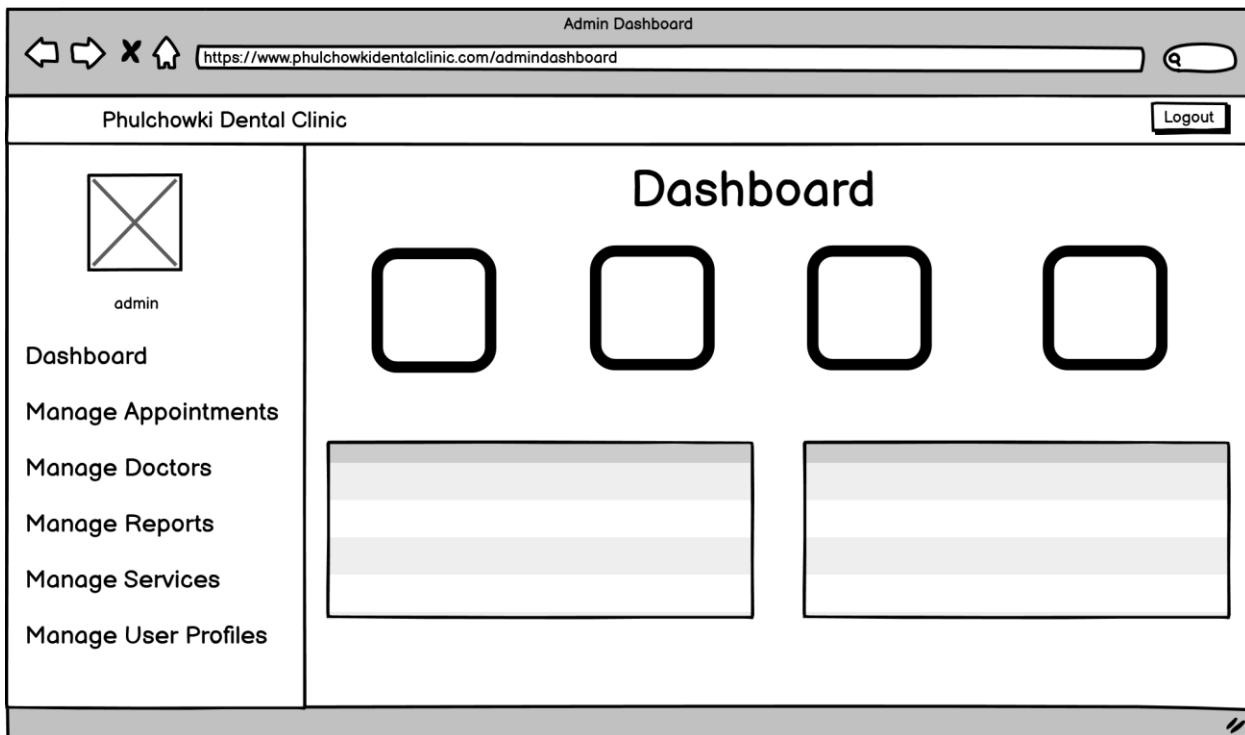


Figure 420: Wireframe - Admin Dashboard

14. Admin View Appointment Page

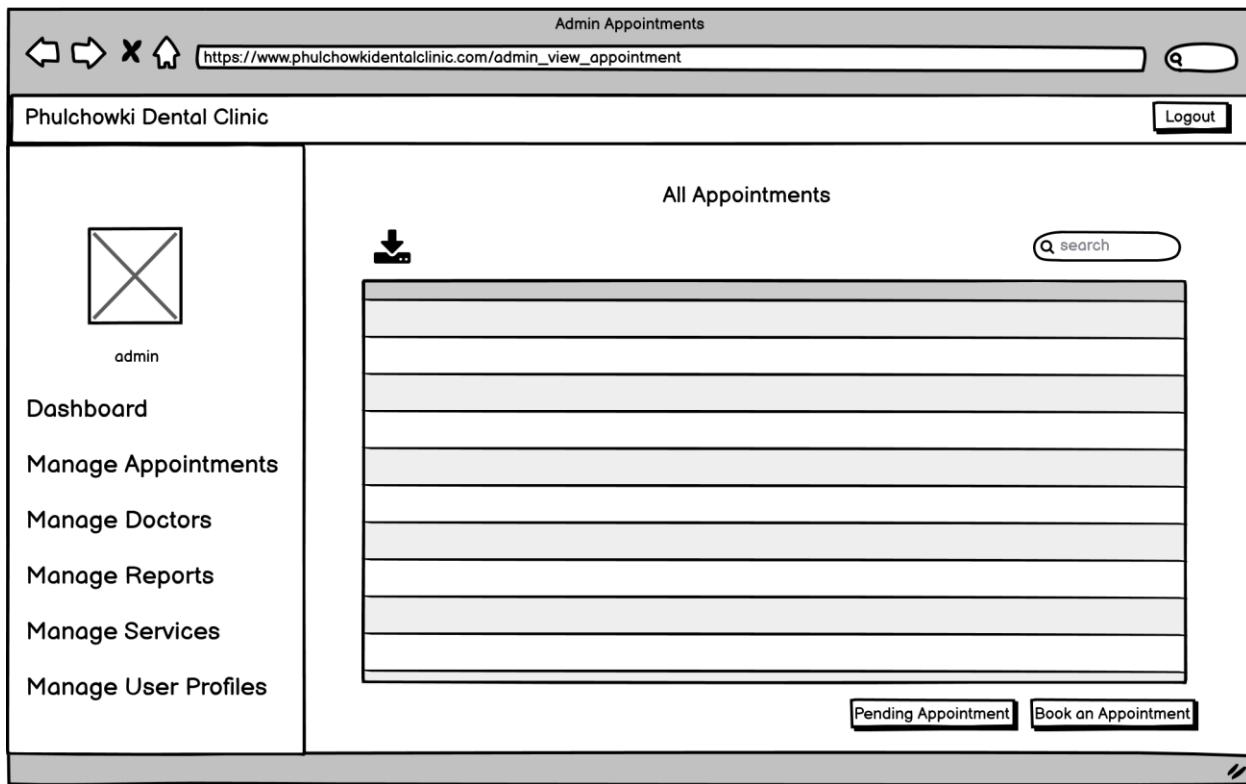


Figure 421: Wireframe - Admin View Appointment

15. Admin Approve Appointment Page

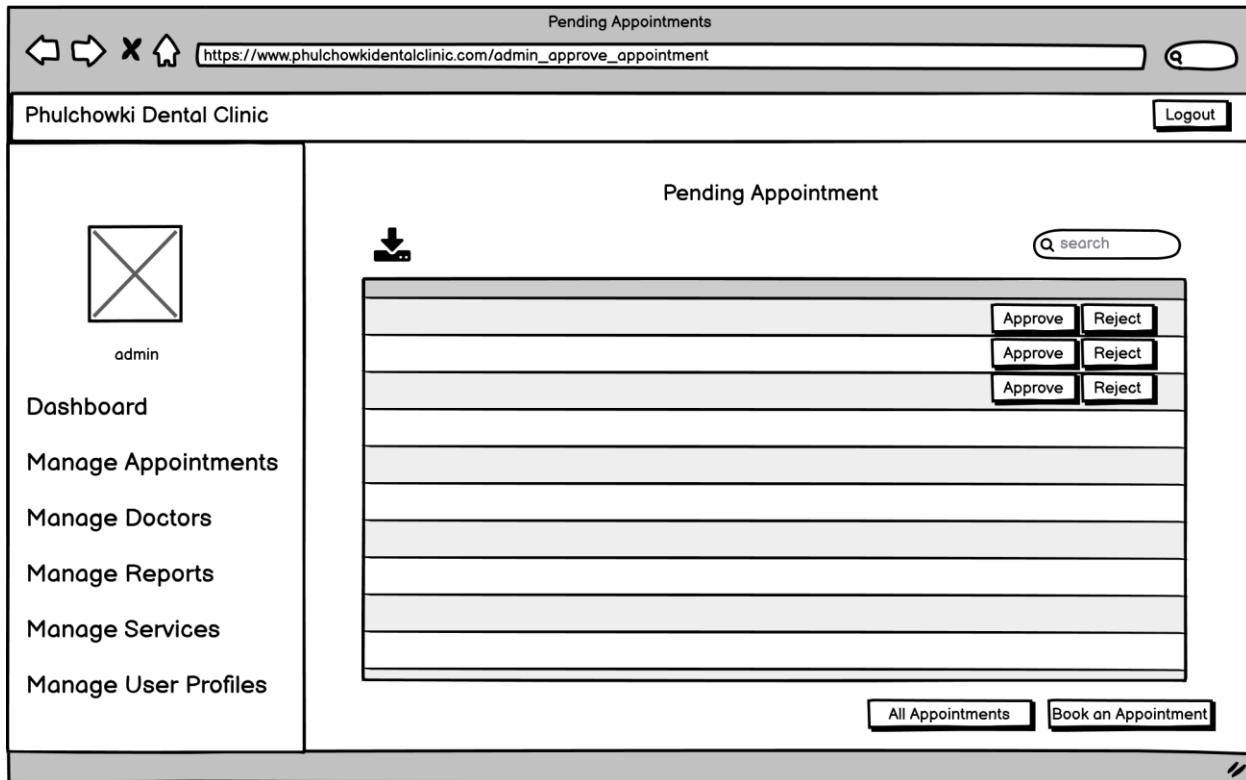
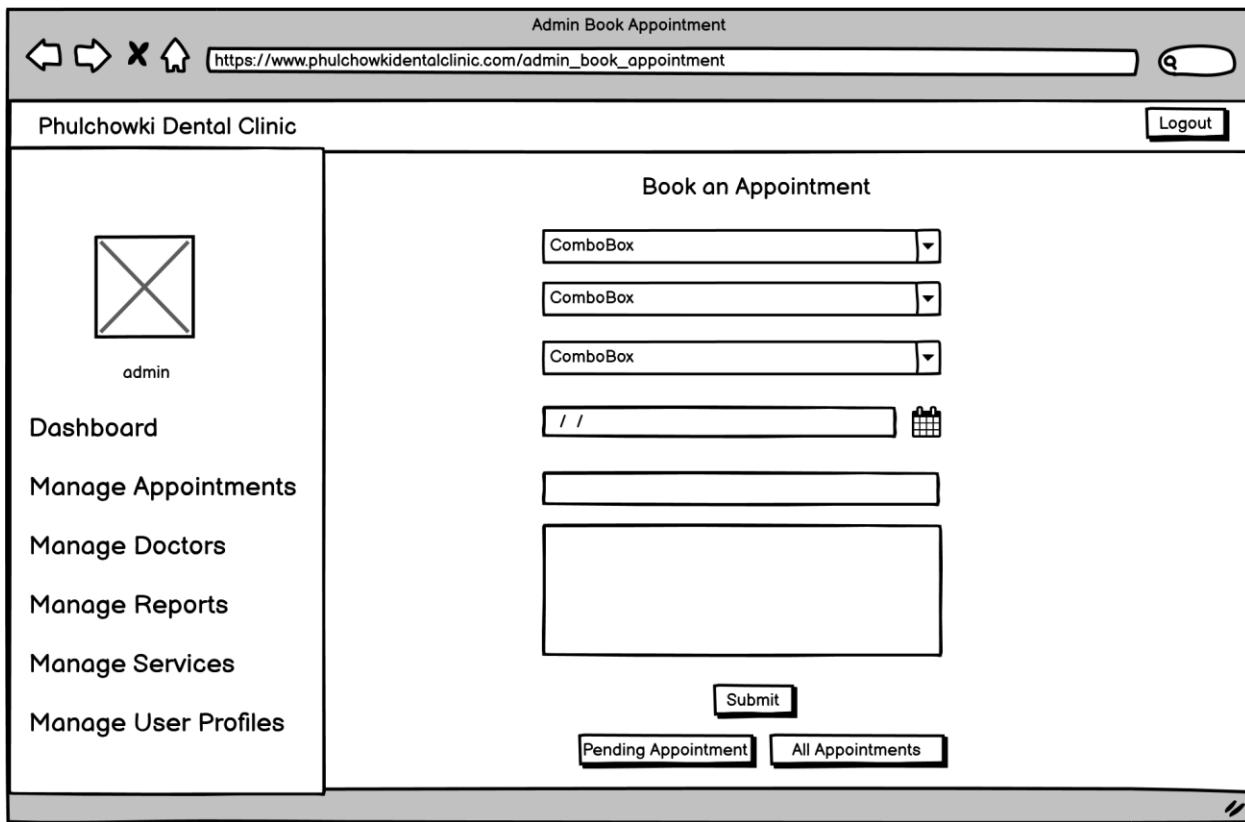


Figure 422: Wireframe - Admin Approve Appointment

16. Admin Book Appointment Page



The wireframe illustrates the 'Admin Book Appointment' page for Phulchowki Dental Clinic. The page is titled 'Admin Book Appointment' at the top left. A navigation bar at the top includes standard browser icons (back, forward, search, etc.) and a URL bar showing 'https://www.phulchowkidentalclinic.com/admin_book_appointment'. On the right side of the header is a 'Logout' button.

The main content area is titled 'Book an Appointment'. It features a sidebar on the left containing a user icon labeled 'admin' and a list of administrative functions: Dashboard, Manage Appointments, Manage Doctors, Manage Reports, Manage Services, and Manage User Profiles.

The central form area contains several input fields:

- Three dropdown menus (ComboBox) stacked vertically.
- A date input field with a calendar icon.
- A large text input field.
- A large text area (text box).

At the bottom of the form are three buttons: 'Submit' (highlighted in black), 'Pending Appointment', and 'All Appointments'.

Figure 423: Wireframe - Admin Book Appointment

17. Admin View Doctor Page

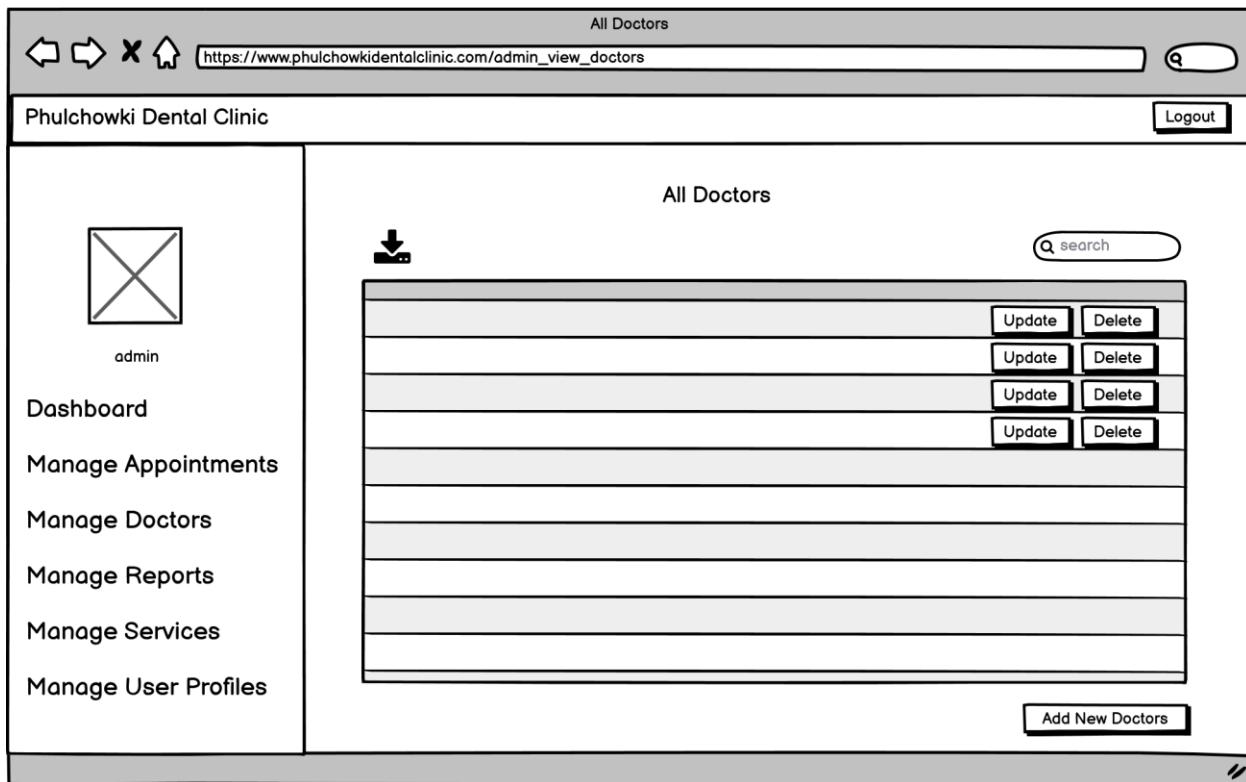


Figure 424: Wireframe - Admin View Doctor

18. Admin Add Doctor Page

The wireframe shows the 'Admin Add Doctors' page. At the top, there are standard browser navigation icons (back, forward, stop, home) and a search bar with the URL https://www.phulchowkidentalclinic.com/admin_add_doctors. Below the header, the clinic's name 'Phulchowki Dental Clinic' is displayed next to a user icon (a square with an X) and a 'Logout' button. The left sidebar contains links for 'Dashboard', 'Manage Appointments', 'Manage Doctors', 'Manage Reports', 'Manage Services', and 'Manage User Profiles'. The main content area is titled 'Add Doctors' and contains several input fields: a first name field, a last name field, a date of birth field with a calendar icon, and five other empty input fields for middle names, suffixes, gender, address, and phone number. At the bottom are a 'Submit' button and a 'View All Doctors' link.

Figure 425: Wireframe - Admin Add Doctor

19. Admin View Report Page

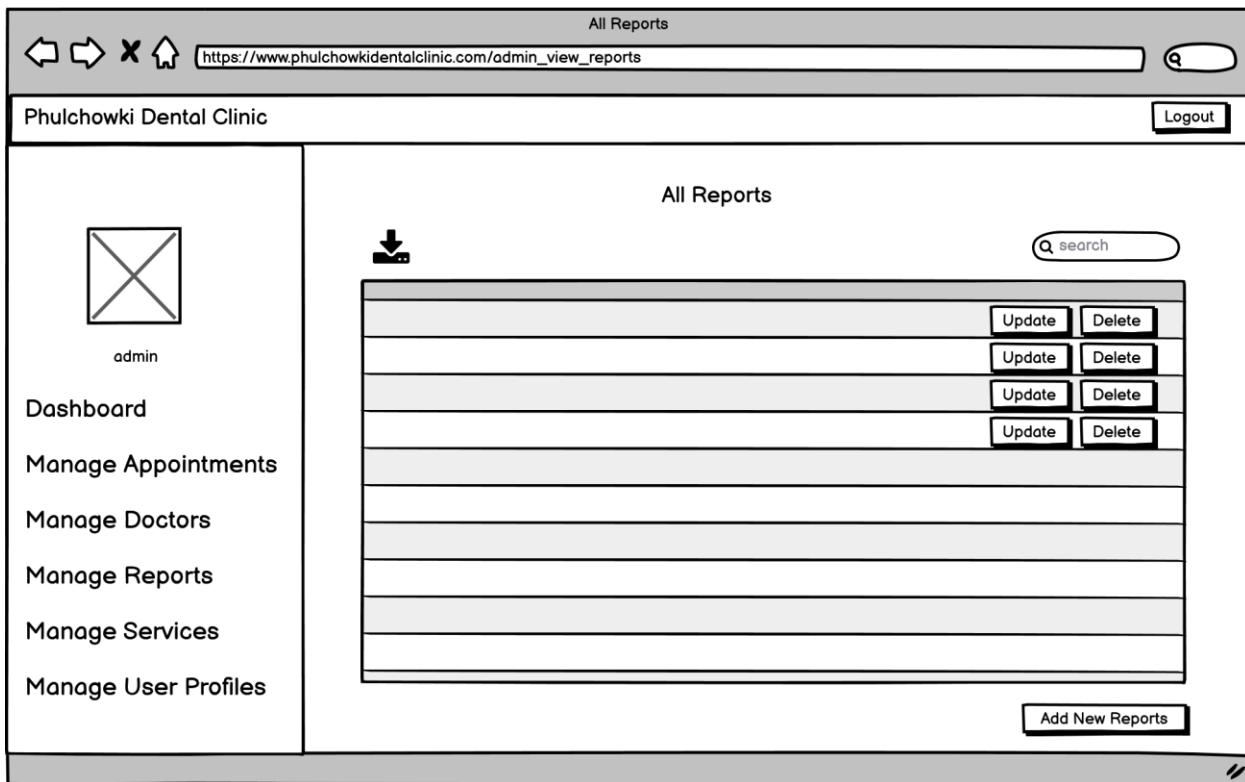
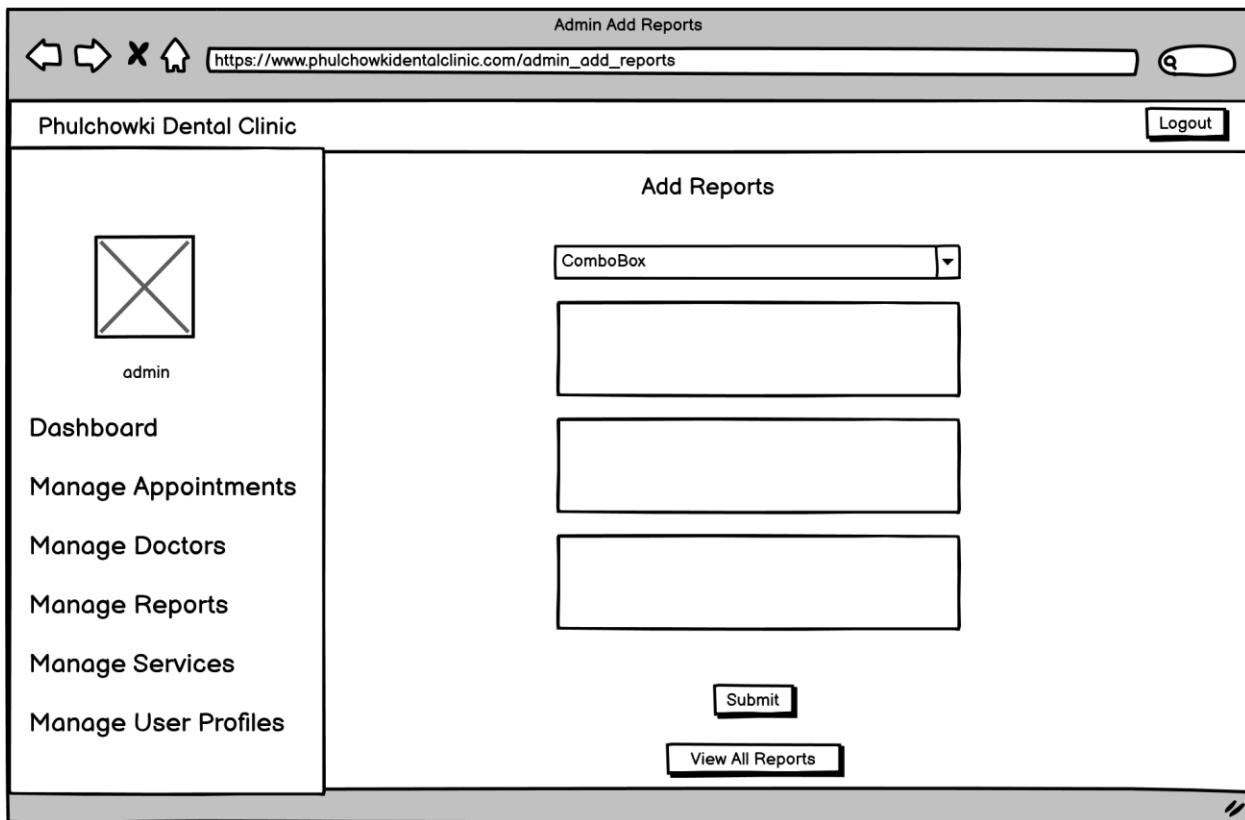


Figure 426: Wireframe - Admin View Report

20. Admin Add Report Page



The wireframe illustrates the 'Admin Add Reports' page for Phulchowki Dental Clinic. The page is titled 'Admin Add Reports' at the top center. On the left, there is a sidebar with a user icon labeled 'admin' and a list of management functions: Dashboard, Manage Appointments, Manage Doctors, Manage Reports, Manage Services, and Manage User Profiles. The main content area is titled 'Add Reports' and contains three empty rectangular input fields stacked vertically. Below these fields is a 'Submit' button and a 'View All Reports' button.

Figure 427: Wireframe - Admin Add Report

21. Admin View Service Page

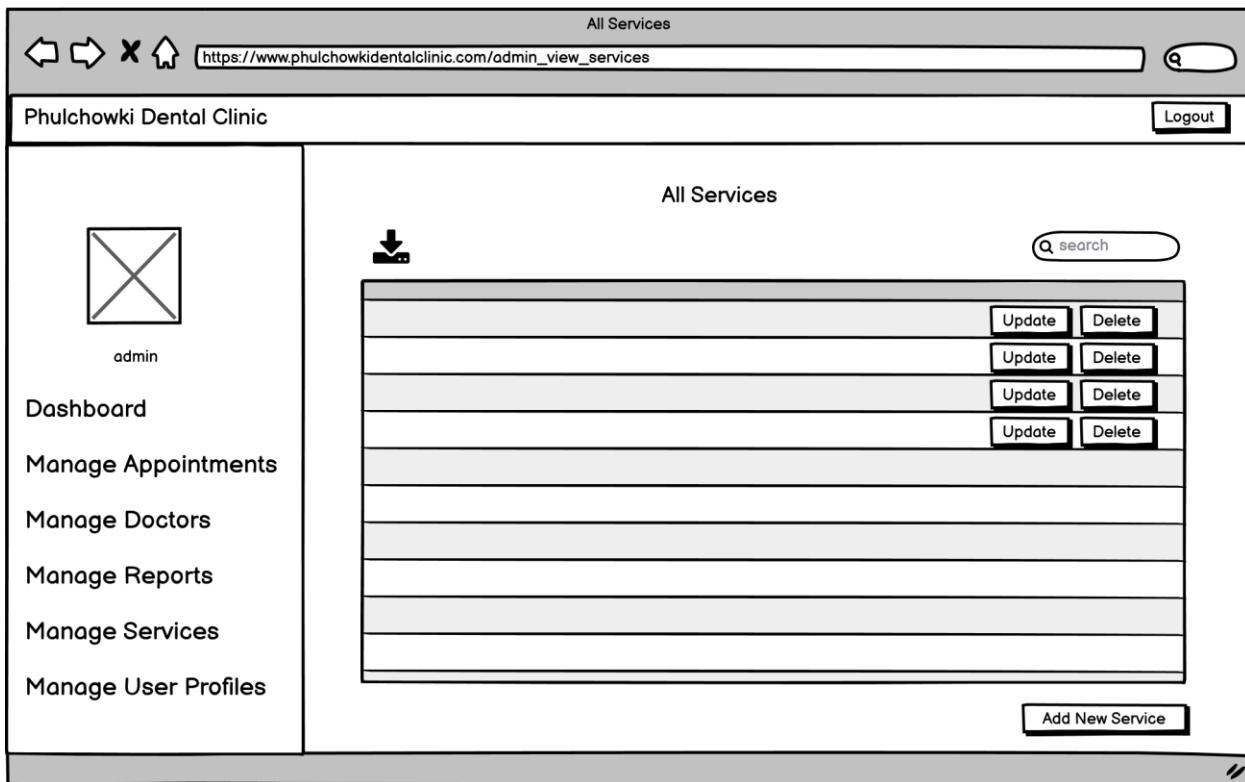


Figure 428: Wireframe - Admin View Service

22. Admin Add Service Page

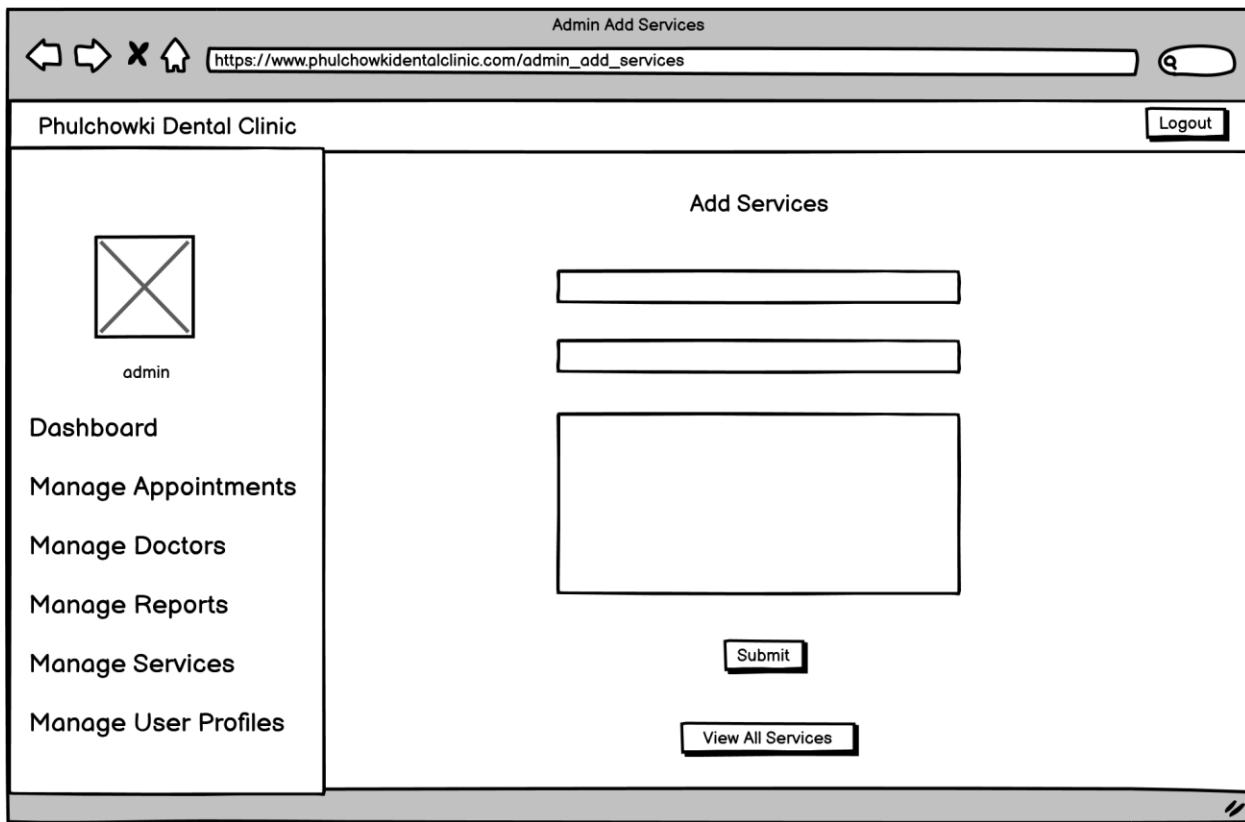


Figure 429: Wireframe - Admin Add Service

23. Admin View Users Page

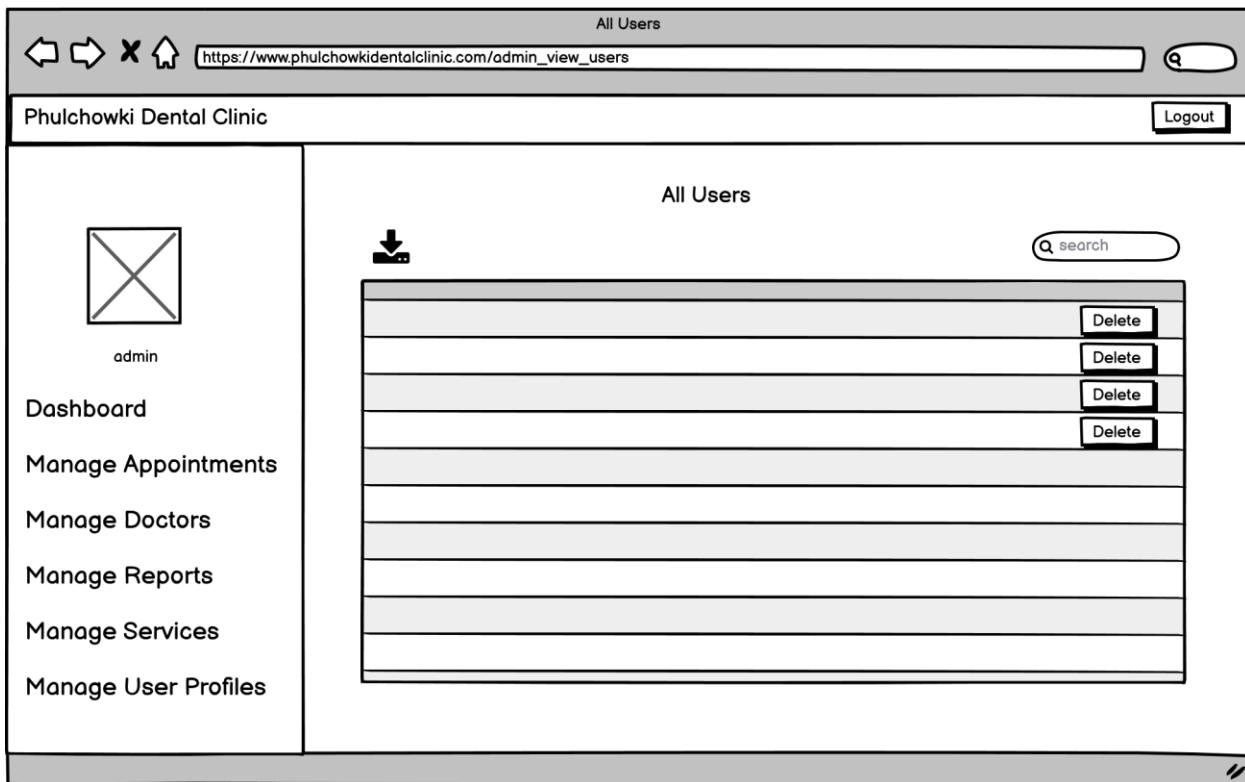


Figure 430: Wireframe - Admin View Users

7.7 Appendix G: Testing

7.7.1 Black Box Testing

[Continued from here](#)

Test No.5	
Objective	To view user's report by the user.
Action	The 'View Report' button was clicked in the side bar.
Expected Output	Only current user's reports would be shown.
Actual Output	All user's reports was shown.
Result	The test was unsuccessful.

Table 77: Black Box Test 5 (Unsuccessful)

Name	Appointment Date	Doctor	Service	Problems	Prescription	Other Details	Total Cost
Abhishek Rokaya	April 28, 2022	Prasamhsa Khadka	Tooth Extraction	Wisdom tooth was causing the patient a lot of pain.	Flexon 3-times a day after food. Wisgel- 2 times a day while brushing.	If there is still a lot of pain, please contact us as soon as possible.	Rs.15000
Abhishek Rokaya	April 26, 2022	Prasamhsa Khadka	General Visit	Everything seems to be perfectly fine except in hygiene. The teeth appears to be a bit yellow.	Brush at least 2 times a day.	Scaling is recommended to clean the teeth.	Rs.350
Lionel Messi	April 27, 2022	Animesh Basnet	General Visit	There is a cavity in the upper right canal teeth.	The teeth needs to be filled. For now, please brush at least 2 times a day.	Avoid drinking cold drinks.	Rs.350

Figure 431: Black Box Test 5 (Unsuccessful)

Code that caused the error:

```
#View for seeing all of user's report.
@login_required(login_url='/login')
@user_passes_test(is_patient)
def user_view_reports(request):
    patientUser = PatientUser.objects.get(user=request.user.id)
    reports = Report.objects.all()
    return render(request,'user_view_reports.html', {'reports':reports , 'patientUser':patientUser})
```

Figure 432: Black Box Test 5 - Error Code

Fixing the code:

```
#View for seeing all of user's report.
@login_required(login_url='/login')
@user_passes_test(is_patient)
def user_view_reports(request):
    patientUser = PatientUser.objects.get(user=request.user.id)
    reports = Report.objects.filter(appointment_patientUser=patientUser).all()
    return render(request,'user_view_reports.html', {'reports':reports , 'patientUser':patientUser})
```

Figure 433: Black Box Test 5 - Fixed Code

Test No.5	
Objective	To view user's report by the user.
Action	The 'View Report' button was clicked in the side bar.
Expected Output	Only current user's reports would be shown.
Actual Output	Only current user's reports was shown.
Result	The test was successful.
Cause of Error	The report was not filtered according to current user ID.

Table 78: Black Box Test 5 (Successful)

Name	Appointment Date	Doctor	Service	Problems	Prescription	Other Details	Total Cost
Abhishek Rokaya	April 28, 2022	Prasamhsa Khadka	Tooth Extraction	Wisdom tooth was causing the patient a lot of pain.	Flexon 3-times a day after food. Wisgel- 2 times a day while brushing.	If there is still a lot of pain, please contact us as soon as possible.	Rs.1500
Abhishek Rokaya	April 26, 2022	Prasamhsa Khadka	General Visit	Everything seems to be perfectly fine except in hygiene. The teeth appears to be a bit yellow.	Brush at least 2 times a day.	Scaling is recommended to clean the teeth.	Rs.350

Figure 434: Black Box Test 5 (Successful)

Test No.6	
Objective	To book an appointment outside the clinic time.
Action	The booking form was filled with 4am as the booking time and the form was submitted by the user.
Expected Output	The booking would not be done and the error would be shown.
Actual Output	The booking was done and error was not shown.
Result	The test was unsuccessful.

Table 79: Black Box Test 6 (Unsuccessful)

The screenshot shows a web application for booking dental appointments. The main page title is "Book an Appointment". The form fields are pre-filled as follows:

- Doctor: Prasamhsa Khadka
- Service: Scaling
- Date: 04/26/2022
- Time: 04:00 AM

A text area below the time field contains the text "Scaling needed".

The left sidebar displays a user profile for "Kushal" and a navigation menu with the following items:

- Dashboard
- Appointments
- Doctors
- Reports
- Services
- My Profile

Figure 435: Black Box Test 6 (Unsuccessful) - 1

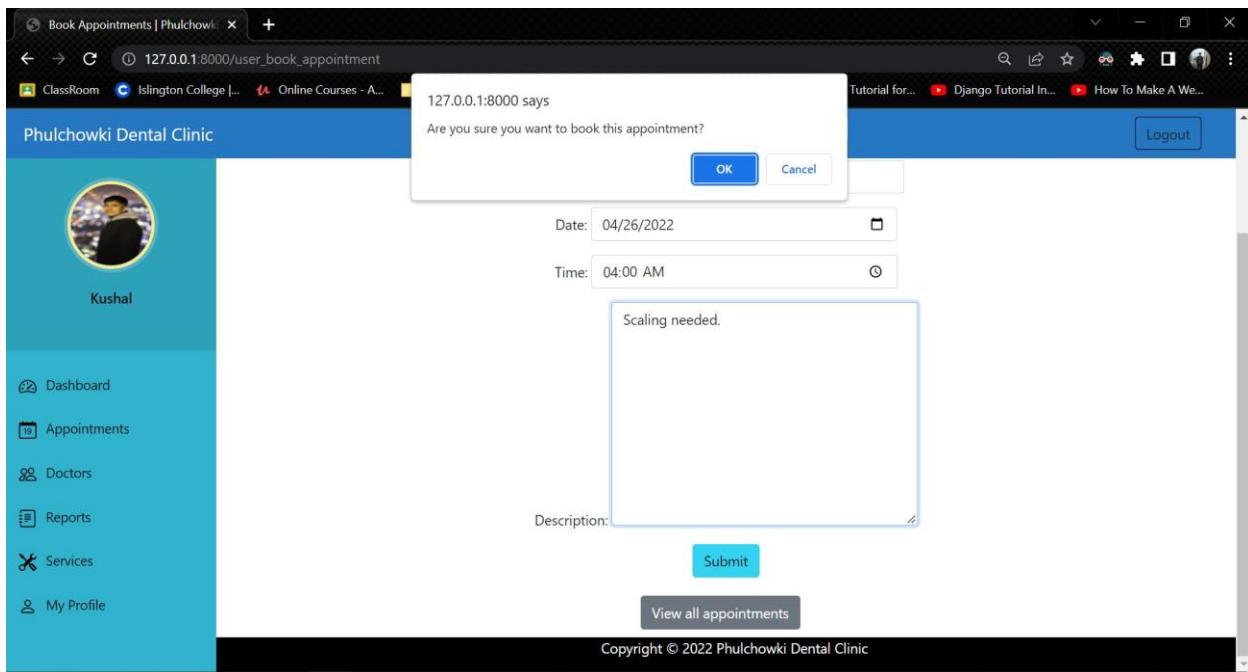


Figure 436: Black Box Test 6 (Unsuccessful) - 2

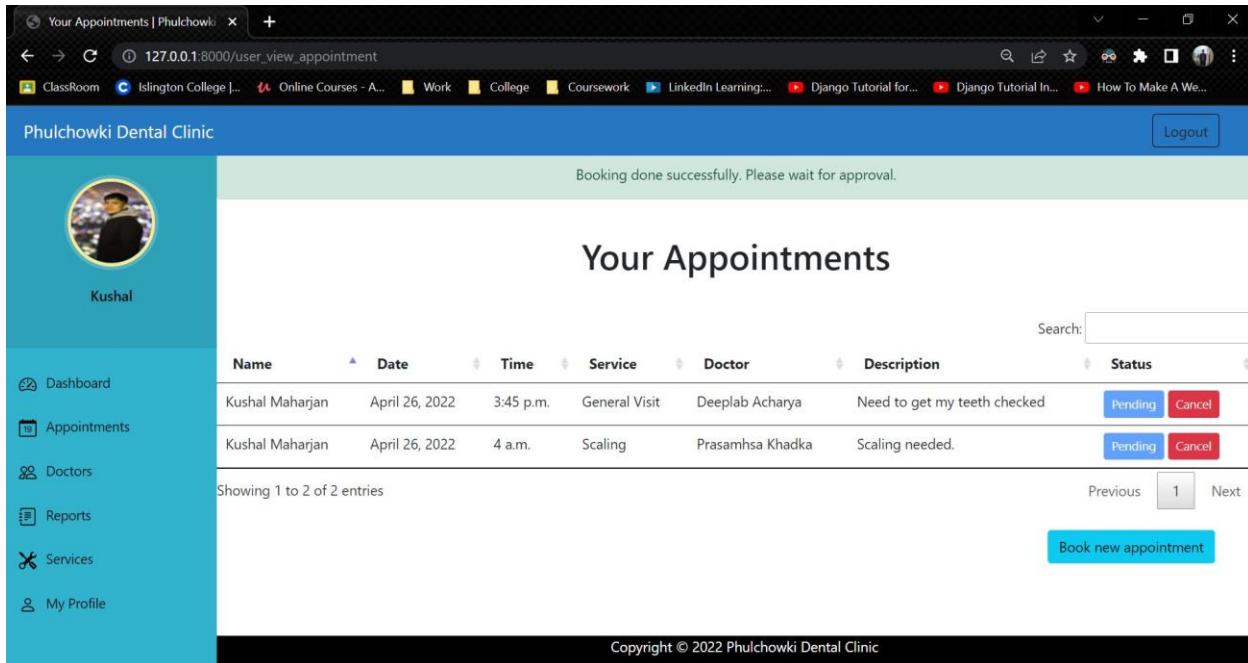


Figure 437: Black Box Test 6 (Unsuccessful) – 3

Code that caused the error:

```
class UserAppointmentForm(ModelForm):
    class Meta:
        model = Appointment
        fields = ('doctor', 'service', 'date', 'time', 'description')

        widgets = {
            'doctor': forms.Select(attrs={'class': 'form-control'}),
            'service': forms.Select(attrs={'class': 'form-control'}),
            'date': forms.DateInput(attrs={'class': 'form-control', 'type': 'date', 'id': 'datePickerId'}),
            'time': forms.TimeInput(attrs={'class': 'form-control', 'type': 'time'}),
            'description': forms.Textarea(attrs={'class': 'form-control'})}
```

*Figure 438: Black Box Test 6 - Error Code***Fixing the code:**

```
class UserAppointmentForm(ModelForm):
    class Meta:
        model = Appointment
        fields = ('doctor', 'service', 'date', 'time', 'description')

        widgets = {
            'doctor': forms.Select(attrs={'class': 'form-control'}),
            'service': forms.Select(attrs={'class': 'form-control'}),
            'date': forms.DateInput(attrs={'class': 'form-control', 'type': 'date', 'id': 'datePickerId'}),
            'time': forms.TimeInput(attrs={'class': 'form-control', 'type': 'time', 'min': '09:00:00', 'max': '17:00:00'}),
            'description': forms.Textarea(attrs={'class': 'form-control'})}
```

Figure 439: Black Box Test 6 - Fixed Code

Test No.6	
Objective	To book an appointment outside the clinic time.
Action	The booking form was filled with 4am as the booking time and the form was submitted by the user.
Expected Output	The booking would not be done and the error would be shown.
Actual Output	The booking was not done and error was shown.
Result	The test was successful.
Cause of Error	The mix and min value was not declared.

Table 80: Black Box Test 6 (Successful)

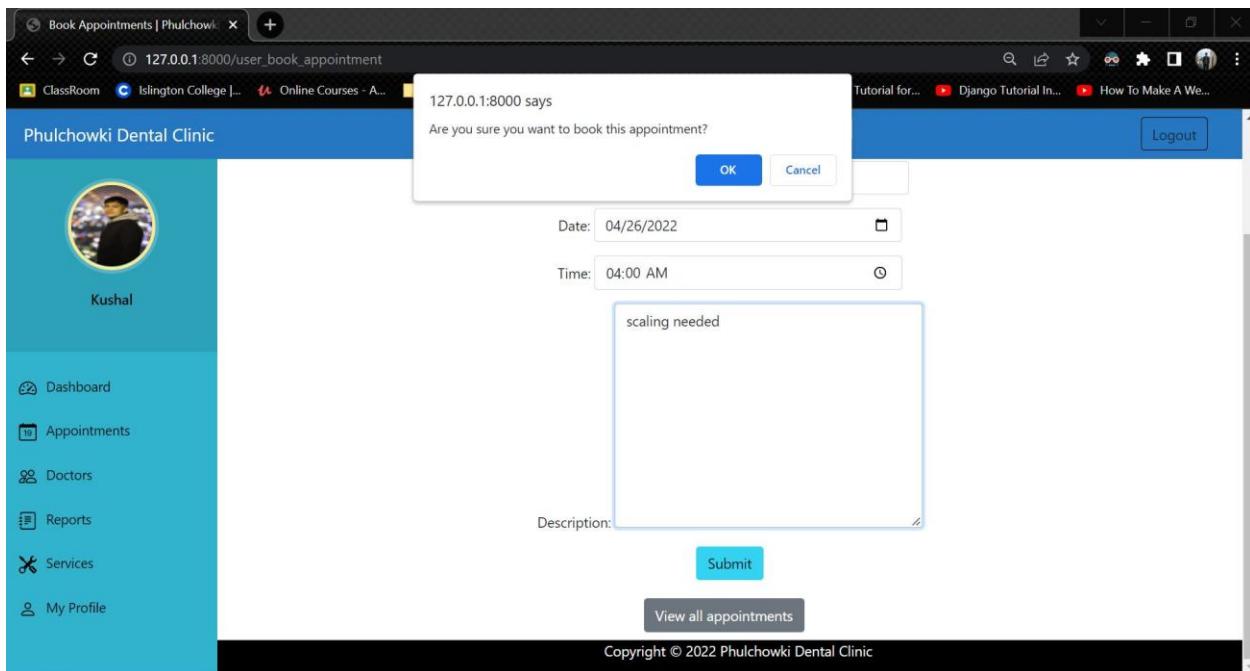


Figure 440: Black Box Test 6 (Successful) - 1

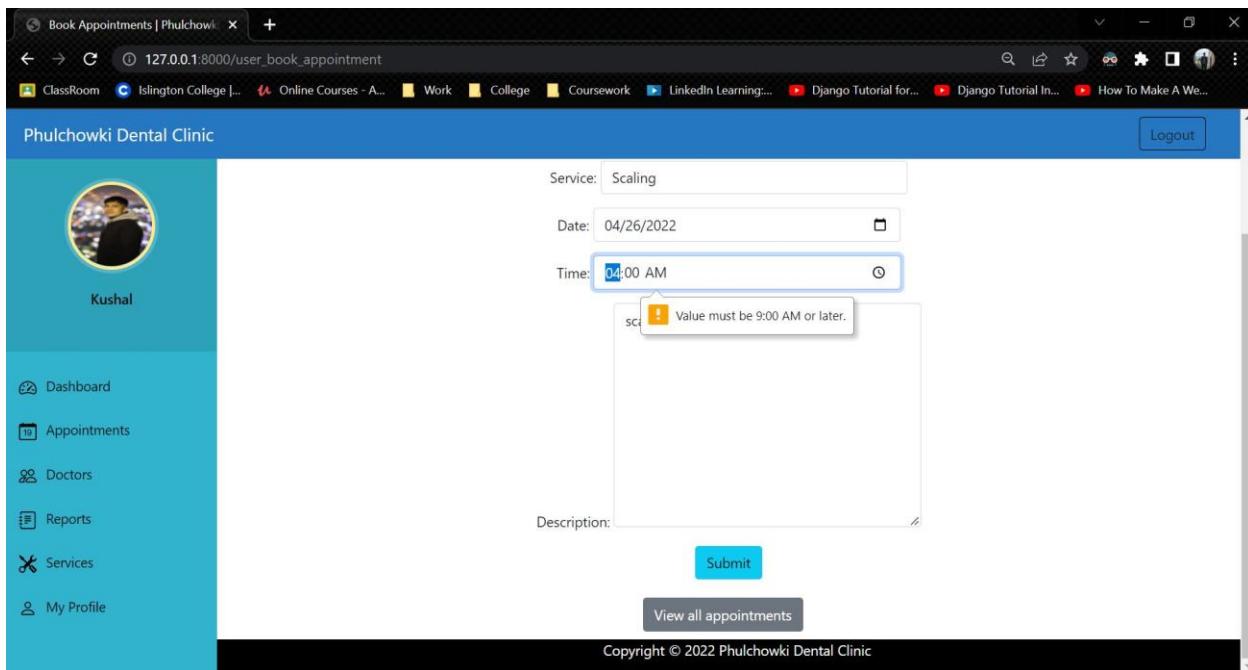


Figure 441: Black Box Test 6 (Successful) – 2

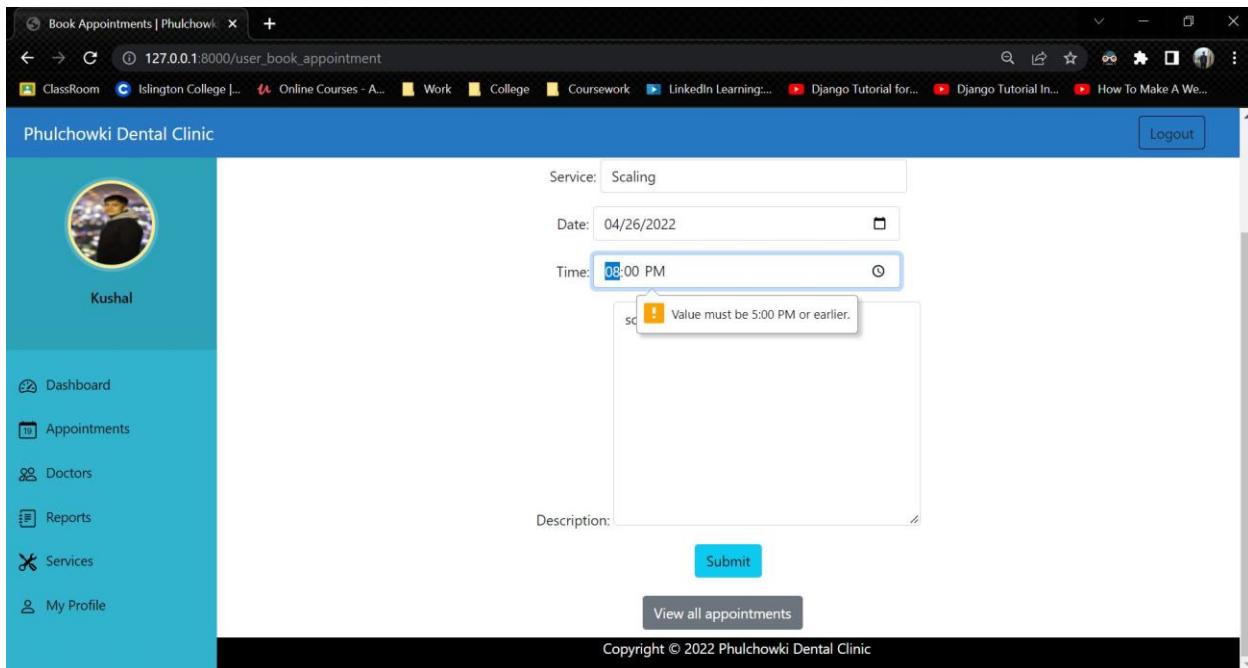


Figure 442: Black Box Test 6 (Successful) - 3

Here, the clinic time is 9am-6pm so the booking can only be done from 9am to 5pm.

Test No.7	
Objective	To view pending appointment by the admin.
Action	The 'Approve Appointment' button was clicked in the admin panel.
Expected Output	The list of pending appointments would be shown.
Actual Output	All the appointments which has been approved/rejected was also shown.
Result	The test was unsuccessful.

Table 81: Black Box Test 7 (Unsuccessful)

Pending Appointments | Phulchowki

127.0.0.1:8000/admin/approve_appointment

Phulchowki Dental Clinic

Pending Appointments

Name	Date	Time	Service	Doctor	Description	Status
Abhishek Rokaya	April 28, 2022	2:45 p.m.	Tooth Extraction	Prasamhsa Khadka	I need to extract my wisdom tooth.	
Abhishek Rokaya	April 29, 2022	12:50 p.m.	General Visit	Animesh Basnet	General checkup needed.	
Abhishek Rokaya	May 2, 2022	2 p.m.	Scaling	Animesh Basnet	I need to get my teeth cleaned.	
Abhishek Rokaya	May 6, 2022	4 p.m.	General Visit	Rabina Lama	Need a general visit.	<button>Approve</button> <button>Reject</button>
Abhishek Rokaya	April 26, 2022	2:01 p.m.	General Visit	Prasamhsa Khadka	A general visit with the doctor	
Lionel Messi	April 29, 2022	4 p.m.	Scaling	Prasamhsa Khadka	Scaling needed.	
Lionel Messi	April 27, 2022	4:30 p.m.	General Visit	Animesh Basnet	A regular checkup	
Amrit Nepali	May 7, 2022	3 p.m.	Tooth Extraction	Rabina Lama	tooth	
Lionel Messi	May 6, 2022	3 p.m.	General Visit	Suresh Magar	a general check-up	

Copyright © 2022 Phulchowki Dental Clinic

Figure 443: Black Box Test 7 (Unsuccessful)

Code that caused the error:

```
#view for viewing all the pending appointment list.  
@login_required(login_url='/adminlogin')  
@user_passes_test(is_admin)  
def admin_approve_appointment(request):  
    appointments = Appointment.objects.all().filter(status__gte=0)  
    return render(request,'admin_approve_appointment.html', {'appointment':appointments})
```

Figure 444: Black Box Test 7 - Error Code

Fixing the code:

```
#view for viewing all the pending appointment list.  
@login_required(login_url='/adminlogin')  
@user_passes_test(is_admin)  
def admin_approve_appointment(request):  
    appointments = Appointment.objects.all().filter(status=0)  
    return render(request,'admin_approve_appointment.html', {'appointment':appointments})
```

Figure 445: Black Box Test 7 - Fixed Code

Test No.7	
Objective	To view pending appointment by the admin.
Action	The 'Approve Appointment' button was clicked in the admin panel.
Expected Output	The list of pending appointments would be shown.
Actual Output	The list of only pending appointments was be shown.
Result	The test was successful.
Cause of Error	The status was shown of greater than 0 but should be 0.

Table 82: Black Box Test 7 (Successful)

Pending Appointments

Name	Date	Time	Service	Doctor	Description	Status
Abhishek Rokaya	May 6, 2022	4 p.m.	General Visit	Rabina Lama	Need a general visit.	<button>Approve</button> <button>Reject</button>
Kushal Maharjan	April 26, 2022	3:45 p.m.	General Visit	Deeplab Acharya	Need to get my teeth checked	<button>Approve</button> <button>Reject</button>
Kushal Maharjan	April 26, 2022	2 p.m.	Scaling	Prasamhsa Khadka	scaling needed	<button>Approve</button> <button>Reject</button>

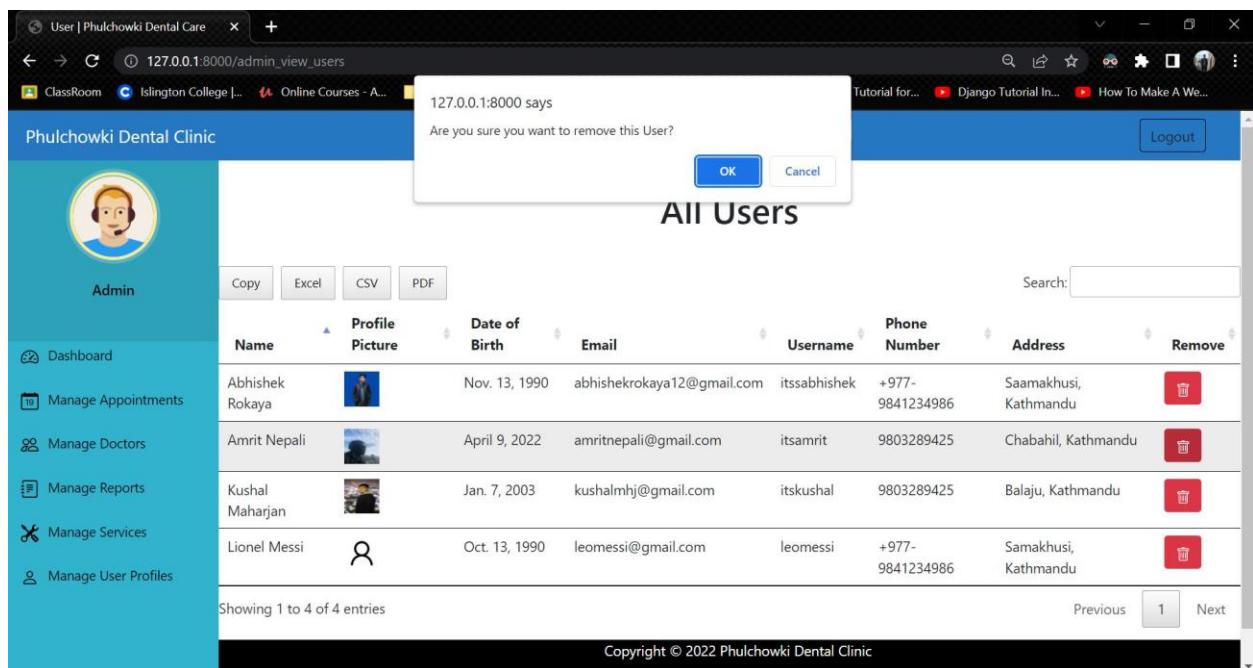
Showing 1 to 3 of 3 entries

Copyright © 2022 Phulchowki Dental Clinic

Figure 446: Black Box Test 7 (Successful)

Test No.8	
Objective	To delete user from the system by the admin.
Action	The delete button of the user to be deleted was clicked.
Expected Output	The user would be deleted.
Actual Output	An error was shown and the user was not deleted.
Result	The test was unsuccessful.

Table 83: Black Box Test 8 (Unsuccessful)



A screenshot of a web browser window titled "User | Phulchowki Dental Care". The URL is "127.0.0.1:8000/admin_view_users". The page displays a list of users under the heading "All Users". A modal dialog box is open in the center, asking "Are you sure you want to remove this User?". There are "OK" and "Cancel" buttons. The browser's address bar shows "127.0.0.1:8000 says" above the main content area.

Name	Profile Picture	Date of Birth	Email	Username	Phone Number	Address	Remove
Abhishek Rokaya		Nov. 13, 1990	abhishekrokaya12@gmail.com	itssabhishek	+977-9841234986	Saamakhusi, Kathmandu	
Amrit Nepali		April 9, 2022	amritnepali@gmail.com	itsamrit	9803289425	Chabahil, Kathmandu	
Kushal Mahajan		Jan. 7, 2003	kushalmjh@gmail.com	itskushal	9803289425	Balaju, Kathmandu	
Lionel Messi		Oct. 13, 1990	leomessi@gmail.com	leomessi	+977-9841234986	Samakhusi, Kathmandu	

Showing 1 to 4 of 4 entries

Copyright © 2022 Phulchowki Dental Clinic

Figure 447: Black Box Test 8 (Unsuccessful) - 1

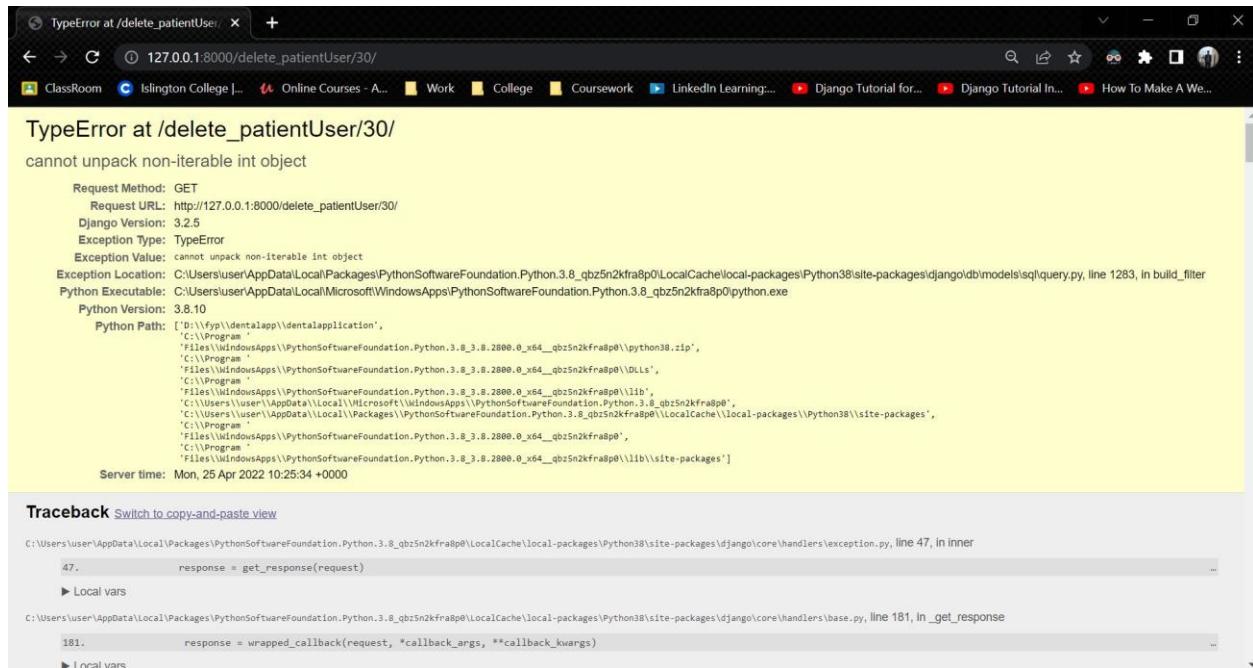


Figure 448: Black Box Test 8 (Unsuccessful) – 2

Code that caused the error:

```
#View for removing the selected users from the database
@login_required(login_url='/adminlogin')
@user_passes_test(is_admin)
def delete_patientUser(request, pk):
    user = User.objects.get(pk)
    user.delete()
    messages.warning(request, 'User has been removed.')
    return redirect(['admin_view_users'])
```

Figure 449: Black Box Test 8 - Error Code

Fixing the code:

```
#View for removing the selected users from the database
@login_required(login_url='/adminlogin')
@user_passes_test(is_admin)
def delete_patientUser(request, pk):
    user = User.objects.get(pk=pk)
    user.delete()
    messages.warning(request, 'User has been removed.')
    return redirect('admin_view_users')
```

Figure 450: Black Box Test 8 - Fixed Code

Test No.8	
Objective	To delete user from the system by the admin.
Action	The delete button of the user to be deleted was clicked.
Expected Output	The user would be deleted.
Actual Output	The user was deleted.
Result	The test was successful.
Cause of Error	The id was not specified for deleting which caused error in deleting the selected user.

Table 84: Black Box Test 8 (Successful)

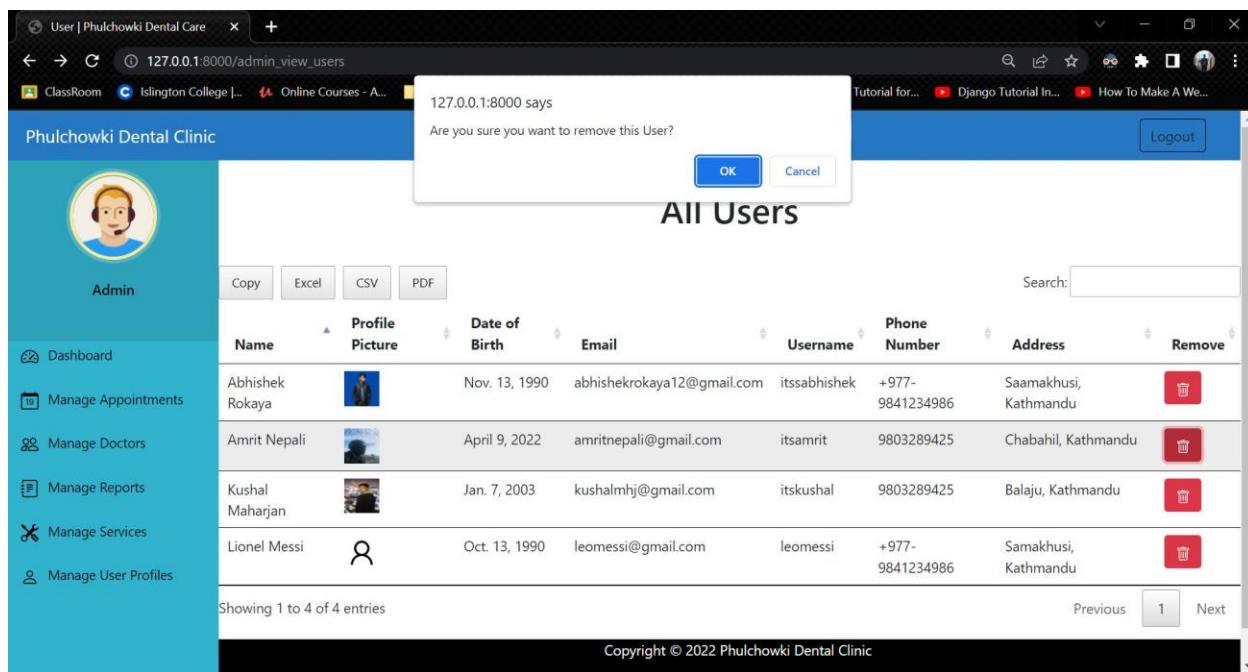


Figure 451: Black Box Test 8 (Successful) - 1

Name	Profile Picture	Date of Birth	Email	Username	Phone Number	Address	Remove
Abhishek Rokaya		Nov. 13, 1990	abhishekrokaya12@gmail.com	itssabhishek	+977-9841234986	Saamakhusi, Kathmandu	
Kushal Maharjan		Jan. 7, 2003	kushalmhj@gmail.com	itskushal	9803289425	Balaju, Kathmandu	
Lionel Messi		Oct. 13, 1990	leomessi@gmail.com	leomessi	+977-9841234986	Samakhusi, Kathmandu	

Showing 1 to 3 of 3 entries

Copyright © 2022 Phulchowki Dental Clinic

Figure 452: Black Box Test 8 (Successful) - 2

#	id	password	last_login	is_superuser	username	last_name	email	is_staff	is_active	date_joined	first_name
1	pbkdf2_sh...	2022-04-2...	1	admin				1	1	2022-02-0...	
20	pbkdf2_sh...	2022-04-2...	0	itssabhishek	Rokaya	abhishek...	abhishek...	0	1	2022-04-0...	Abhishek
21	pbkdf2_sh...	2022-04-2...	0	leomessi	Messi	leomessi...	leomessi...	0	1	2022-04-0...	Lionel
30	pbkdf2_sh...	2022-04-2...	0	itsamrit	Nepali	amritnepal...	amritnepal...	0	1	2022-04-1...	Amrit
32	pbkdf2_sh...	2022-04-2...	0	itskushal	Maharjan	kushalmh...	kushalmh...	0	1	2022-04-2...	Kushal

Figure 453: Black Box Test 8 - Database before

#	id	password	last_login	is_superuser	username	last_name	email	is_staff	is_active	date_joined	first_name
1	pbkdf2_sh...	2022-04-2...	1	admin				1	1	2022-02-0...	
20	pbkdf2_sh...	2022-04-2...	0	itssabhishek	Rokaya	abhishek...	abhishek...	0	1	2022-04-0...	Abhishek
21	pbkdf2_sh...	2022-04-2...	0	leomessi	Messi	leomessi...	leomessi...	0	1	2022-04-0...	Lionel
32	pbkdf2_sh...	2022-04-2...	0	itskushal	Maharjan	kushalmh...	kushalmh...	0	1	2022-04-2...	Kushal

Figure 454: Black Box Test 8 - Database after

[View the rest of the Black Box Testing Here.](#)

[View the test plan for the Black Box Testing Here.](#)

7.7.2 White Box Testing

[Continued from here](#)

Test No.6	
Objective	To view doctor by user.
Action	The 'Doctors' button in the sidebar was clicked.
Expected Output	The user would see the available doctors with their details.
Actual Output	The user saw the available doctors with their details.
Result	The test was successful.

Table 85: White Box Test 6

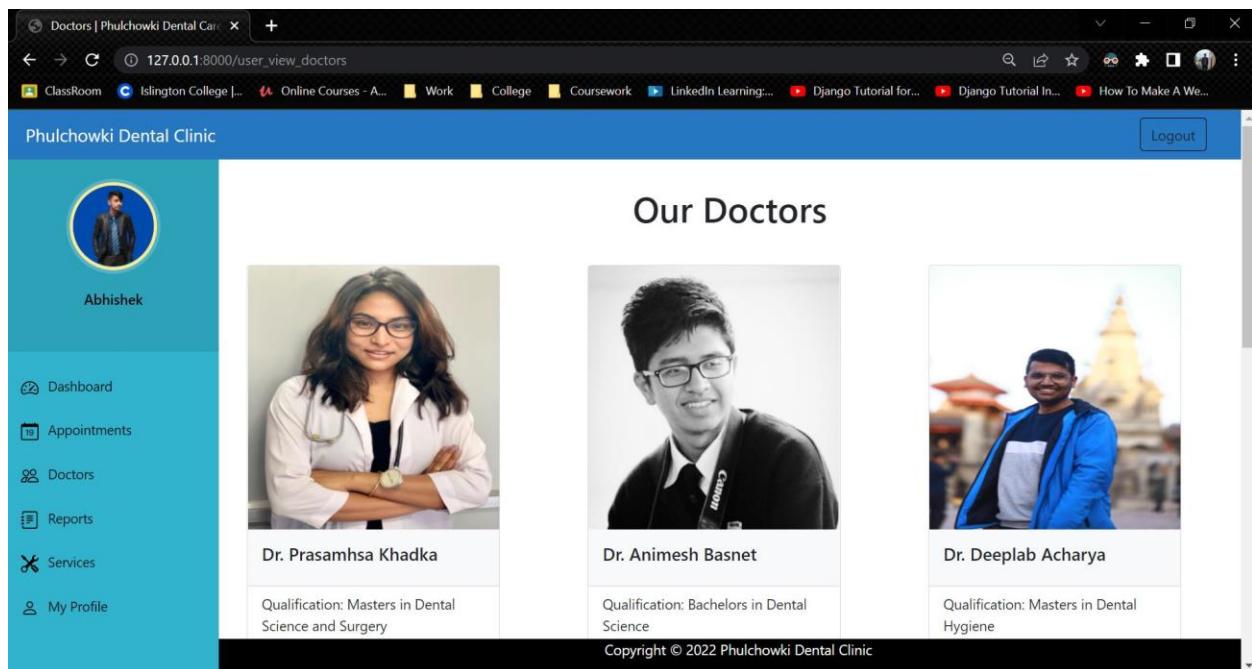


Figure 455: White Box Test 6 - 1

Doctors | Phulchowki Dental Care

127.0.0.1:8000/user_view_doctors

Logout

Phulchowki Dental Clinic

Abhishek

Dashboard Appointments Doctors Reports Services My Profile

	Dr. Prasamhsa Khadka		Dr. Animesh Basnet
Qualification: Masters in Dental Science and Surgery	Qualification: Bachelors in Dental Science	Experience: 5 years	Experience: 3 years
Phone Number: +977-9861239634	Phone Number: +977-9841273098	Email: prsma12@gmail.com	Email: animesh.b@gmail.com
Book an Appointment now.		Book an Appointment now.	

	Dr. Deeplab Acharya
Qualification: Masters in Dental Hygiene	Experience: 4 years
Phone Number: +977-9818273946	Email: deeplove123@gmail.com
Book an Appointment now.	

Copyright © 2022 Phulchowki Dental Clinic

Figure 456: White Box Test 6 – 2

Doctors | Phulchowki Dental Care

127.0.0.1:8000/user_view_doctors

Logout

Phulchowki Dental Clinic

Abhishek

Dashboard Appointments Doctors Reports Services My Profile

	Dr. Subikhyat Ghimire		Dr. Rabina Lama		Dr. Suresh Magar
Qualification: Bachelors in Dental Science	Qualification: Masters in Dental Surgery	Experience: 2 years	Experience: 6 years	Phone Number: +977-9851117284	Phone Number: +977-9866283956
Book an Appointment now.		Book an Appointment now.		Book an Appointment now.	

Copyright © 2022 Phulchowki Dental Clinic

Figure 457: White Box Test 6 - 3

Test No.7	
Objective	To view service by user.
Action	The 'Service' button in the sidebar was clicked.
Expected Output	The user would see the available services with their details.
Actual Output	The user saw the available services with their details.
Result	The test was successful.

Table 86: White Box Test 7

The screenshot shows a web application interface for a dental clinic. The top navigation bar includes a logo, the text 'Services | Phulchowki Dental Clinic', and a search bar. Below the header, a blue navigation bar displays the clinic's name and a 'Logout' button. On the left, a sidebar menu lists 'Abhishek' (profile picture), 'Dashboard', 'Appointments', 'Doctors', 'Reports', 'Services' (selected), and 'My Profile'. The main content area is titled 'Our Services' and contains a table with the following data:

Name	Cost	Description
Braces	28000	We do all kinds of braces implantation to align your teeth. The base price starts from Rs.28000.
General Visit	350	We do a thorough check up on this and recommend necessary treatments.
RCT	4500	Root Canal Treatment (RCT) is a procedure done when the tooth decay reaches the pulp (innermost layer of the tooth) causing inflammation.
Scaling	1500	Scaling is the cleaning of the teeth.
Tooth Extraction	15000	Tooth extraction is done when the teeth needs to be removed. The costs started from 15000 and can be raised according to the complexity.

At the bottom of the main content, it says 'Showing 1 to 5 of 5 entries'. The footer contains a copyright notice: 'Copyright © 2022 Phulchowki Dental Clinic'.

Figure 458: White Box Test 7

Test No.8	
Objective	To download the report in pdf by the user.
Action	The 'Report' button in the sidebar was clicked to view the reports. Then the 'PDF' button was clicked.
Expected Output	The reports would be downloaded.
Actual Output	The reports were downloaded.
Result	The test was successful.

Table 87: White Box Test 8

Name	Appointment Date	Doctor	Service	Problems	Prescription	Other Details	Total Cost
Lionel Messi	April 27, 2022	Animesh Basnet	General Visit	There is a cavity in the upper right canal teeth.	The teeth needs to be filled. For now, please brush at least 2 times a day.	Avoid drinking cold drinks.	Rs.350

Figure 459: White Box Test 8 - 1

Your Reports

Name	Appointment Date	Doctor	Service	Problems	Prescription	Other Details	Total Cost
Lionel Messi	April 27, 2022	Animesh Basnet	General Visit	There is a cavity in the upper right canal teeth.	The teeth needs to be filled. For now, please brush at least 2 times a day.	Avoid drinking cold drinks.	Rs.350

Showing 1 to 1 of 1 entries

Copyright © 2022 Phulchowki Dental Clinic

Figure 460: White Box Test 8 – 2

Reports
| Phulchowki Dental Care

Name	Appointment Date	Doctor	Service	Problems	Prescription	Other Details	Total Cost
Lionel Messi	April 27, 2022	Animesh Basnet	General Visit	There is a cavity in the upper right canal teeth.	The teeth needs to be filled. For now, please brush at least 2 times a day.	Avoid drinking cold drinks.	Rs.350

Figure 461: White Box Test 8 - 3

Test No.9	
Objective	To sort the appointment according to the date.
Action	The 'Appointment' button was clicked in the sidebar to view all appointments. Then the 'Appointment date' header of the table was clicked to sort it according to date.
Expected Output	The appointments would be sorted according to date.
Actual Output	The appointments was sorted according to date.
Result	The test was successful.

Table 88: White Box Test 9

Name	Date	Doctor	Service	Problems	Prescription	Other Details	Total Cost
Abhishek Rokaya	April 28, 2022	Prasamhsa Khadka	Tooth Extraction	Wisdom tooth was causing the patient a lot of pain.	Flexon 3-times a day after food. Wigel- 2 times a day while brushing.	If there is still a lot of pain, please contact us as soon as possible.	Rs.15000
Abhishek Rokaya	April 26, 2022	Prasamhsa Khadka	General Visit	Everything seems to be perfectly fine except in hygiene. The teeth appears to be a bit yellow.	Brush at least 2 times a day.	Scaling is recommended to clean the teeth.	Rs.350
Abhishek Rokaya	April 29, 2022	Animesh Basnet	General Visit	The teeth seems to be yellowish than normal.	Brush at least 2 times a day.	Scaling is recommended once every six months.	Rs.350

Figure 462: White Box Test 9 - 1

Your Reports

Name	Appointment Date	Doctor	Service	Problems	Prescription	Other Details	Total Cost
Abhishek Rokaya	April 29, 2022	Animesh Basnet	General Visit	The teeth seems to be yellowish than normal.	Brush at least 2 times a day.	Scaling is recommended once every six months.	Rs.350
Abhishek Rokaya	April 28, 2022	Prasamhsa Khadka	Tooth Extraction	Wisdom tooth was causing the patient a lot of pain.	Flexon 3-times a day after food. Wisgel- 2 times a day while brushing.	If there is still a lot of pain, please contact us as soon as possible.	Rs.15000
Abhishek Rokaya	April 26, 2022	Prasamhsa Khadka	General Visit	Everything seems to be perfectly fine except in hygiene. The teeth appears to be a bit yellow.	Brush at least 2 times a day.	Scaling is recommended to clean the teeth.	Rs.350

Showing 1 to 3 of 3 entries

Copyright © 2022 Phulchowki Dental Clinic

Figure 463: White Box Test 9 - 2

Test No.10	
Objective	To search the report.
Action	The 'Report' button in the sidebar was clicked to view the reports. Then 'April 29' was entered to filter the report list of April 29 only.
Expected Output	The reports would be filtered.
Actual Output	The reports were filtered.
Result	The test was successful.

Table 89: White Box Test 10

The screenshot shows a web application interface for a dental clinic. On the left, there's a sidebar with icons for Dashboard, Appointments, Doctors, Reports, Services, and My Profile. The main area is titled "Your Reports". It features a table with the following data:

Name	Appointment Date	Doctor	Service	Problems	Prescription	Other Details	Total Cost
Abhishek Rokaya	April 29, 2022	Animesh Basnet	General Visit	The teeth seem to be yellowish than normal.	Brush at least 2 times a day.	Scaling is recommended once every six months.	Rs.350
Abhishek Rokaya	April 28, 2022	Prasamhsa Khadka	Tooth Extraction	Wisdom tooth was causing the patient a lot of pain.	Flexon 3-times a day after food. Wisgel- 2 times a day while brushing.	If there is still a lot of pain, please contact us as soon as possible.	Rs.15000
Abhishek Rokaya	April 26, 2022	Prasamhsa Khadka	General Visit	Everything seems to be perfectly fine except in hygiene. The teeth appears to be a bit yellow.	Brush at least 2 times a day.	Scaling is recommended to clean the teeth.	Rs.350

At the bottom, it says "Showing 1 to 3 of 3 entries".

Figure 464: White Box Test 10 - 1

The screenshot shows a web application interface for 'Phulchowki Dental Clinic'. On the left is a sidebar with a user profile picture of 'Abhishek' and a list of navigation items: Dashboard, Appointments, Doctors, Reports (selected), Services, and My Profile. The main content area is titled 'Your Reports' and displays a table of appointment details. The table has columns for Name, Date, Doctor, Service, Problems, Prescription, Other Details, and Total Cost. One entry is shown for 'Abhishek Rokaya' on 'April 29, 2022' with 'Animesh Basnet' as the doctor, 'General Visit' as the service, 'The teeth seems to be yellowish than normal.' as the problem, 'Brush at least 2 times a day.' as the prescription, 'Scaling is recommended once every six months.' as other details, and a total cost of 'Rs.350'. There are buttons for Copy, Excel, CSV, PDF, and a search bar set to 'April 29'. At the bottom, it says 'Showing 1 to 1 of 1 entries (filtered from 3 total entries)' and includes previous/next page links. The footer copyright notice is 'Copyright © 2022 Phulchowki Dental Clinic'.

Appointment							
Name	Date	Doctor	Service	Problems	Prescription	Other Details	Total Cost
Abhishek Rokaya	April 29, 2022	Animesh Basnet	General Visit	The teeth seems to be yellowish than normal.	Brush at least 2 times a day.	Scaling is recommended once every six months.	Rs.350

Figure 465: White Box Test 10 - 2

Test No.11	
Objective	To view user's profile.
Action	The 'My Profile' button in the sidebar was clicked.
Expected Output	The user's profile with the logged in user's data would be shown.
Actual Output	The user's profile with the logged in user's data was shown.
Result	The test was successful.

Table 90: White Box Test 11

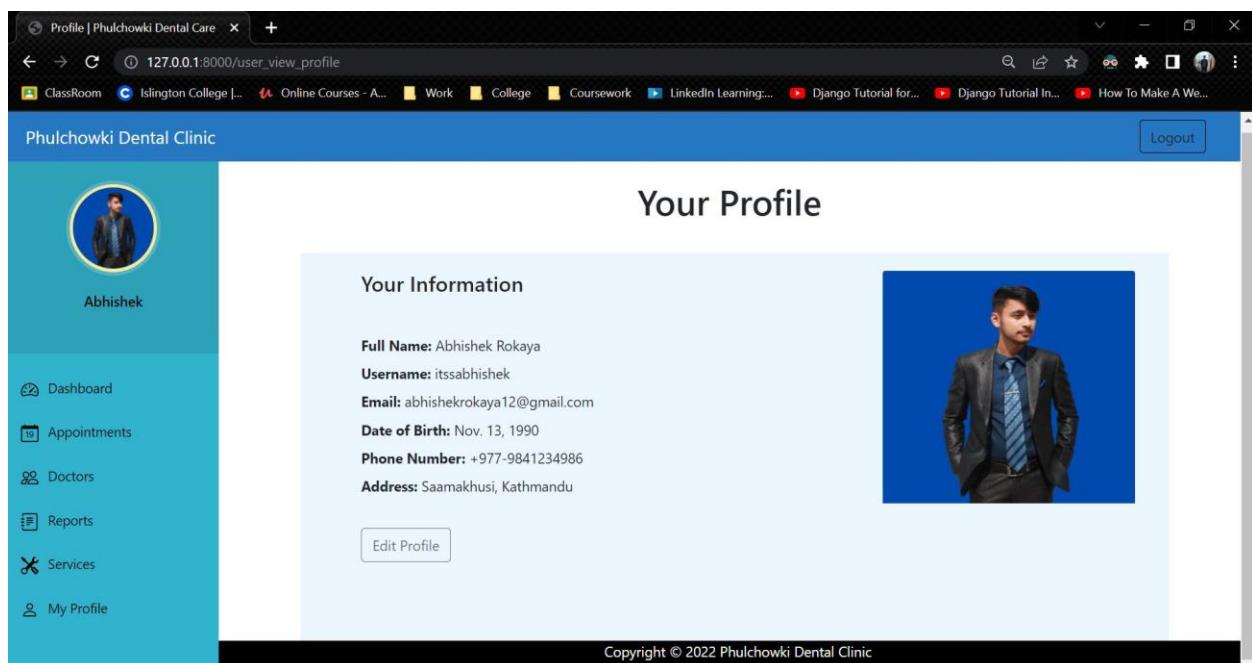


Figure 466: White Box Test 11

Test No.12	
Objective	To logout.
Action	The 'logout' button in the top right was clicked.
Expected Output	The user would be logged out.
Actual Output	The user was logged out.
Result	The test was successful.

Table 91: White Box Test 12

The screenshot shows the user dashboard for Phulchowki Dental Clinic. On the left, there's a sidebar with a profile picture of Abhishek and links for Dashboard, Appointments, Doctors, Reports, Services, and My Profile. The main area has a title 'Dashboard' and five summary boxes: Total Doctors (6), Total Reports (3), Total Services (5), Finished Appointments (4), and Pending Appointments (0). Below this is a section for 'Upcoming Appointments' with a table:

Name	Date	Time	Service	Doctor	Description	Status
Abhishek Rokaya	April 26, 2022	4:30 p.m.	Tooth Extraction	Prasamhsa Khadka	need to get an extraction.	Confirmed

At the bottom, there's a 'View all appointments' button and a copyright notice: 'Copyright © 2022 Phulchowki Dental Clinic'.

Figure 467: White Box Test 12 - 1

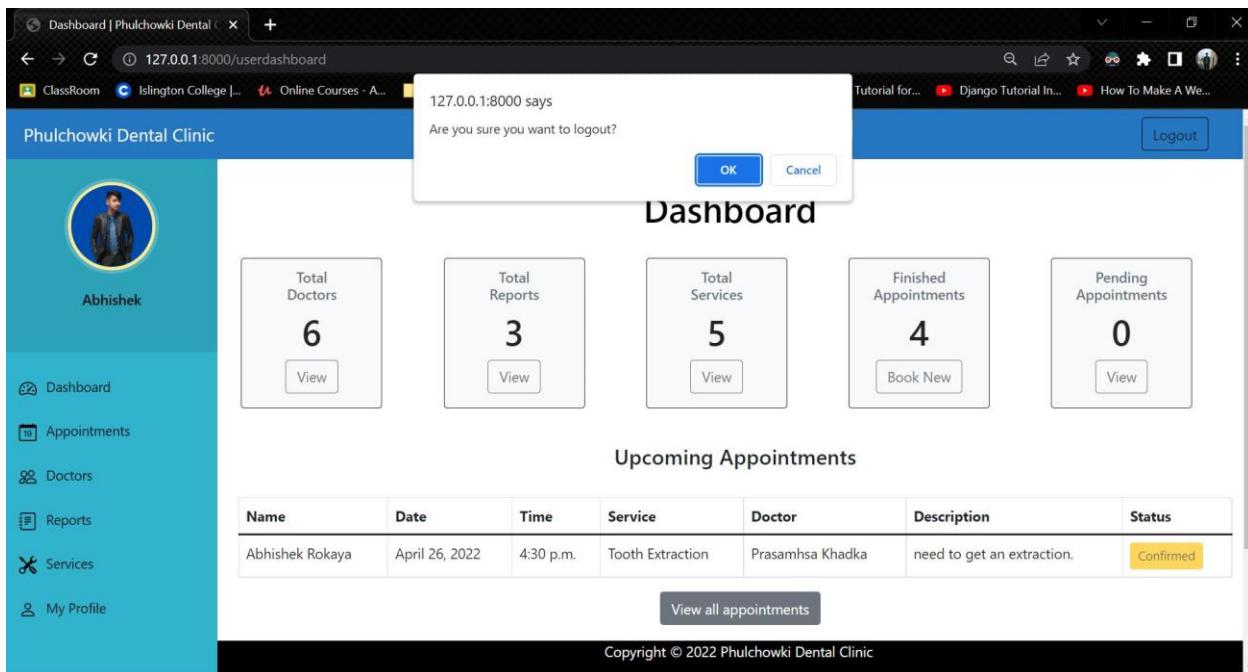


Figure 468: White Box Test 12 – 2

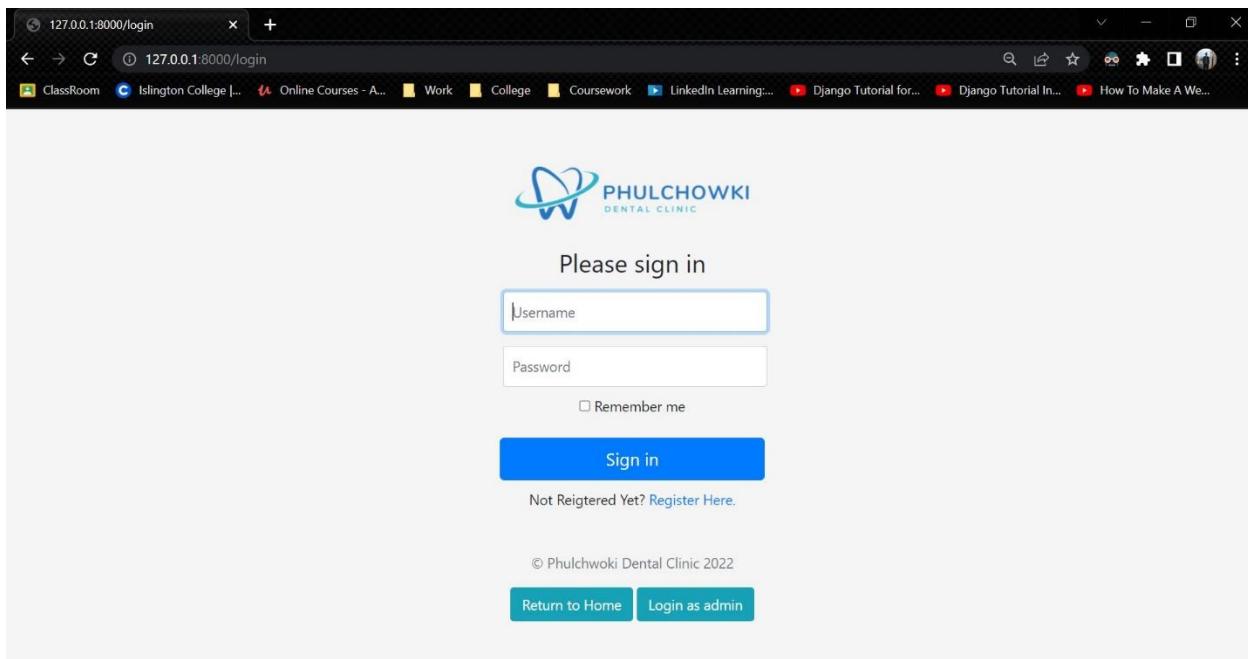


Figure 469: White Box Test 12 - 3

Test No.13	
Objective	To approve appointment of the user by the admin.
Action	The 'approve' button was clicked in the pending appointment list.
Expected Output	The appointment would be approved.
Actual Output	The appointment was approved.
Result	The test was successful.

Table 92: White Box Test 13

Pending Appointments | Phulchowki Dental Clinic

Phulchowki Dental Clinic

Pending Appointments

Name	Date	Time	Service	Doctor	Description	Status
Abhishek Rokaya	May 6, 2022	4 p.m.	General Visit	Rabina Lama	Need a general visit.	<button>Approve</button> <button>Reject</button>
Abhishek Rokaya	April 26, 2022	4:30 p.m.	Tooth Extraction	Prasamhsa Khadka	need to get an extraction.	<button>Approve</button> <button>Reject</button>

Showing 1 to 2 of 2 entries

Previous 1 Next

All Appointments Book an Appointment

Copyright © 2022 Phulchowki Dental Clinic

Figure 470: White Box Test 13 - 1

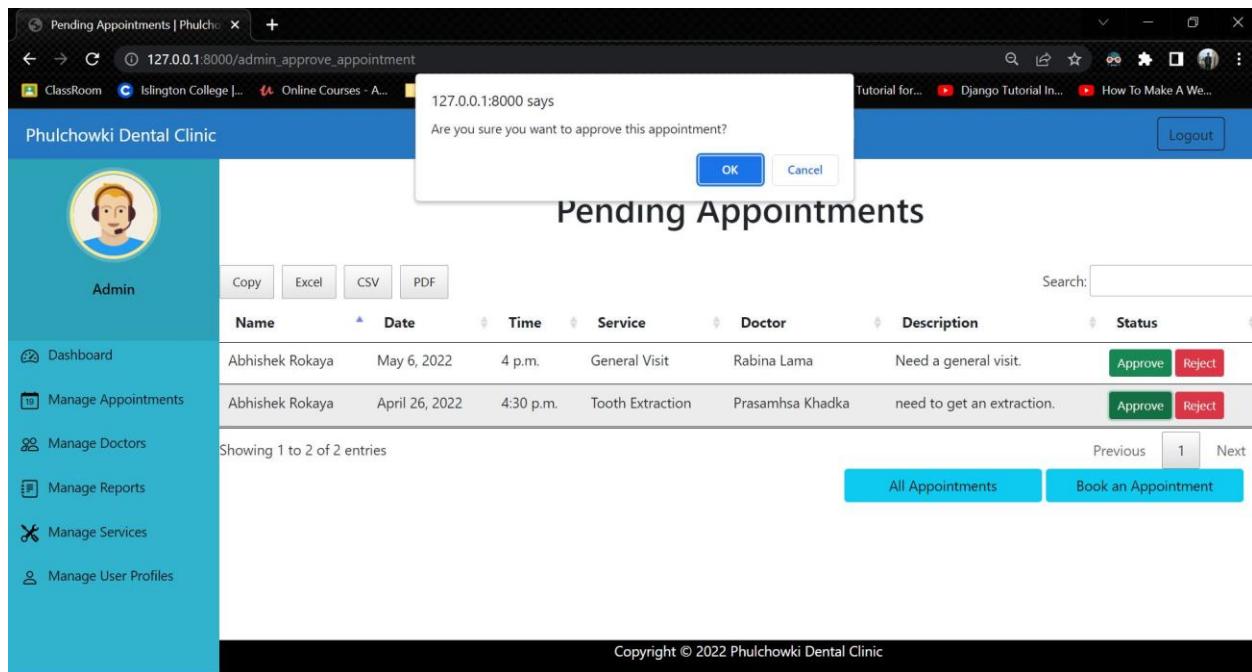


Figure 471: White Box Test 13 – 2

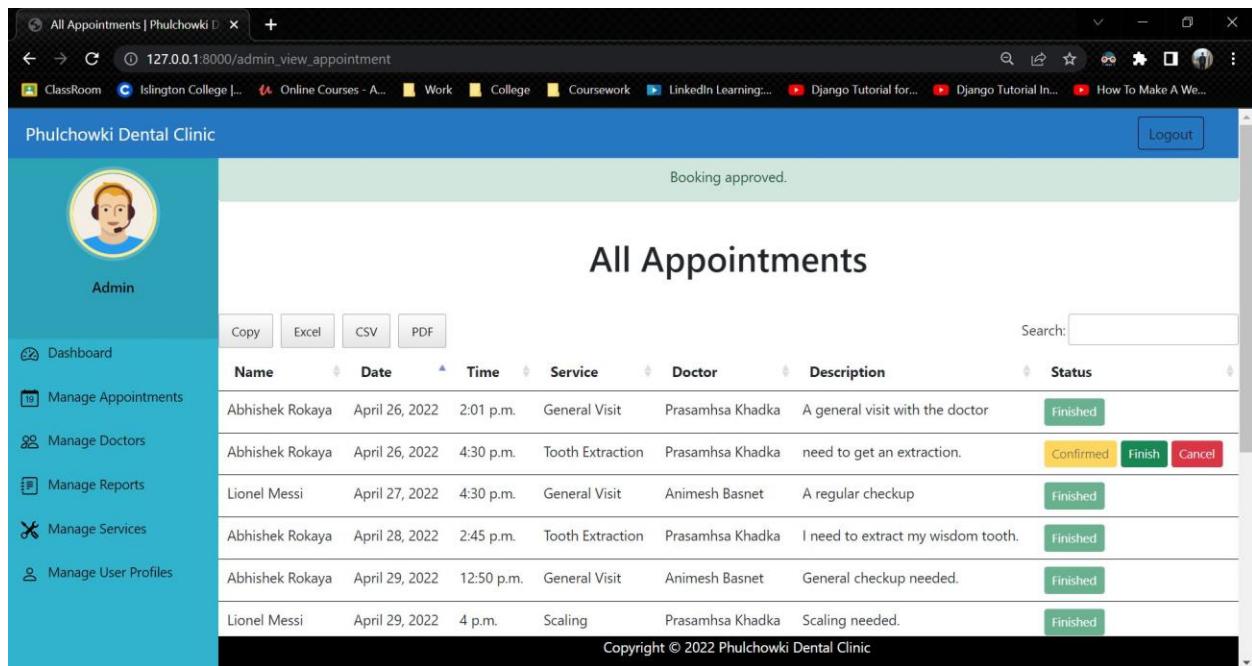


Figure 472: White Box Test 13 - 3

Test No.14	
Objective	To reject appointment of the user by the admin.
Action	The 'reject' button was clicked in the pending appointment list.
Expected Output	The appointment would be rejected.
Actual Output	The appointment was rejected.
Result	The test was successful.

Table 93: White Box Test 14

Name	Date	Time	Service	Doctor	Description	Status
Kushal Maharjan	May 7, 2022	9:30 a.m.	Scaling	Animesh Basnet	asd	<button>Approve</button> <button>Reject</button>
Kushal Maharjan	April 26, 2022	3:45 p.m.	General Visit	Deeplab Acharya	Need to get my teeth checked	<button>Approve</button> <button>Reject</button>
Kushal Maharjan	April 26, 2022	2 p.m.	Scaling	Prasamhsa Khadka	scaling needed	<button>Approve</button> <button>Reject</button>

Figure 473: White Box Test 14 - 1

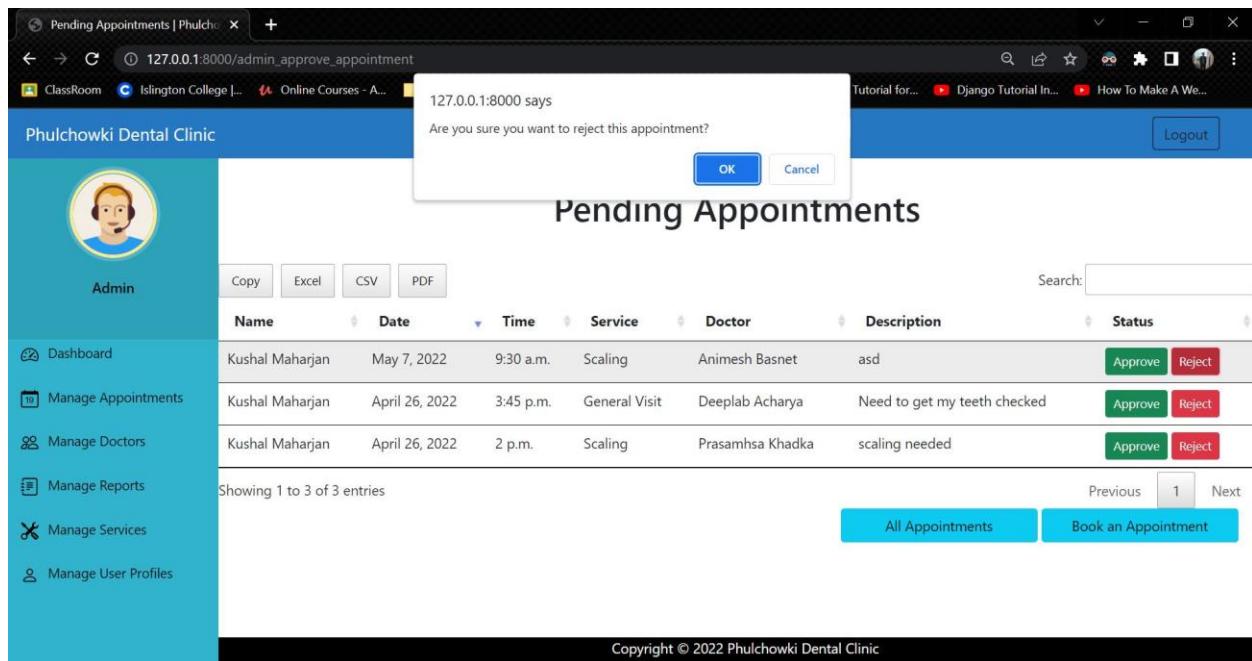


Figure 474: White Box Test 14 – 2

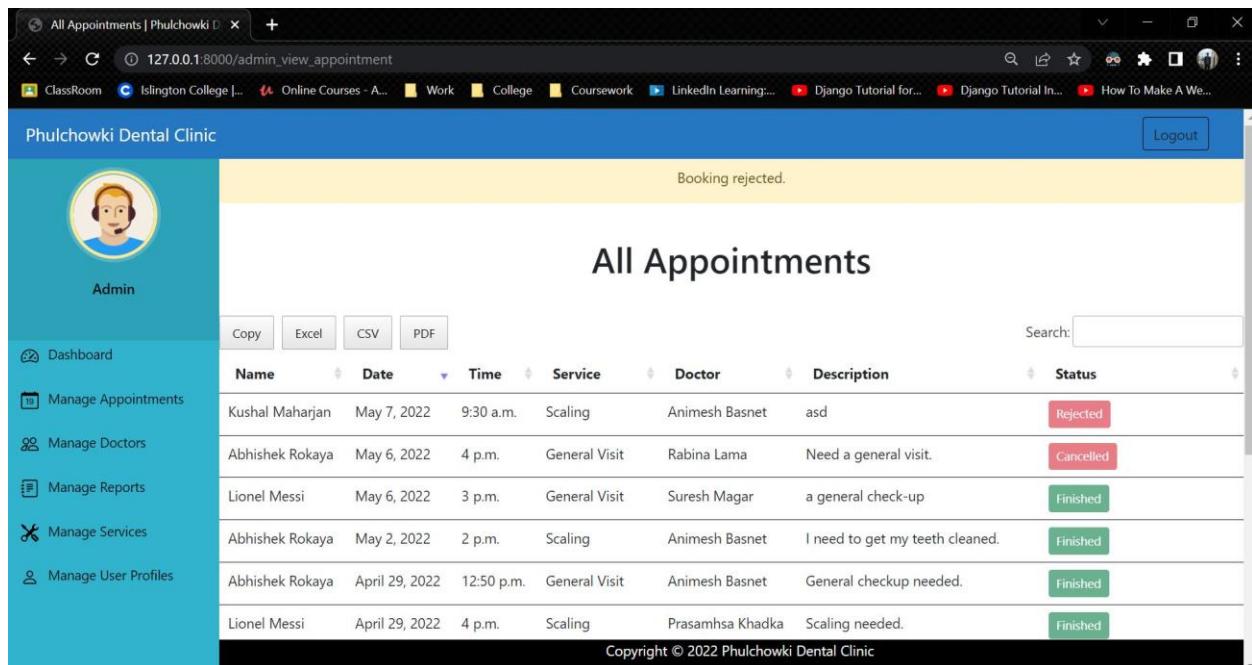


Figure 475: White Box Test 14 – 3

Test No.15	
Objective	To cancel the confirmed appointment of the user by the admin.
Action	The 'cancel' button was clicked in the appointment list.
Expected Output	The appointment would be cancelled.
Actual Output	The appointment was cancelled.
Result	The test was successful.

Table 94: White Box Test 15

Name	Date	Time	Service	Doctor	Description	Status
Abhishek Rokaya	June 8, 2022	10:30 a.m.	Scaling	Prasamhsa Khadka	scaling needed	Cancelled
Abhishek Rokaya	May 27, 2022	12:30 p.m.	Braces	Suresh Magar	Braces needed	Confirmed
Abhishek Rokaya	April 29, 2022	12:50 p.m.	General Visit	Animesh Basnet	General checkup needed.	Finished
Abhishek Rokaya	May 2, 2022	2 p.m.	Scaling	Animesh Basnet	I need to get my teeth cleaned.	Finished
Abhishek Rokaya	April 26, 2022	2:01 p.m.	General Visit	Prasamhsa Khadka	A general visit with the doctor	Finished
Abhishek Rokaya	April 28, 2022	2:45 p.m.	Tooth Extraction	Prasamhsa Khadka	I need to extract my wisdom tooth.	Finished
Lionel Messi	May 6, 2022	3 p.m.	General Visit	Suresh Magar	a general check-up	Finished

Figure 476: White Box Test 15 - 1

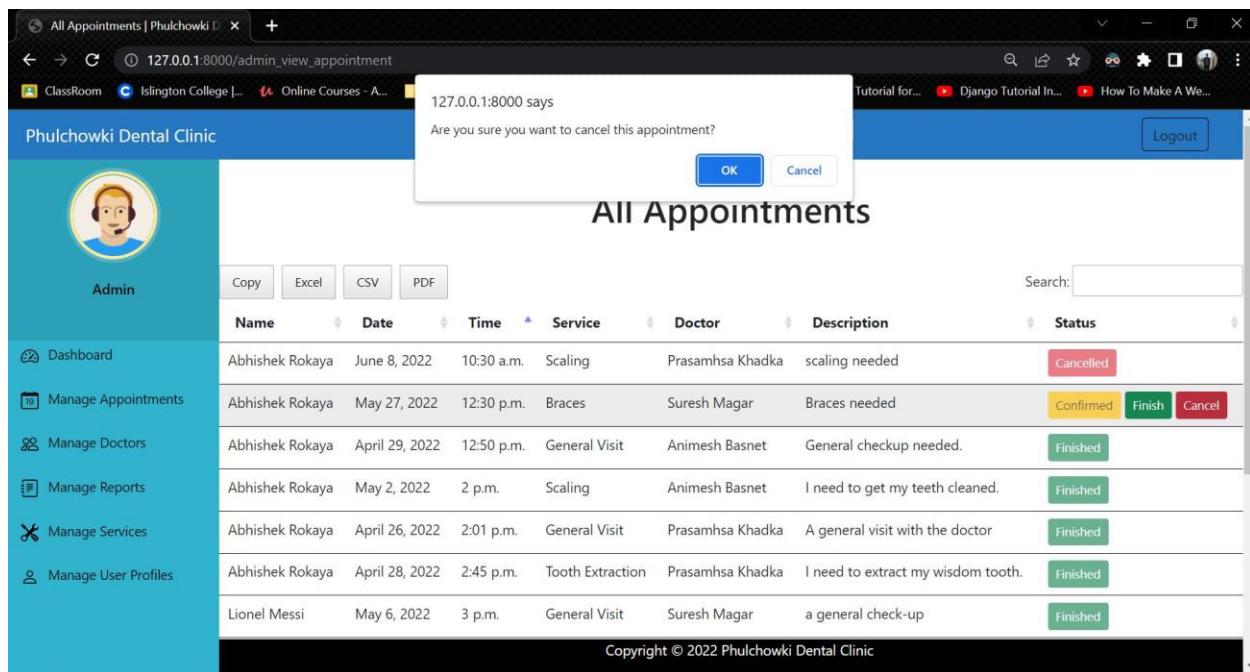


Figure 477: White Box Test 15 – 2

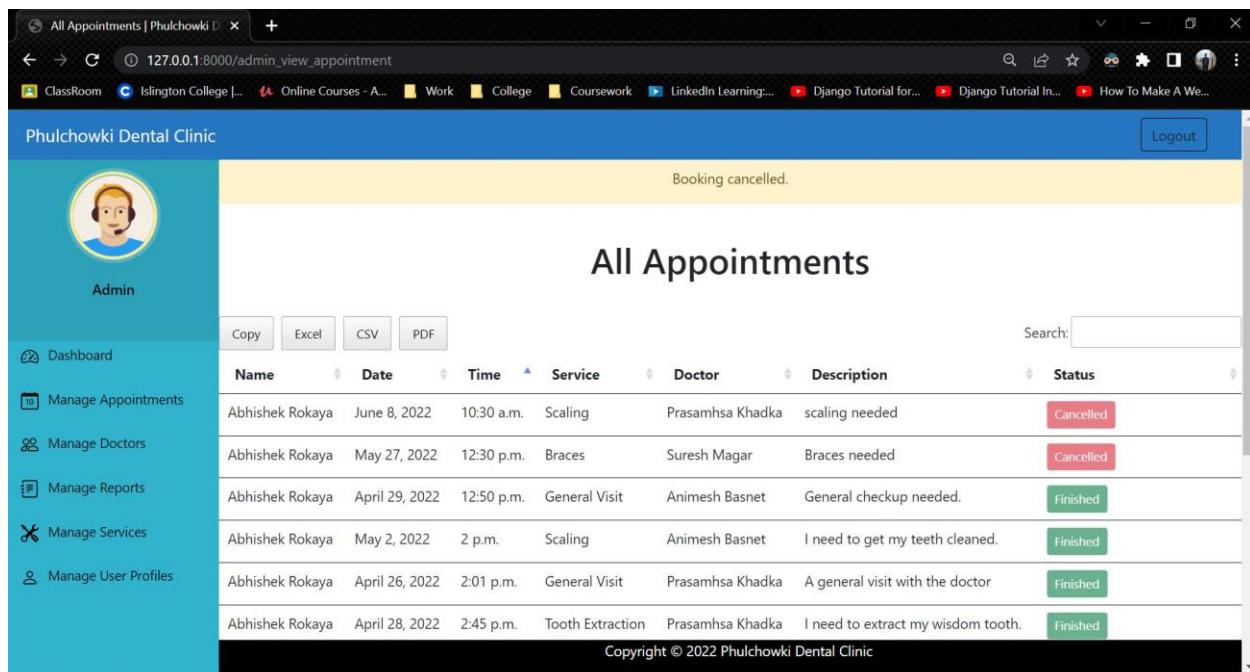


Figure 478: White Box Test 15 – 3

Test No.16	
Objective	To finish the appointment of the user by the admin.
Action	The 'finish' button was clicked in the appointment list.
Expected Output	The appointment would be finished.
Actual Output	The appointment was finished.
Result	The test was successful.

Table 95: White Box Test 16

Name	Date	Time	Service	Doctor	Description	Status
Abhishek Rokaya	April 26, 2022	4:30 p.m.	Tooth Extraction	Prasamhsa Khadka	need to get an extraction.	Confirmed Finish Cancelled
Abhishek Rokaya	April 26, 2022	2:01 p.m.	General Visit	Prasamhsa Khadka	A general visit with the doctor	Finished
Lionel Messi	April 27, 2022	4:30 p.m.	General Visit	Animesh Basnet	A regular checkup	Finished
Abhishek Rokaya	April 28, 2022	2:45 p.m.	Tooth Extraction	Prasamhsa Khadka	I need to extract my wisdom tooth.	Finished
Lionel Messi	April 29, 2022	4 p.m.	Scaling	Prasamhsa Khadka	Scaling needed.	Finished
Abhishek Rokaya	April 29, 2022	12:50 p.m.	General Visit	Animesh Basnet	General checkup needed.	Finished
Abhishek Rokaya	June 8, 2022	10:30 a.m.	Scaling	Prasamhsa Khadka	scaling needed	Cancelled

Copyright © 2022 Phulchowki Dental Clinic

Figure 479: White Box Test 16 - 1

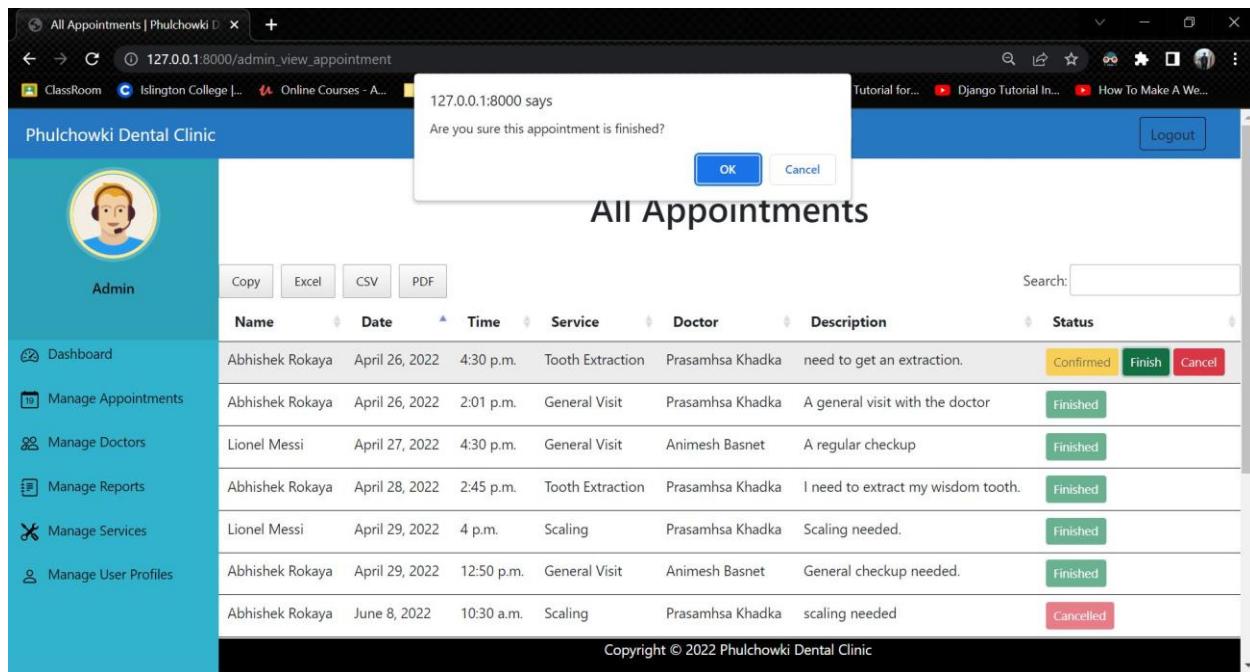


Figure 480: White Box Test 16 - 2

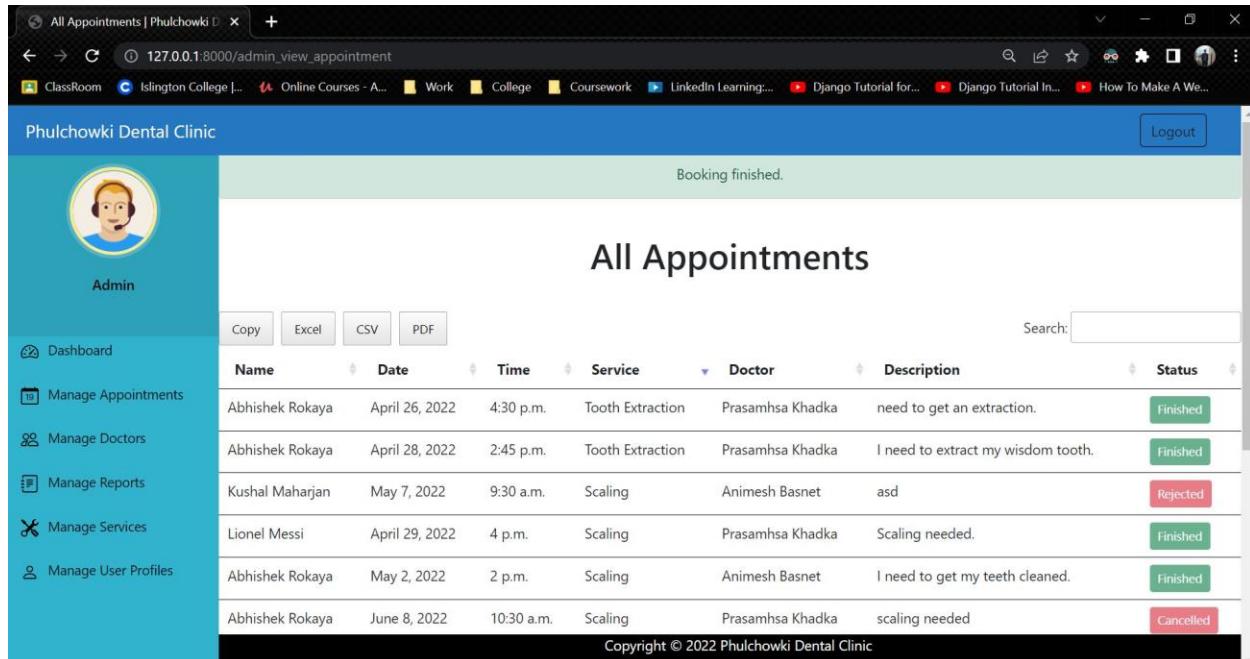


Figure 481: White Box Test 16 - 3

Test No.17	
Objective	To add another report of the same appointment.
Action	The report form was filled and the appointment which already had an report was selected and submitted.
Expected Output	An error would be shown.
Actual Output	An error was shown.
Result	The test was successful.

Table 96: White Box Test 17

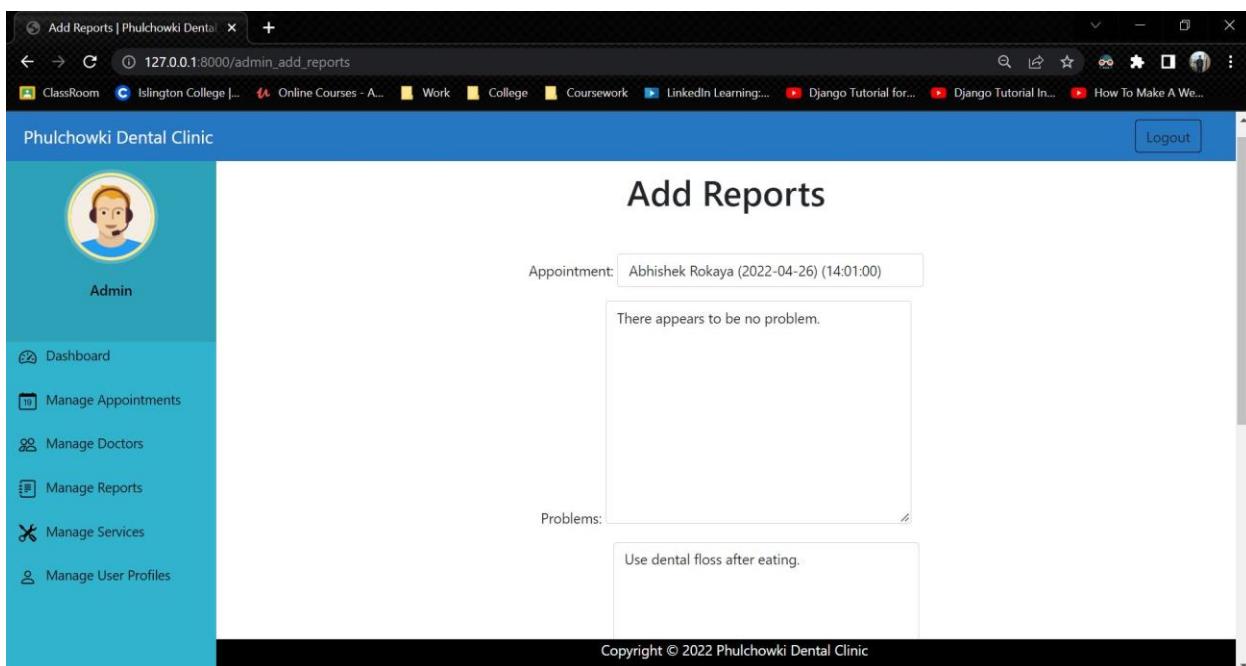


Figure 482: White Box Test 17 - 1

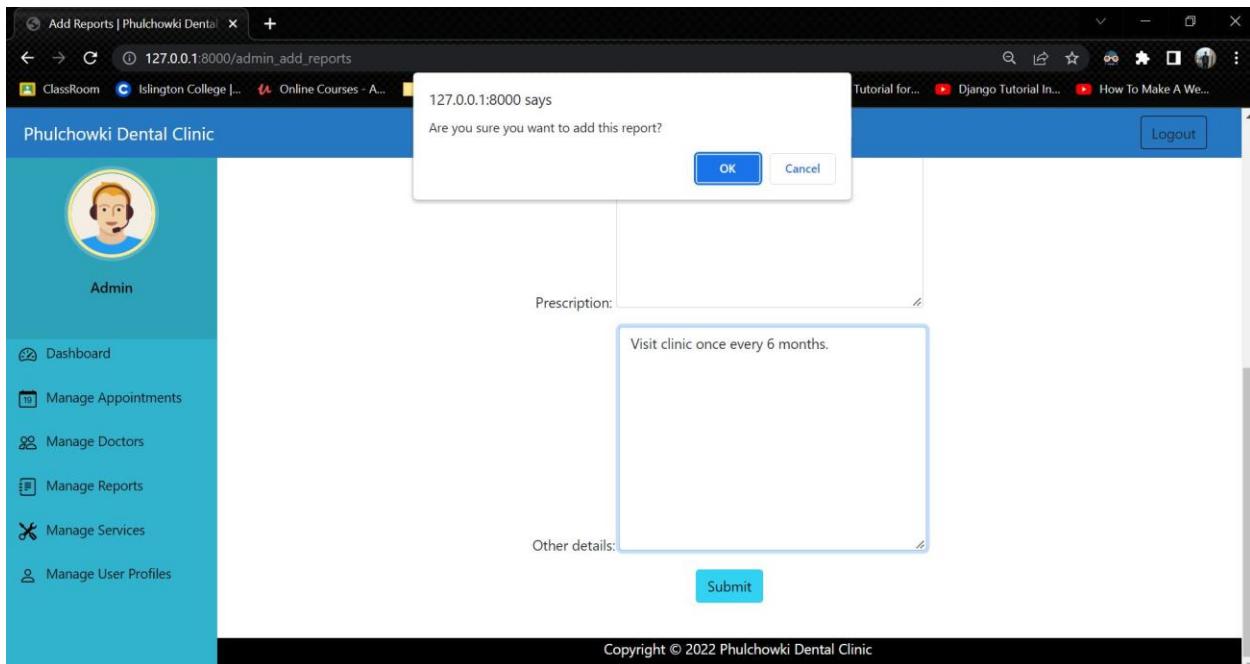


Figure 483: White Box Test 17 - 2

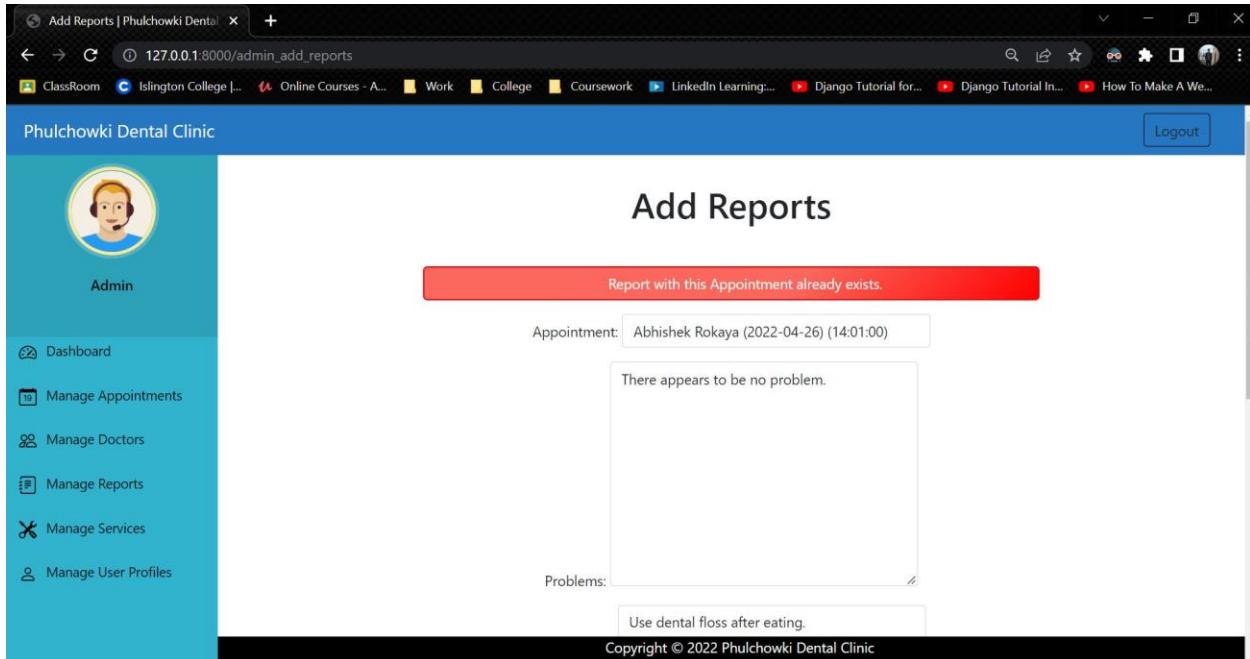


Figure 484: White Box Test 17 - 3

[View the rest of the White Box Testing Here.](#)

[View the test plan of the White Box Testing here.](#)

7.8 Appendix H: Screenshots of the System

Landing Home Page

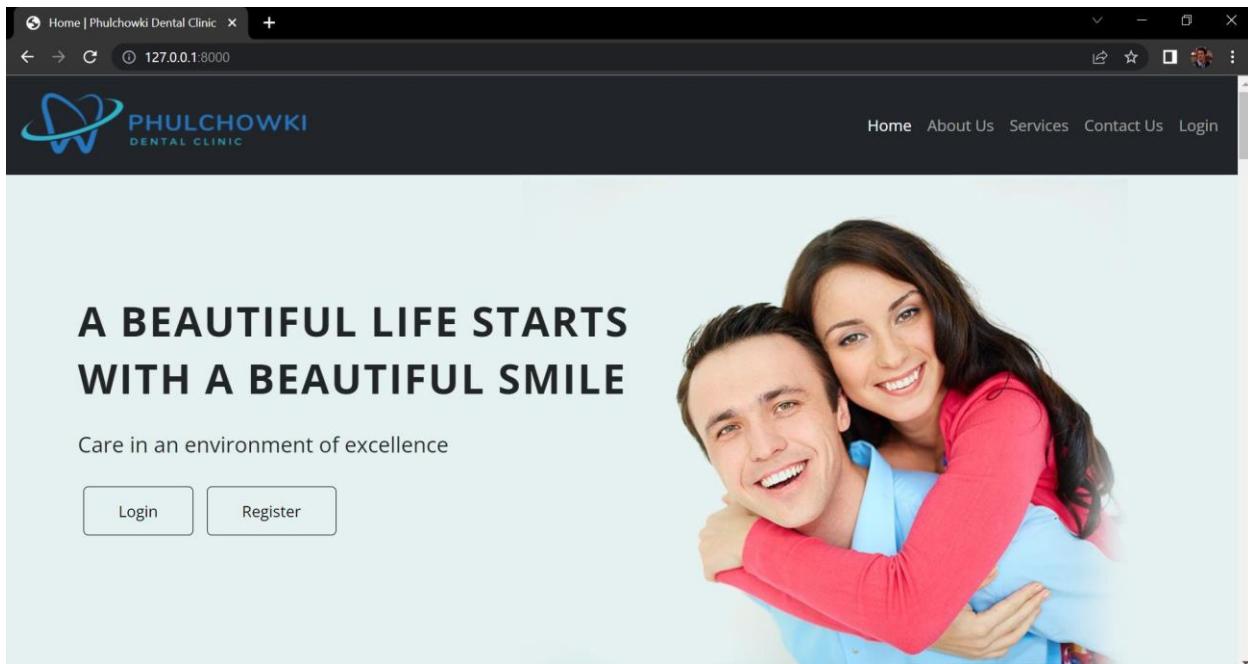


Figure 485: Screenshot of the system - 1

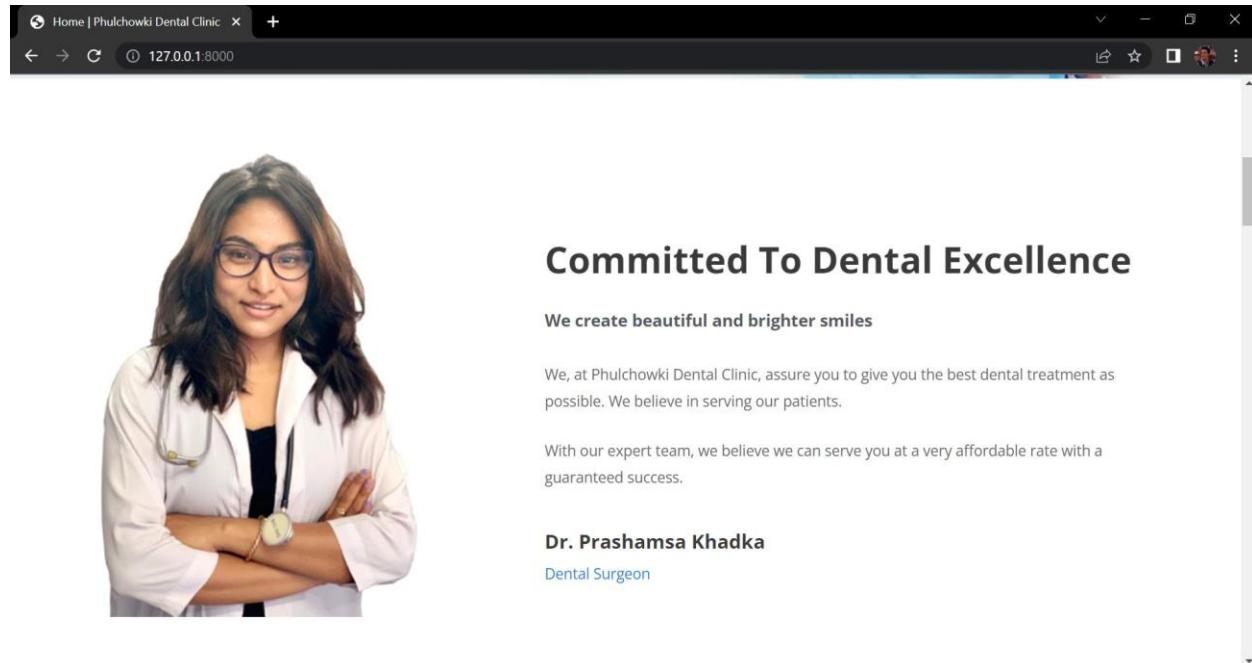


Figure 486: Screenshot of the system - 2

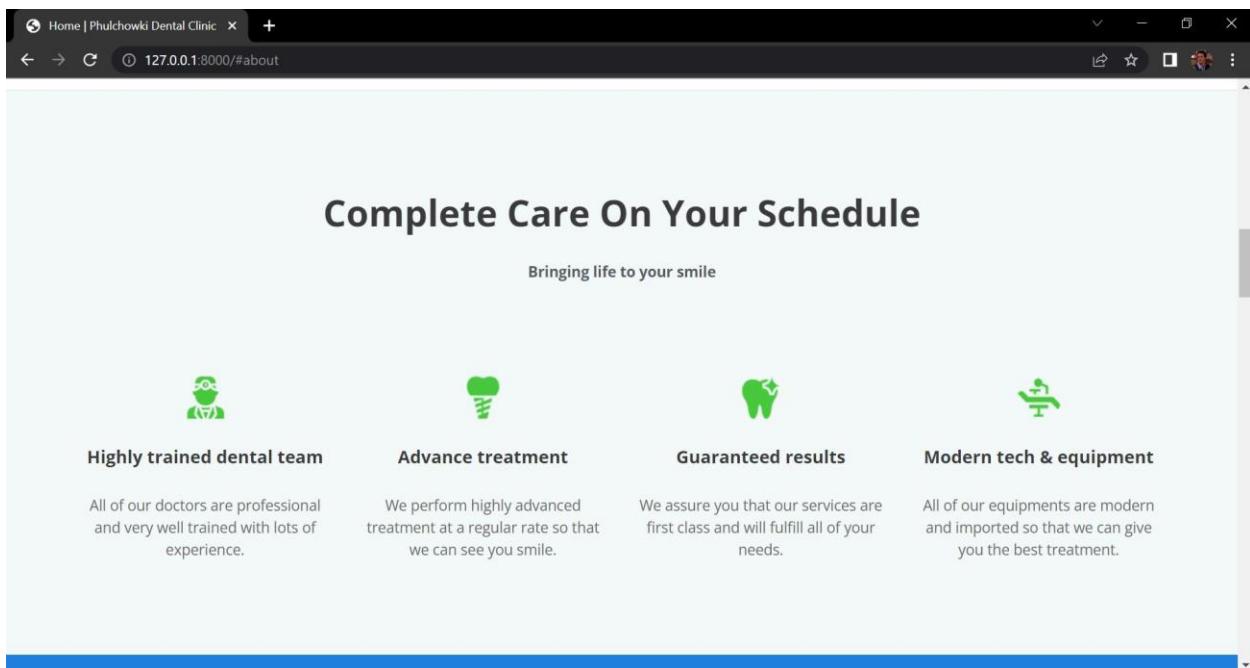


Figure 487: Screenshot of the system - 3

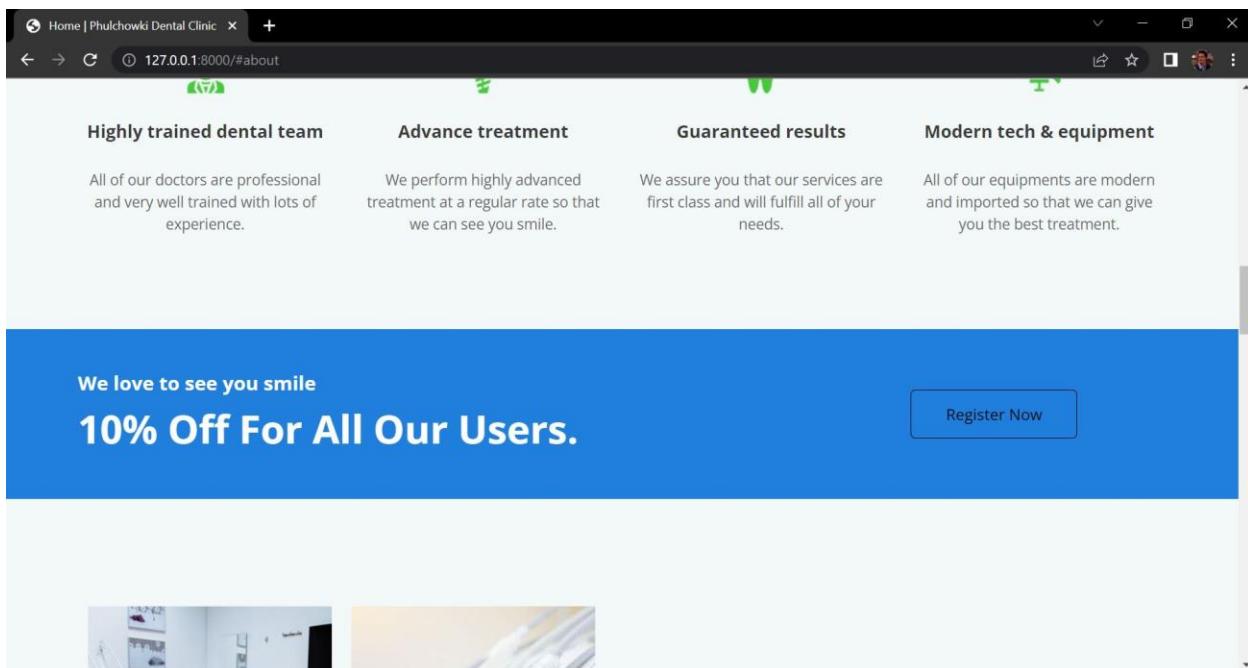


Figure 488: Screenshot of the system - 4

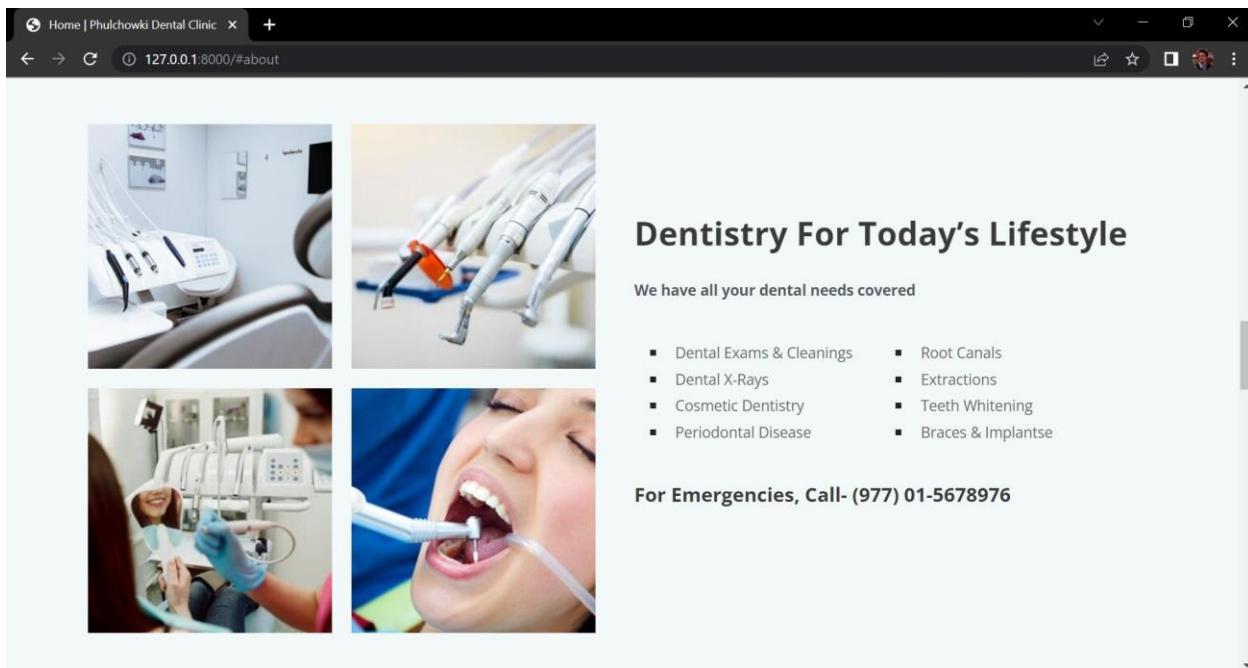


Figure 489: Screenshot of the system - 5

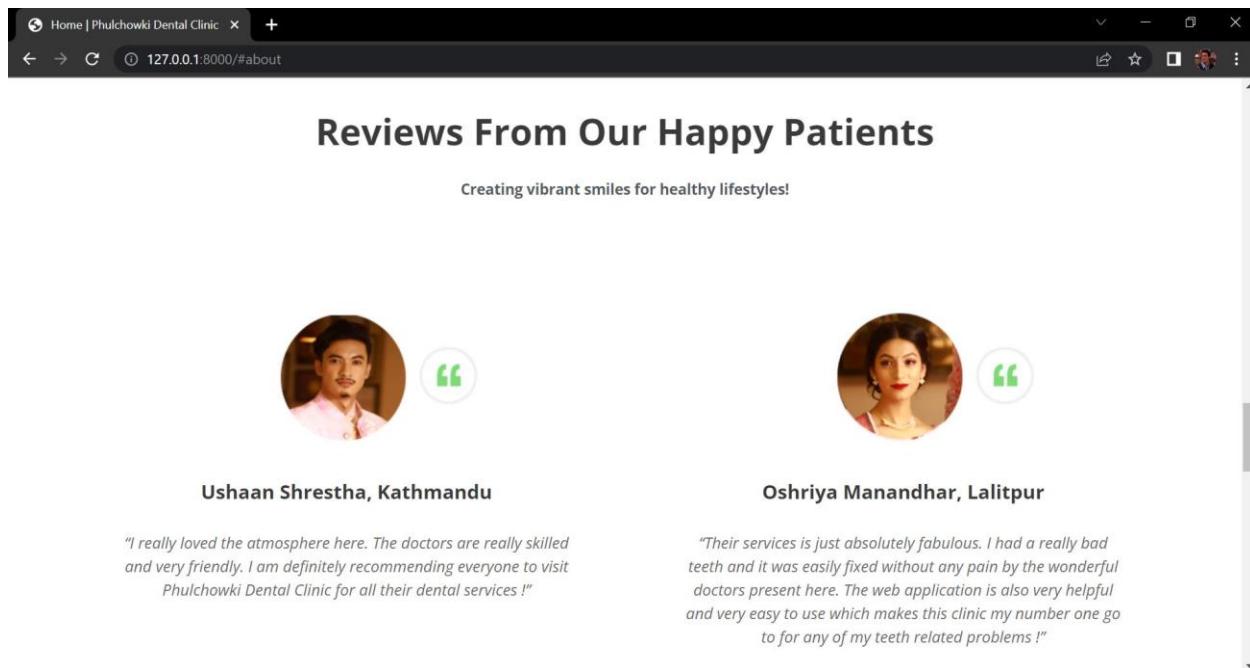


Figure 490: Screenshot of the system - 6

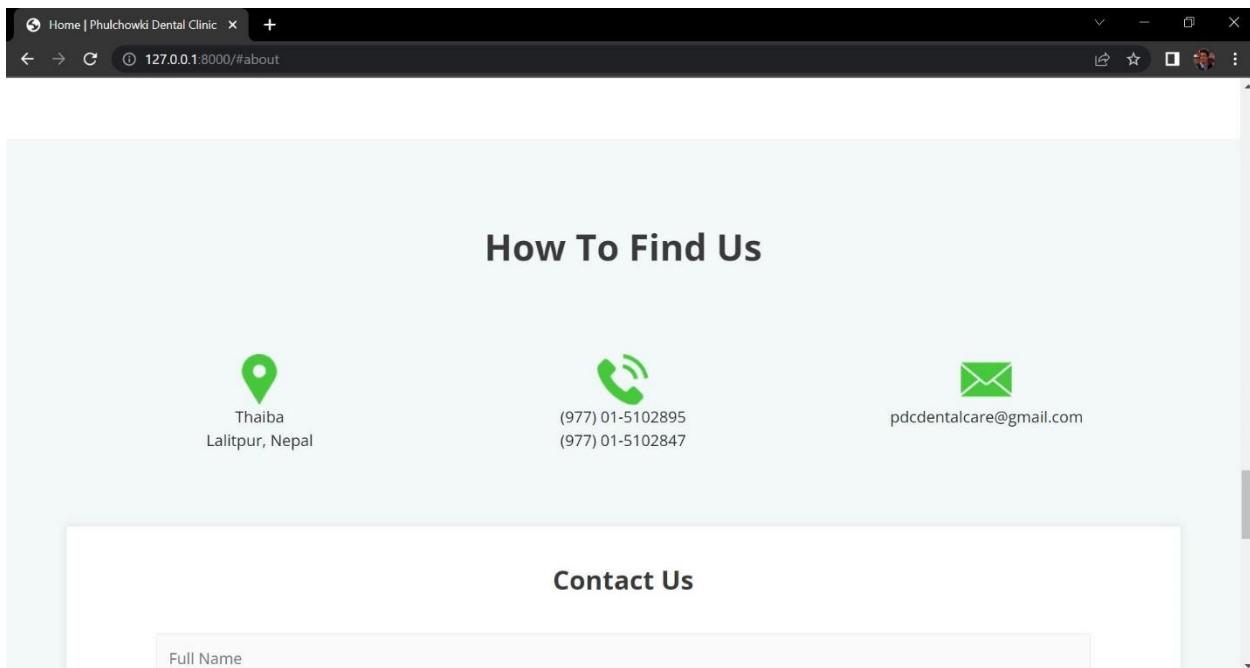


Figure 491: Screenshot of the system - 7

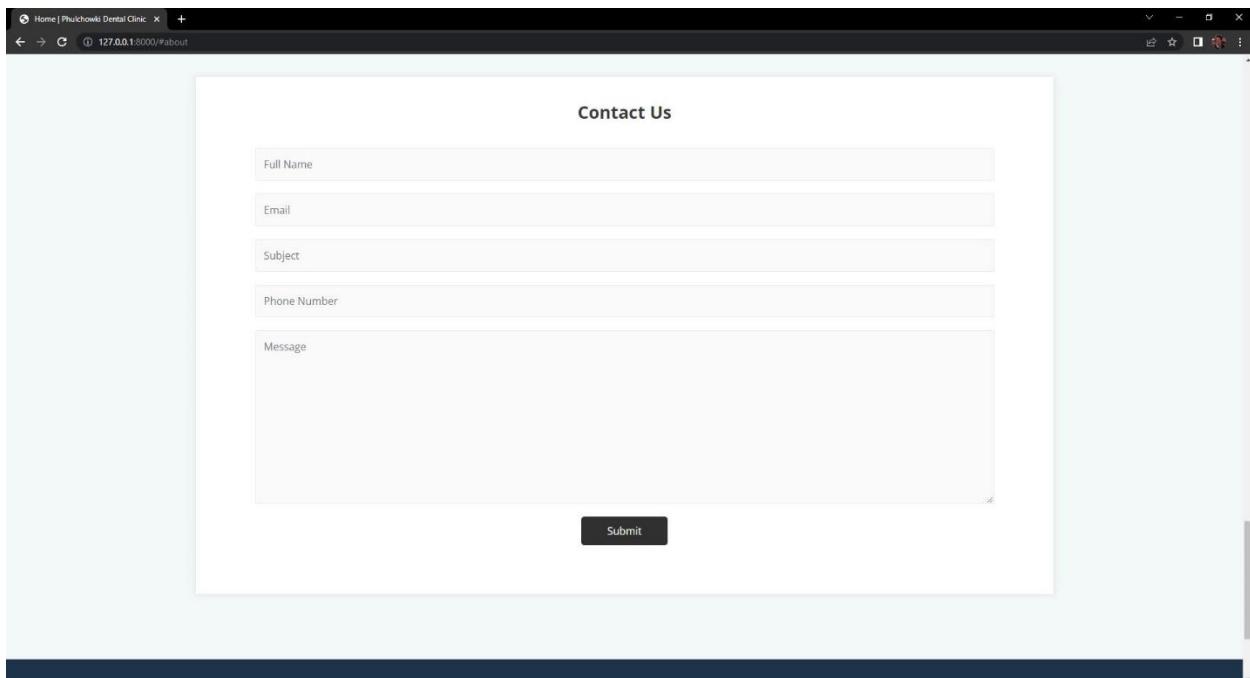


Figure 492: Screenshot of the system - 8

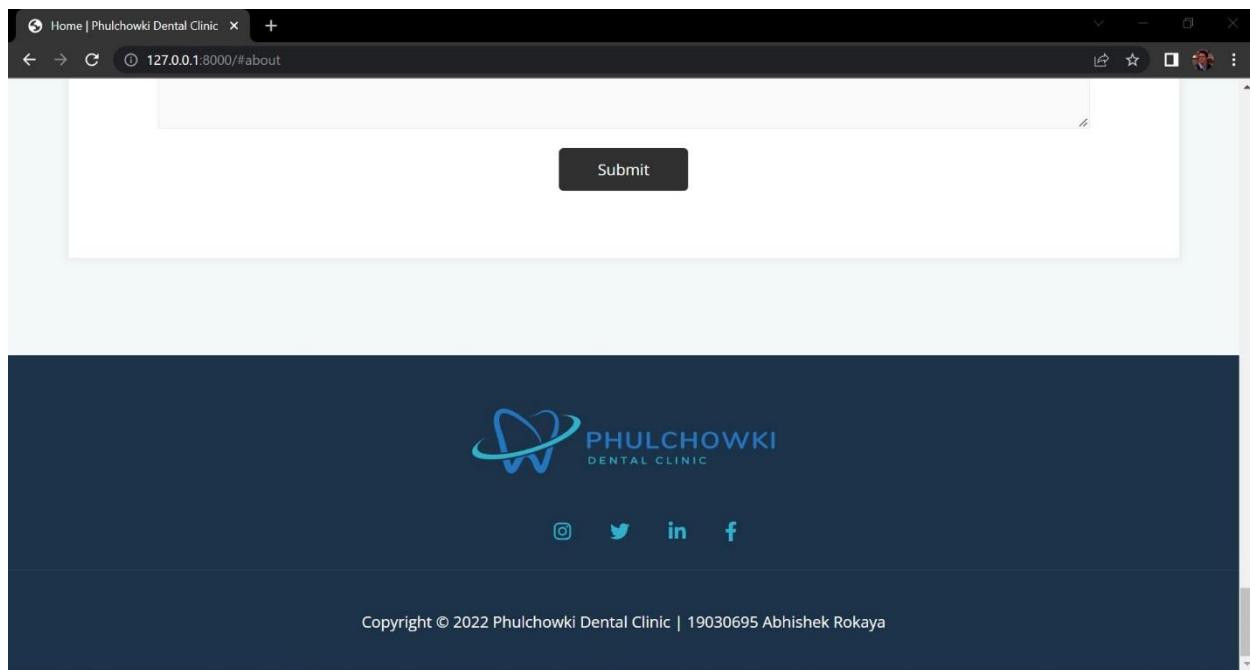


Figure 493: Screenshot of the system - 9

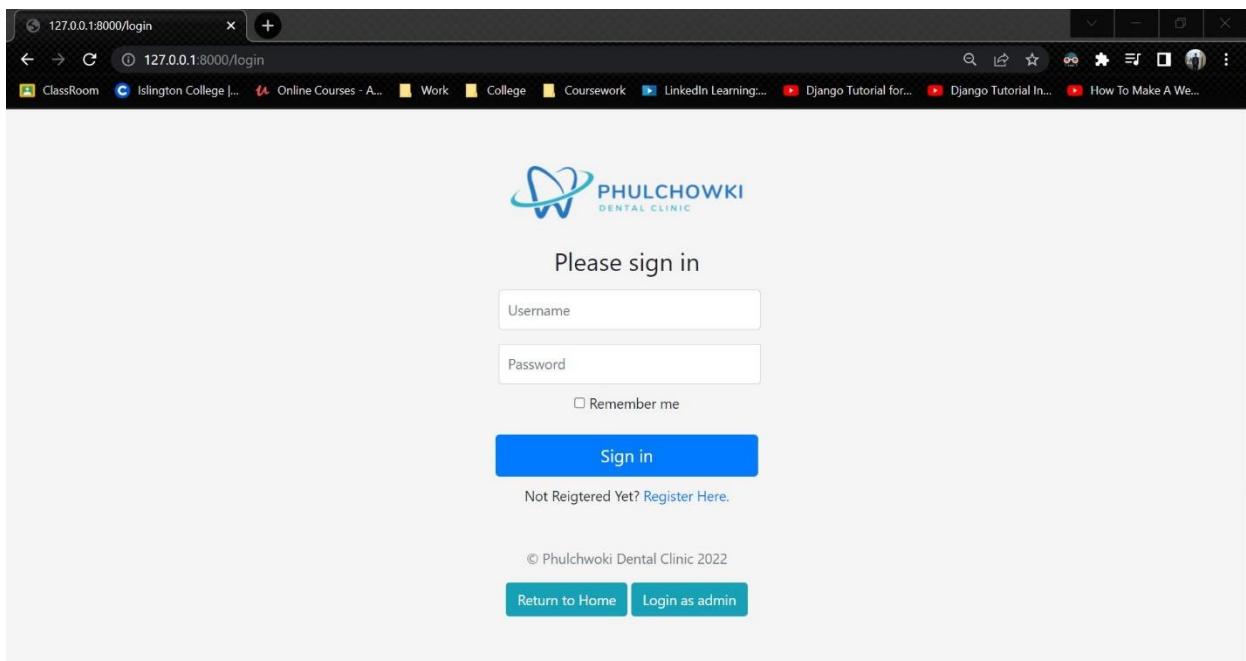


Figure 494: Screenshot of the system - 10

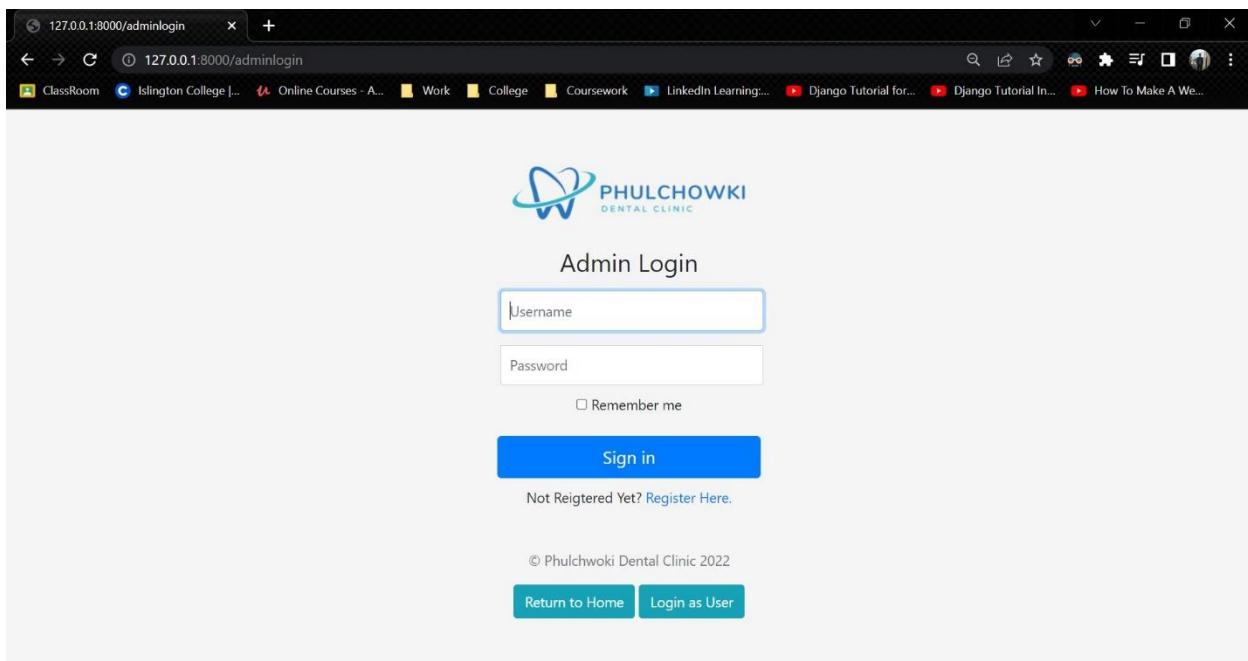


Figure 495: Screenshot of the system - 11

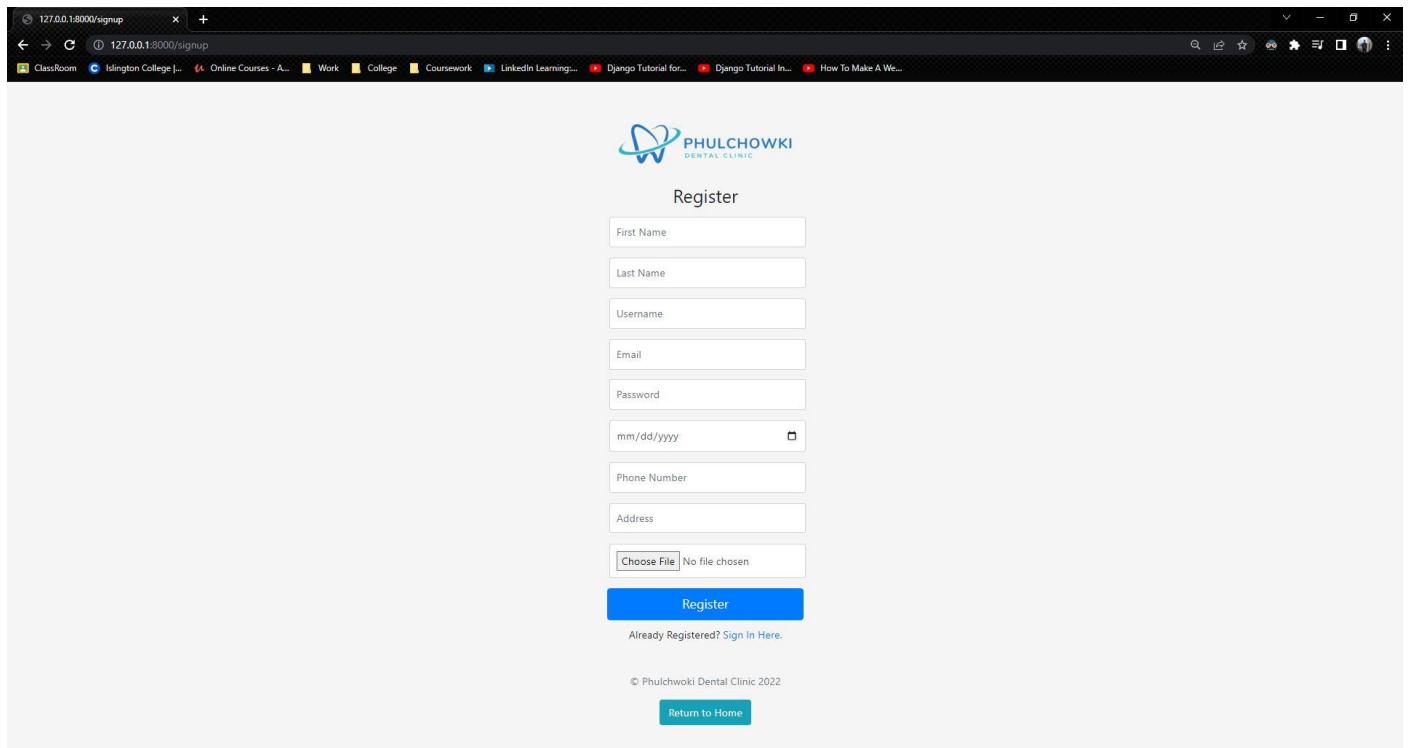


Figure 496: Screenshot of the system - 12

User Side UI

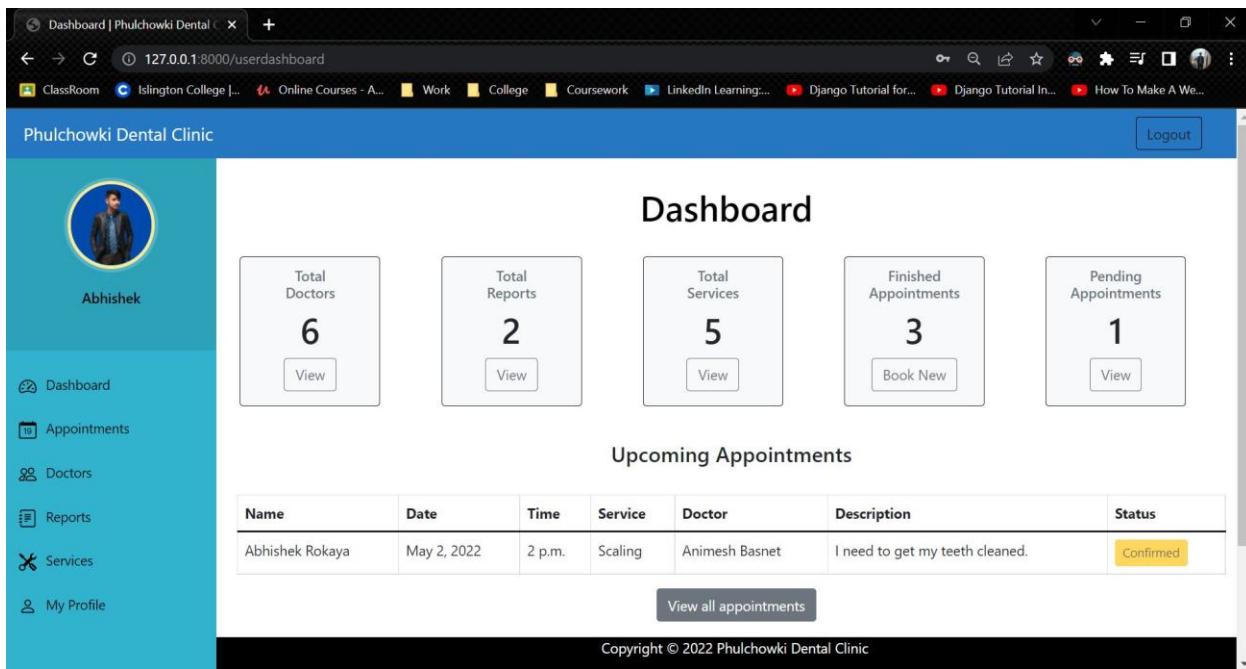


Figure 497: Screenshot of the system - 13

Your Appointments | Phulchowki

127.0.0.1:8000/user_view_appointment

Phulchowki Dental Clinic

Logout

Your Appointments

Name	Date	Time	Service	Doctor	Description	Status
Abhishek Rokaya	May 6, 2022	4 p.m.	General Visit	Rabina Lama	Need a general visit.	Pending
Abhishek Rokaya	May 2, 2022	2 p.m.	Scaling	Animesh Basnet	I need to get my teeth cleaned.	Confirmed
Abhishek Rokaya	April 29, 2022	12:50 p.m.	General Visit	Animesh Basnet	General checkup needed.	Finished
Abhishek Rokaya	April 28, 2022	2:45 p.m.	Tooth Extraction	Prasamhsa Khadka	I need to extract my wisdom tooth.	Finished
Abhishek Rokaya	April 26, 2022	2:01 p.m.	General Visit	Prasamhsa Khadka	A general visit with the doctor	Finished

Showing 1 to 5 of 5 entries

Previous 1 Next

Book new appointment

Copyright © 2022 Phulchowki Dental Clinic

Figure 498: Screenshot of the system - 14

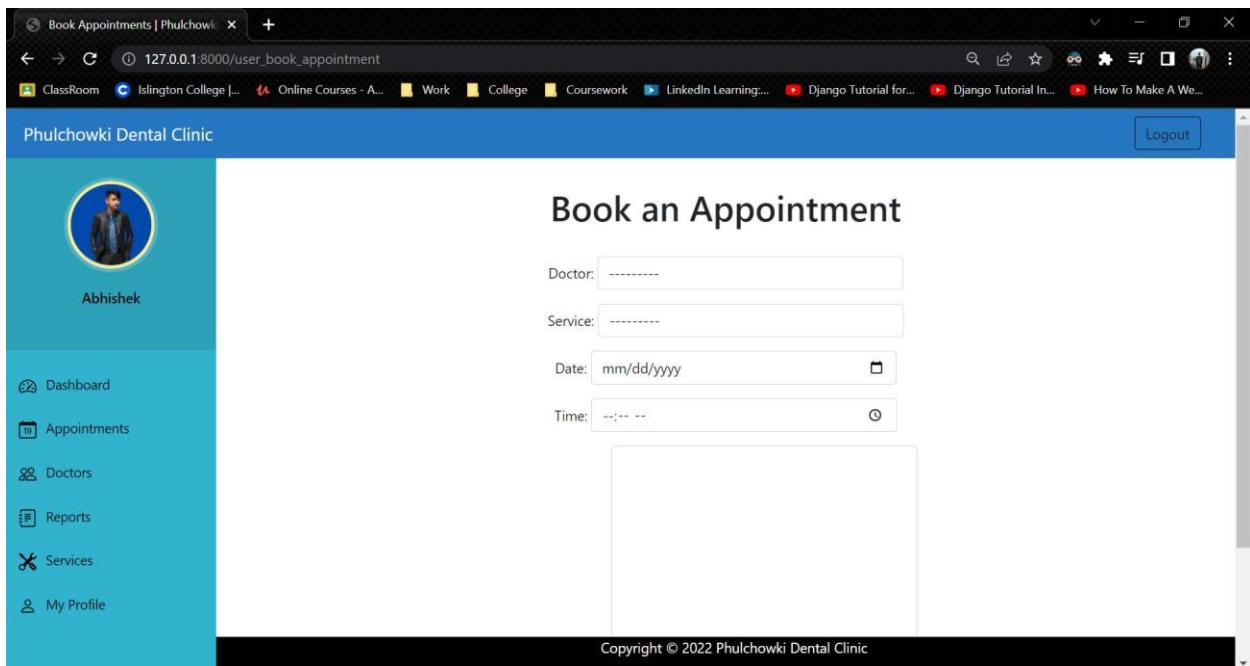


Figure 499: Screenshot of the system - 15

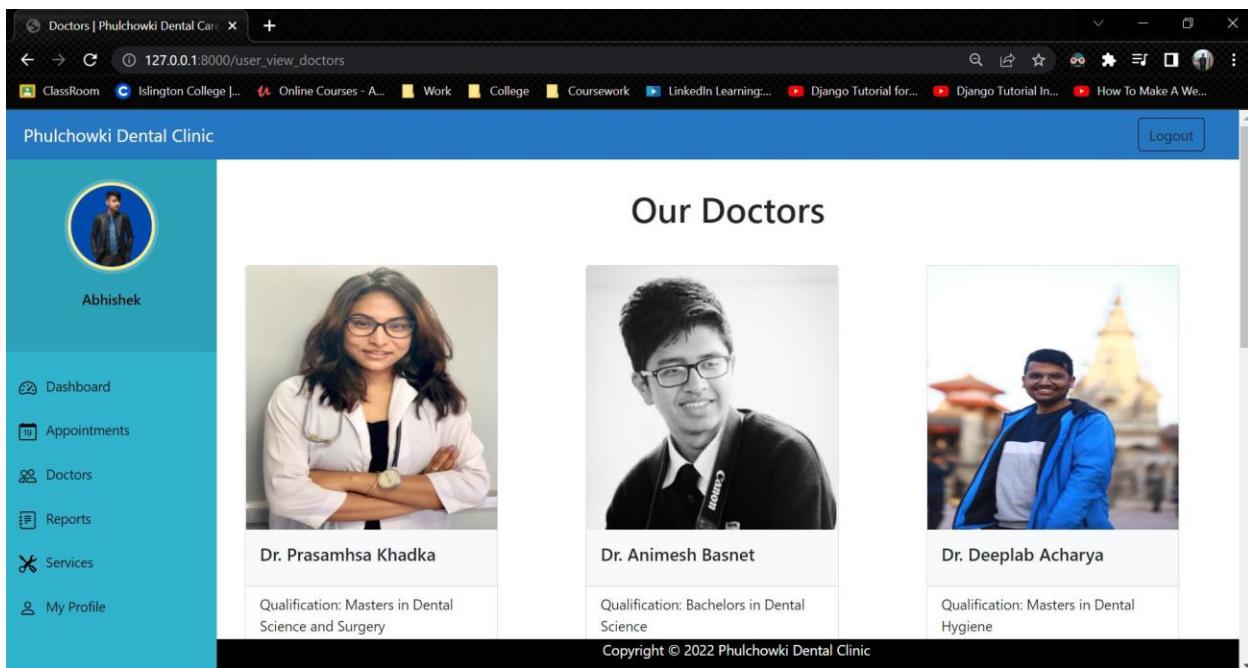


Figure 500: Screenshot of the system - 16

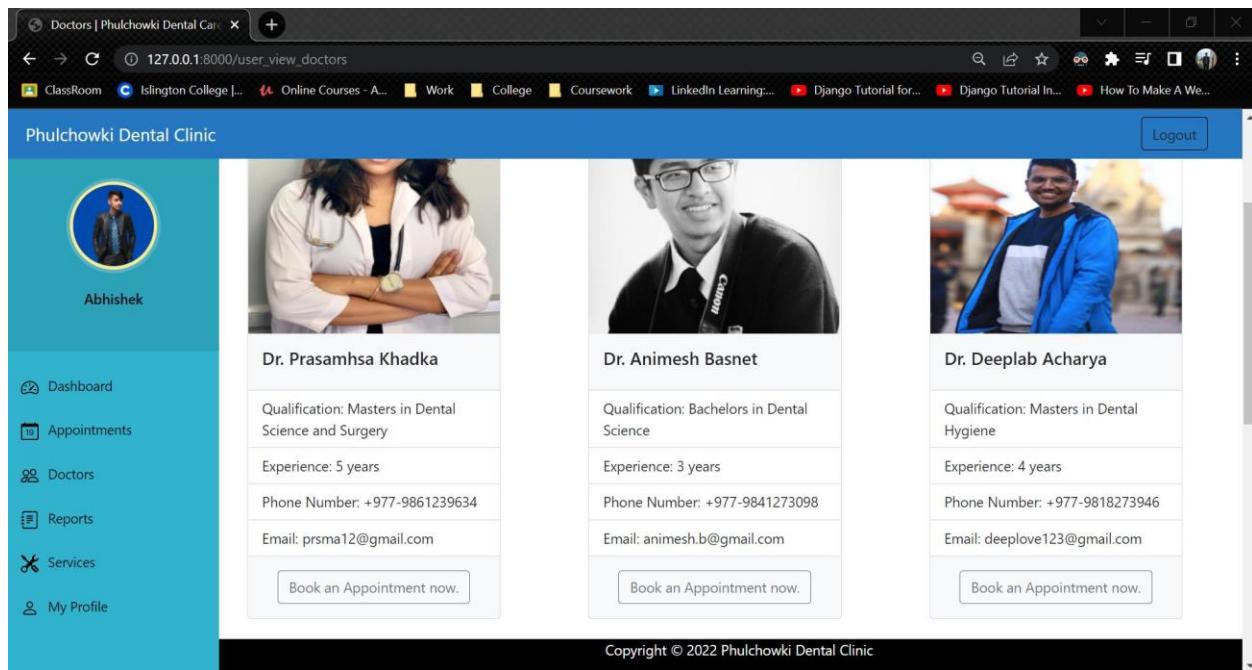


Figure 501: Screenshot of the system - 17

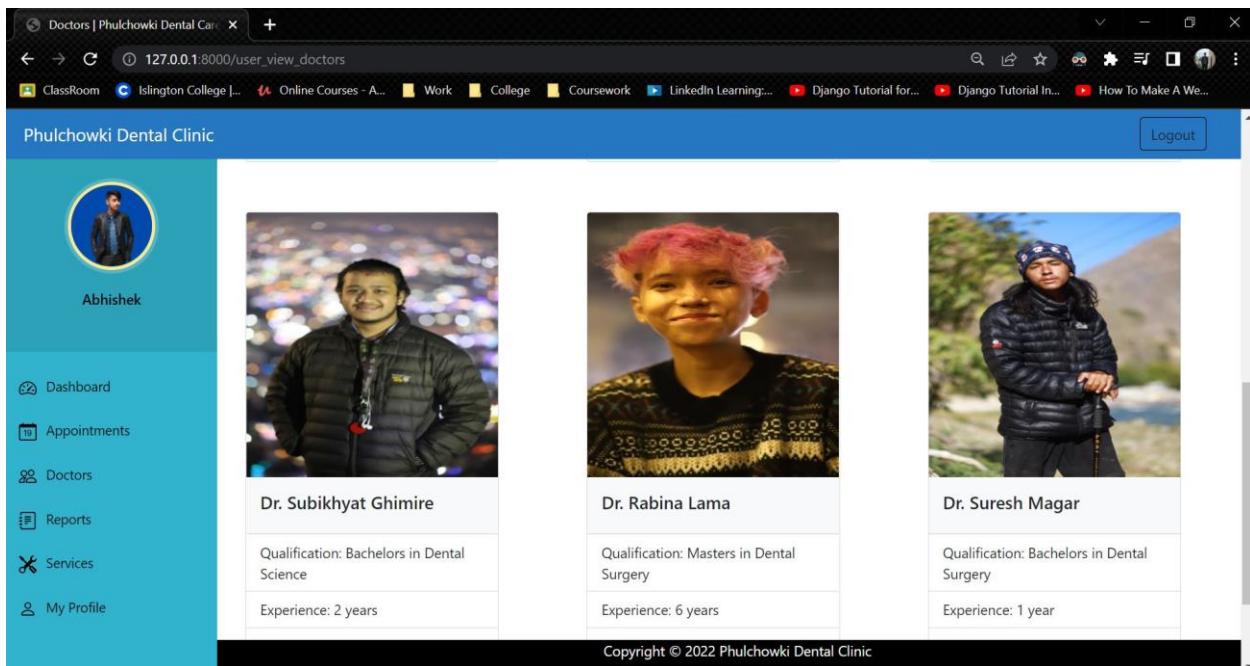


Figure 502: Screenshot of the system - 18

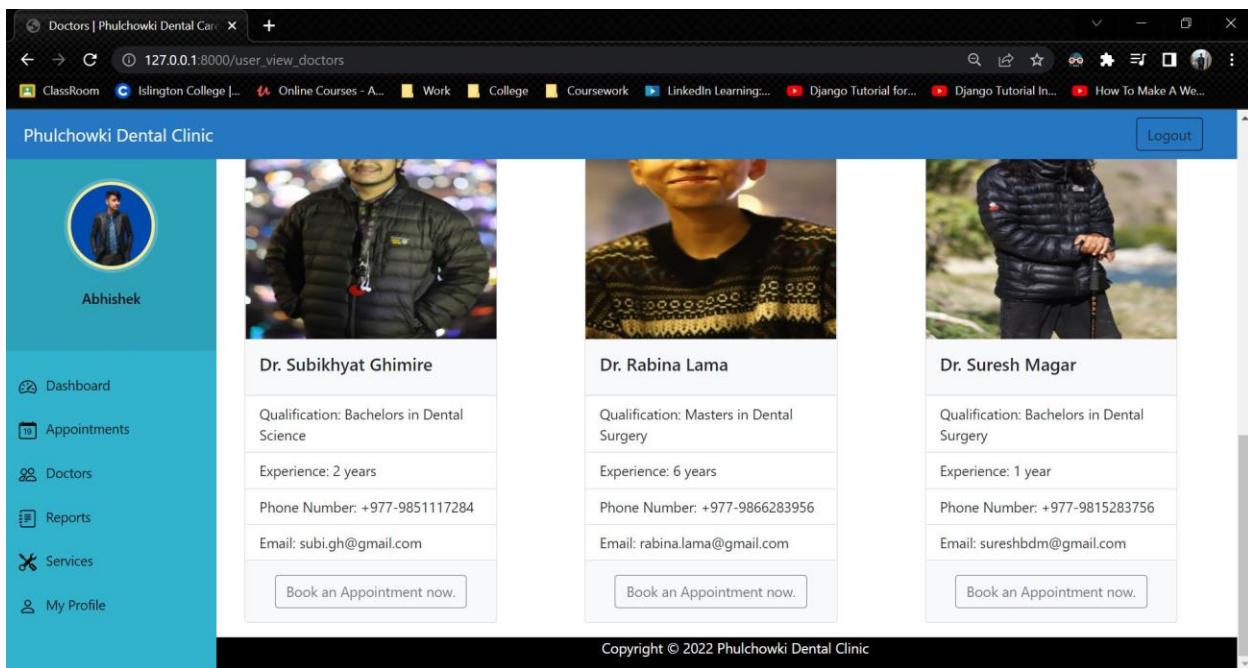


Figure 503: Screenshot of the system - 19

The screenshot shows a web-based application for a dental clinic. On the left is a sidebar with a user profile picture of Abhishek Rokaya and a list of navigation links: Dashboard, Appointments, Doctors, Reports (selected), Services, and My Profile. The main content area is titled "Your Reports". It features a table with columns: Name, Appointment Date, Doctor, Service, Problems, Prescription, Other Details, and Total Cost. Two entries are listed:

Name	Appointment Date	Doctor	Service	Problems	Prescription	Other Details	Total Cost
Abhishek Rokaya	April 28, 2022	Prasamhsa Khadka	Tooth Extraction	Wisdom tooth was causing the patient a lot of pain.	Flexon 3-times a day after food. Wisgel- 2 times a day while brushing.	If there is still a lot of pain, please contact us as soon as possible.	Rs.15000
Abhishek Rokaya	April 26, 2022	Prasamhsa Khadka	General Visit	Everything seems to be perfectly fine except in hygiene. The teeth appears to be a bit yellow.	Brush at least 2 times a day.	Scaling is recommended to clean the teeth.	Rs.350

At the bottom of the main content area, it says "Showing 1 to 2 of 2 entries". Below the table, there are navigation buttons for "Previous", "1" (highlighted), and "Next". The footer contains the copyright notice "Copyright © 2022 Phulchowki Dental Clinic".

Figure 504: Screenshot of the system – 20

The screenshot shows a web application interface for a dental clinic. On the left is a sidebar with a user profile picture and name 'Abhishek'. Below the profile are links for Dashboard, Appointments, Doctors, Reports, Services (which is the active tab), and My Profile. The main content area has a title 'Our Services' and a search bar. A table lists five dental services with columns for Name, Cost, and Description. At the bottom, it shows 'Showing 1 to 5 of 5 entries' and navigation buttons for Previous, Next, and page number 1.

Name	Cost	Description
Braces	28000	We do all kinds of braces implantation to align your teeth. The base price starts from Rs.28000.
General Visit	350	We do a thorough check up on this and recommend necessary treatments.
RCT	4500	Root Canal Treatment (RCT) is a procedure done when the tooth decay reaches the pulp (innermost layer of the tooth) causing inflammation.
Scaling	1500	Scaling is the cleaning of the teeth.
Tooth Extraction	15000	Tooth extraction is done when the teeth needs to be removed. The costs started from 15000 and can be raised according to the complexity.

Showing 1 to 5 of 5 entries

Copyright © 2022 Phulchowki Dental Clinic

Figure 505: Screenshot of the system - 21

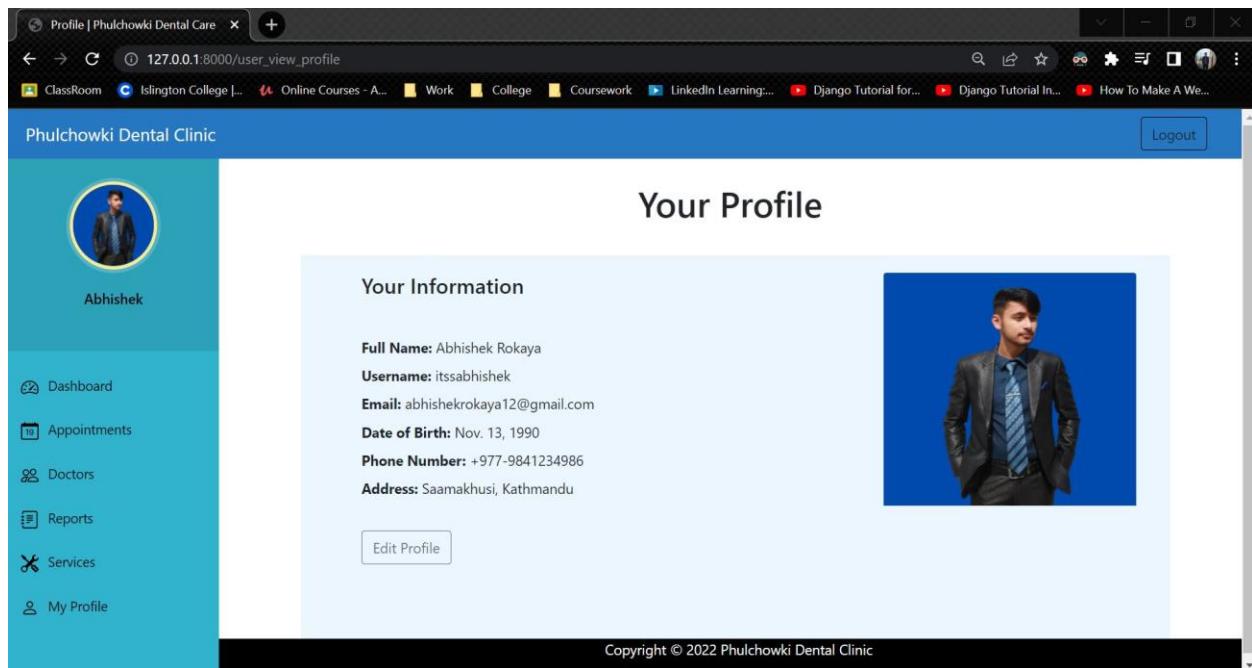


Figure 506: Screenshot of the system - 22

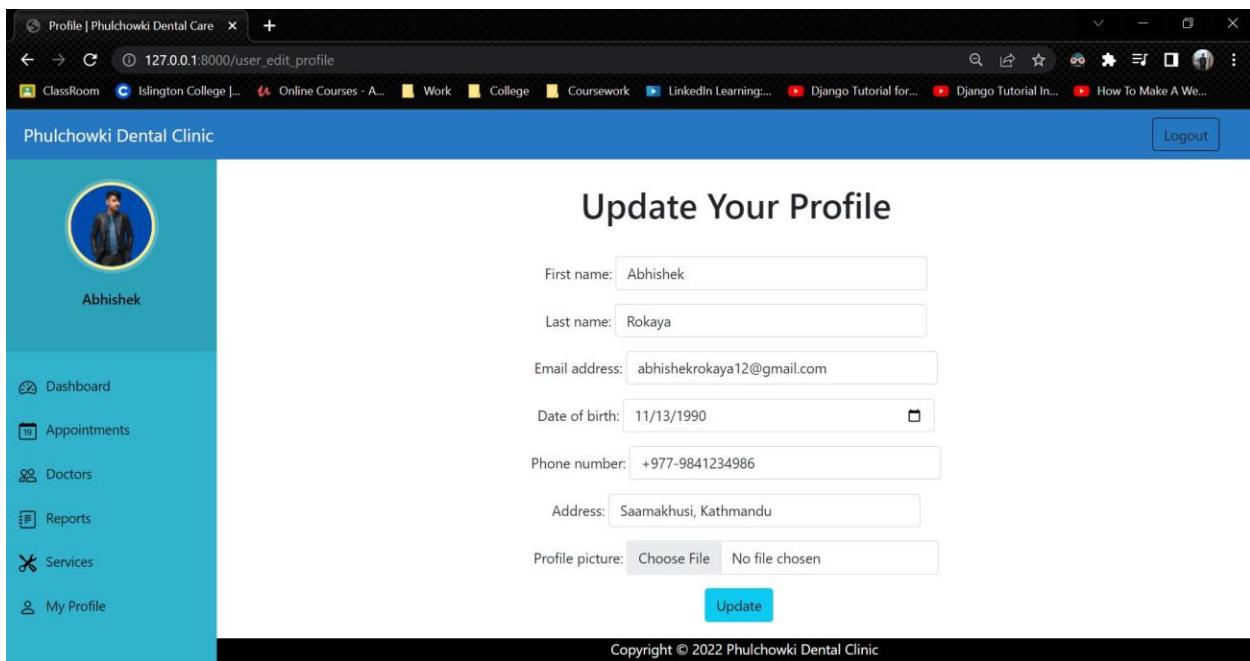


Figure 507: Screenshot of the system - 23

Admin Side

The screenshot shows the Admin Dashboard for Phulchowki Dental Clinic. On the left, a sidebar menu includes: Admin (profile icon), Dashboard, Manage Appointments, Manage Doctors, Manage Reports, Manage Services, and Manage User Profiles. The main content area has a title 'Dashboard' and five summary boxes: Total Doctors (6, View), Total Users (3, View), Total Services (5, View), Total Appointments (6, View), and Pending Appointments (1, View). Below this is a section titled 'Upcoming Appointments' with a table:

Name	Date	Time	Service	Doctor	Description	Status
Lionel Messi	April 29, 2022	4 p.m.	Scaling	Prasamhsa Khadka	Scaling needed.	Confirmed
Abhishek Rokaya	May 2, 2022	2 p.m.	Scaling	Animesh Basnet	I need to get my teeth cleaned.	Confirmed

At the bottom, there is a 'View all appointments' button and a copyright notice: Copyright © 2022 Phulchowki Dental Clinic.

Figure 508: Screenshot of the system - 24

The screenshot shows a web-based administrative interface for a dental clinic. The title bar indicates the page is 'Pending Appointments | Phulchowki' and the URL is '127.0.0.1:8000/admin_approve_appointment'. The header features a logo of a person with a headset, the text 'Phulchowki Dental Clinic', and a 'Logout' button. On the left, a sidebar titled 'Admin' lists navigation options: Dashboard, Manage Appointments, Manage Doctors, Manage Reports, Manage Services, and Manage User Profiles. The main content area is titled 'Pending Appointments' and displays a table of two entries:

	Name	Date	Time	Service	Doctor	Description	Status
1	Abhishek Rokaya	May 6, 2022	4 p.m.	General Visit	Rabina Lama	Need a general visit.	<button>Approve</button> <button>Reject</button>
2	Lionel Messi	May 6, 2022	3 p.m.	General Visit	Suresh Magar	a general check-up	<button>Approve</button> <button>Reject</button>

Below the table, it says 'Showing 1 to 2 of 2 entries'. At the bottom right are buttons for 'All Appointments' and 'Book an Appointment'. A copyright notice at the bottom states 'Copyright © 2022 Phulchowki Dental Clinic'.

Figure 509: Screenshot of the system - 25

The screenshot shows a web-based application interface for a dental clinic. On the left, there is a sidebar with a user icon and the title "Admin". Below the title are several menu items: "Dashboard", "Manage Appointments", "Manage Doctors", "Manage Reports", "Manage Services", and "Manage User Profiles". The main content area is titled "All Appointments". At the top of this section, there are four buttons: "Copy", "Excel", "CSV", and "PDF". To the right is a search bar labeled "Search:" with a placeholder text field. Below these controls is a table with the following columns: Name, Date, Time, Service, Doctor, Description, and Status. The table contains seven rows of appointment data. At the bottom of the main content area, there is a copyright notice: "Copyright © 2022 Phulchowki Dental Clinic".

	Name	Date	Time	Service	Doctor	Description	Status
Dashboard	Amrit Nepali	May 7, 2022	3 p.m.	Tooth Extraction	Rabina Lama	tooth	Rejected
Manage Appointments	Lionel Messi	May 6, 2022	3 p.m.	General Visit	Suresh Magar	a general check-up	Finished
Manage Doctors	Abhishek Rokaya	May 2, 2022	2 p.m.	Scaling	Animesh Basnet	I need to get my teeth cleaned.	Confirmed
Manage Reports	Abhishek Rokaya	April 29, 2022	12:50 p.m.	General Visit	Animesh Basnet	General checkup needed.	Finished
Manage Services	Lionel Messi	April 29, 2022	4 p.m.	Scaling	Prasamhsa Khadka	Scaling needed.	Finished
Manage User Profiles	Abhishek Rokaya	April 28, 2022	2:45 p.m.	Tooth Extraction	Prasamhsa Khadka	I need to extract my wisdom tooth.	Finished
	Lionel Messi	April 27, 2022	4:30 p.m.	General Visit	Animesh Basnet	A regular checkup	Finished

Figure 510: Screenshot of the system - 26

Name	Appointment Date	Doctor	Service	Problems	Prescription	Other Details	Total Cost	Update	Delete
Abhishek Rokaya	April 28, 2022	Prasamhsa Khadka	Tooth Extraction	Wisdom tooth was causing the patient a lot of pain.	Flexon 3-times a day after food. Wisgel- 2 times a day while brushing.	If there is still a lot of pain, please contact us as soon as possible.	Rs.15000	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Abhishek Rokaya	April 26, 2022	Prasamhsa Khadka	General Visit	Everything seems to be perfectly fine except in hygiene. The teeth appears to be a bit yellow.	Brush at least 2 times a day.	Scaling is recommended to clean the teeth.	Rs.350	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Lionel Messi	April 27, 2022	Animesh Basnet	General Visit	There is a cavity in the upper right canal teeth.	The teeth needs to be filled. For now, please brush at least 2 times a day.	Avoid drinking cold drinks.	Rs.350	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Copyright © 2022 Phulchowki Dental Clinic

Figure 511: Screenshot of the system - 27

The screenshot shows a web-based administrative interface for a dental clinic. The left sidebar, titled 'Admin', contains links for Dashboard, Manage Appointments, Manage Doctors, Manage Reports, Manage Services (which is currently selected), and Manage User Profiles. The main content area is titled 'All Services' and displays a table of five services:

Name	Cost	Description	Update	Delete
Braces	Rs.28000	We do all kinds of braces implantation to align your teeth. The base price starts from Rs.28000.	<input type="checkbox"/>	<input type="checkbox"/>
General Visit	Rs.350	We do a thorough check up on this and recommend necessary treatments.	<input type="checkbox"/>	<input type="checkbox"/>
RCT	Rs.4500	Root Canal Treatment (RCT) is a procedure done when the tooth decay reaches the pulp (innermost layer of the tooth) causing inflammation.	<input type="checkbox"/>	<input type="checkbox"/>
Scaling	Rs.1500	Scaling is the cleaning of the teeth.	<input type="checkbox"/>	<input type="checkbox"/>
Tooth Extraction	Rs.15000	Tooth extraction is done when the teeth needs to be removed. The costs started from 15000 and can be raised according to the complexity.	<input type="checkbox"/>	<input type="checkbox"/>

At the bottom, there is a footer with copyright information: 'Copyright © 2022 Phulchowki Dental Clinic'.

Figure 512: Screenshot of the system - 28

All Users

Name	Profile Picture	Date of Birth	Email	Username	Phone Number	Address	Remove
Abhishek Rokaya		Nov. 13, 1990	abhishekrokaya12@gmail.com	itsabhishek	+977-9841234986	Saamakhusi, Kathmandu	
Amrit Nepali		April 9, 2022	amritnepali@gmail.com	itsamrit	9803289425	Chabahil, Kathmandu	
Lionel Messi		Oct. 13, 1990	leomessi@gmail.com	leomessi	+977-9841234986	Samakhusi, Kathmandu	

Showing 1 to 3 of 3 entries

Copyright © 2022 Phulchowki Dental Clinic

Figure 513: Screenshot of the system - 29

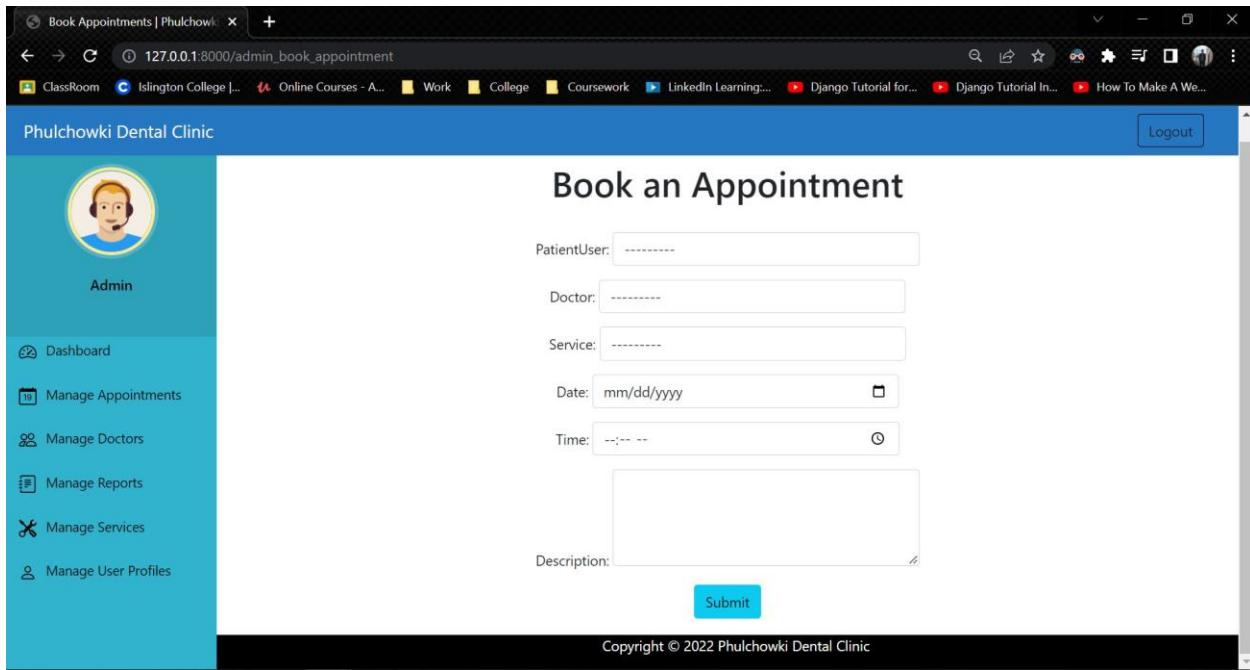


Figure 514: Screenshot of the system - 30

The screenshot shows a web application interface for managing doctors at Phulchowki Dental Clinic. On the left, a sidebar titled 'Admin' lists several menu items: Dashboard, Manage Appointments, Manage Doctors, Manage Reports, Manage Services, and Manage User Profiles. The main content area is titled 'Add Doctors' and contains the following form fields:

- Full name:
- Qualification:
- Date of birth:
- Email:
- Phone number:
- Address:
- Profile picture: No file chosen
- Experience:

A blue 'Submit' button is located below the experience field. At the bottom of the page, a copyright notice reads 'Copyright © 2022 Phulchowki Dental Clinic'.

Figure 515: Screenshot of the system - 31

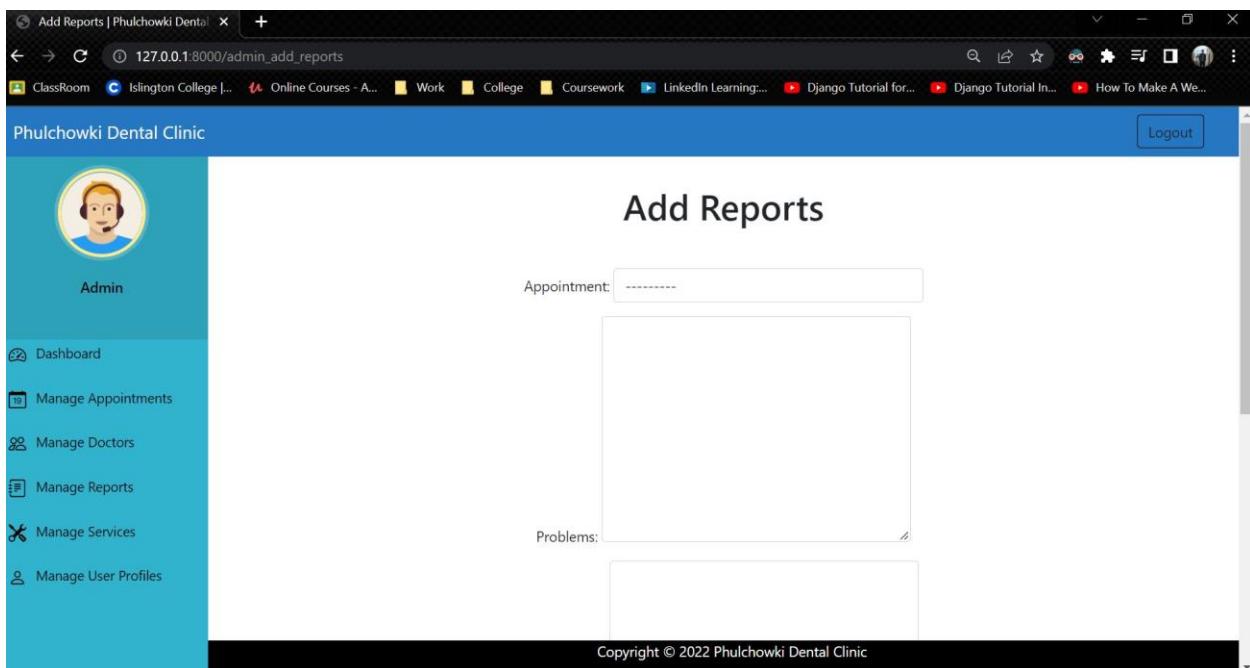


Figure 516: Screenshot of the system - 32

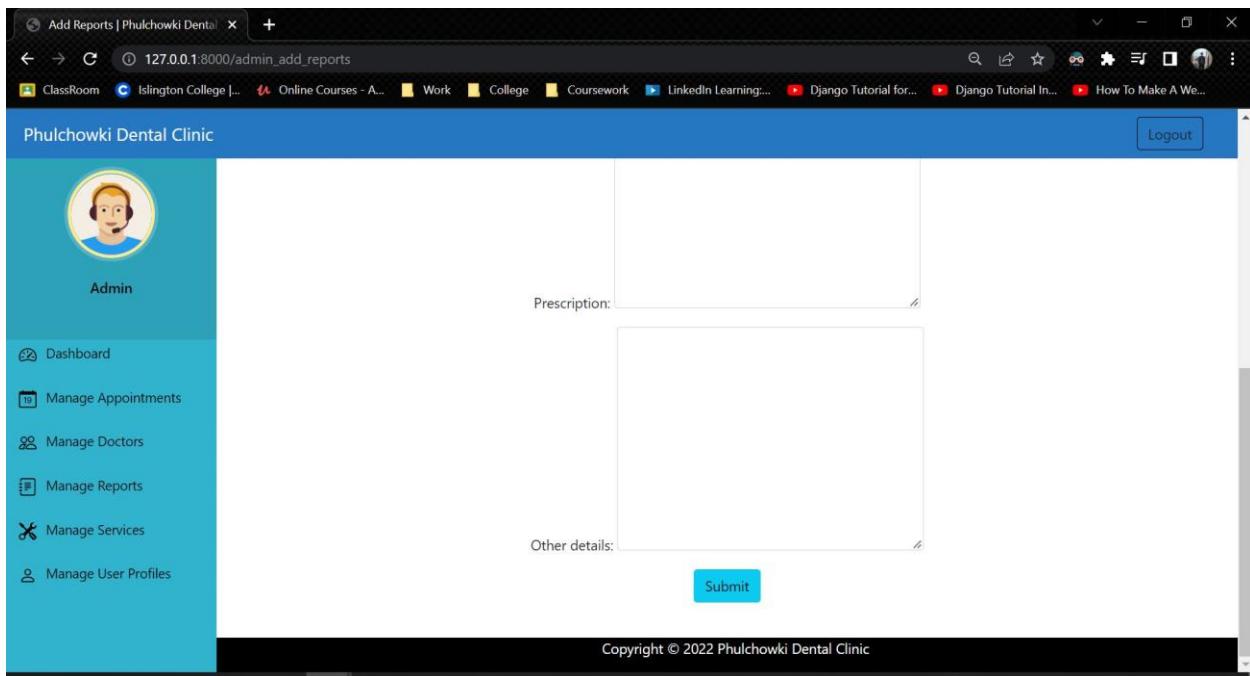


Figure 517: Screenshot of the system - 33

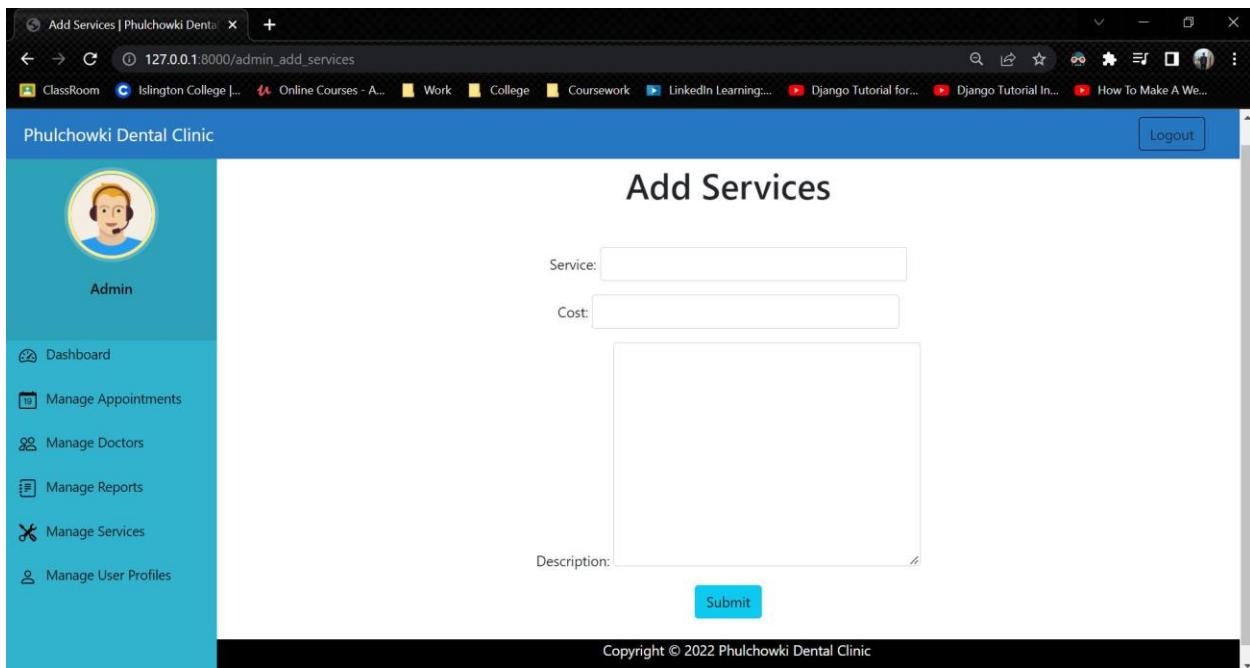


Figure 518: Screenshot of the system - 34

7.9 Appendix I: User Feedback

Link to the User Feedback Survey: <https://forms.gle/UkVzcEvtgrnssjJY6>

7.9.1 User Feedback Form

User Feedback Form for Dental Appointment System

Project Name: Dental Appointment System

Description of Project:
Dental Appointment System is an online appointment system that helps people to save their time by scheduling an appointment for their dental check-up using a web application. This project is all about reducing the problem that people face when they have to get a dental appointment.

np01cp4a190164@islingtoncollege.edu.np (not shared) [Switch accounts](#)

*Required

Figure 519: User Feedback Form - 1

User Feedback Form for Dental A

Your Name *

Your answer

Did you like the application? *

Yes
 No
 Maybe

Did you face any issues while registering? *

Yes
 No

Figure 520: User Feedback Form - 2

User Feedback Form for Dental A

Rate the UI of the application *

1 2 3 4 5

Rate the UX of the application *

1 2 3 4 5

Rate the booking feature *

1 2 3 4 5

Figure 521: User Feedback Form - 3

The screenshot shows a Google Forms interface titled "User Feedback Form for Dental A". It contains three rating scales:

- Rate the report feature ***: A horizontal scale from 1 to 5 with radio buttons. The second button (labeled 2) is selected.
- Rate the browsing doctor feature ***: A horizontal scale from 1 to 5 with radio buttons. The third button (labeled 3) is selected.
- Rate the search feature ***: A horizontal scale from 1 to 5 with radio buttons. The fourth button (labeled 4) is selected.

At the bottom right of the form area, there is a red circular edit icon.

Figure 522: User Feedback Form - 4

The screenshot shows a Google Forms interface titled "User Feedback Form for Dental A". It contains three sections:

- Rate the download report feature ***: A horizontal scale from 1 to 5 with radio buttons. The first button (labeled 1) is selected.
- Will you use this application? ***: Three radio button options:
 - Yes
 - No
 - Maybe
- Will you recommend this application to other people? ***: Three radio button options:
 - Yes
 - No
 - Maybe

At the bottom left is a red "Submit" button, and at the bottom right is a "Clear form" link. There is also a red circular edit icon at the bottom right of the form area.

Figure 523: User Feedback Form - 5

7.9.2 Sample of Filed User Feedback Forms

The screenshot shows a Google Forms survey titled "User Feedback Form for Dental A". The survey consists of several questions:

- Your Name ***: The response is "Abhishek Rokaya".
- Did you like the application? ***: The response is "Yes" (radio button selected).
- Did you face any issues while registering? ***: The response is "No" (radio button selected).
- Rate the UI of the application ***: There is a small icon of a pencil inside a circle.

Figure 524: Sample of Filled User Feedback Form – 1

The screenshot shows a Google Forms survey titled "User Feedback Form for Dental". It contains three rating questions with scales from 1 to 5. The first question is "Rate the UI of the application *". The second is "Rate the UX of the application *". The third is "Rate the booking feature *". In all three cases, the user has selected a rating of 4, which is highlighted with a red circle.

Rating	UI Rating	UX Rating	Booking Rating
1	Unselected	Unselected	Unselected
2	Unselected	Unselected	Unselected
3	Unselected	Unselected	Unselected
4	Selected	Selected	Selected
5	Unselected	Unselected	Unselected

Figure 525: Sample of Filled User Feedback Form - 2

The screenshot shows a Google Forms survey titled "User Feedback Form for Dental". It contains four rating questions with scales from 1 to 5. The first question is "Rate the report feature *". The second is "Rate the browsing doctor feature *". The third is "Rate the search feature *". The fourth question, "Rate the download report feature *", is partially visible at the bottom. In all three visible cases, the user has selected a rating of 5, which is highlighted with a red circle.

Rating	Report Rating	Browsing Doctor Rating	Search Rating
1	Unselected	Unselected	Unselected
2	Unselected	Unselected	Unselected
3	Unselected	Unselected	Unselected
4	Unselected	Unselected	Unselected
5	Selected	Selected	Selected

Figure 526: Sample of Filled User Feedback Form - 3

The screenshot shows a Google Forms survey titled "User Feedback Form for Dental". The first question asks to rate the download report feature, with a scale from 1 to 5. The response "5" is selected. The second and third questions ask if the user will use and recommend the application, both with "Yes" selected. At the bottom are "Submit" and "Clear form" buttons.

Rate the download report feature *

1 2 3 4 5

Will you use this application? *

Yes
 No
 Maybe

Will you recommend this application to other people? *

Yes
 No
 Maybe

Submit Clear form

Figure 527: Sample of Filled User Feedback Form - 4

7.9.3 User Feedback Form Result

Did you like the application?

56 responses

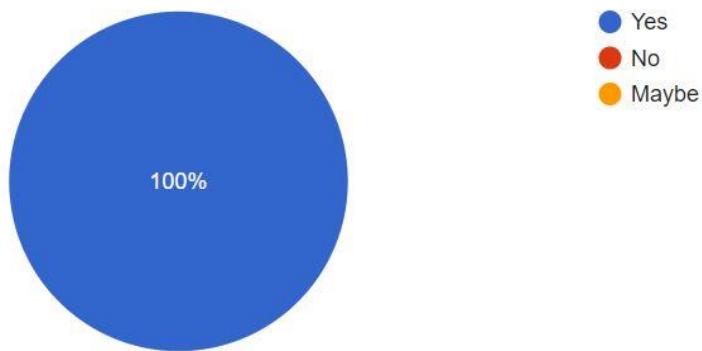


Figure 528: Result of User Feedback Form - 1

Did you face any issues while registering?

56 responses

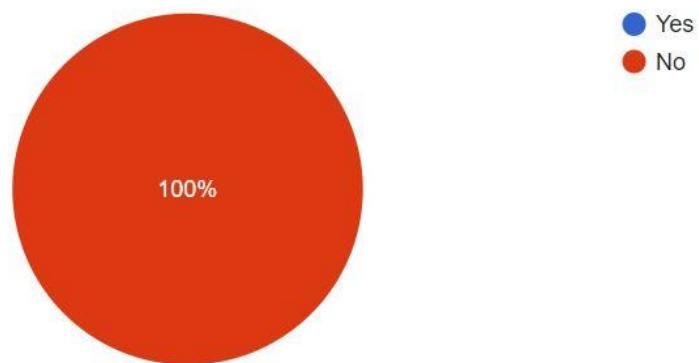


Figure 529: Result of User Feedback Form - 2

Rate the UI of the application

 Copy

56 responses

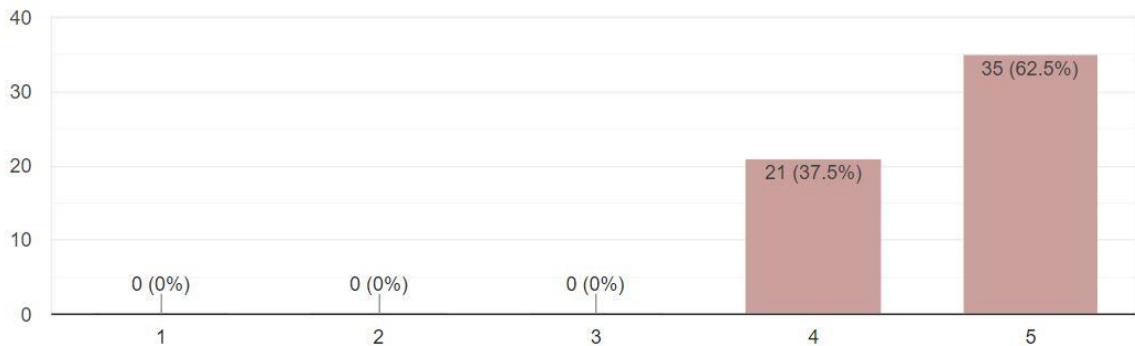


Figure 530: Result of User Feedback Form - 3

Rate the UX of the application

 Copy

56 responses

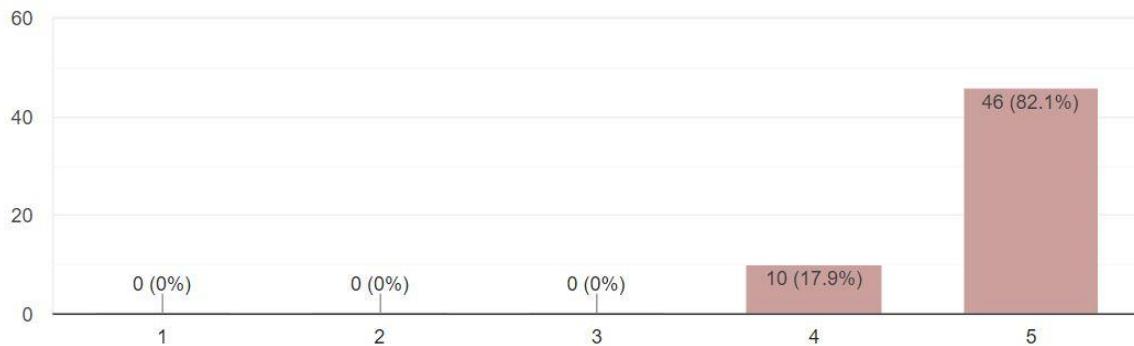


Figure 531: Result of User Feedback Form - 4

Rate the booking feature

56 responses

 Copy

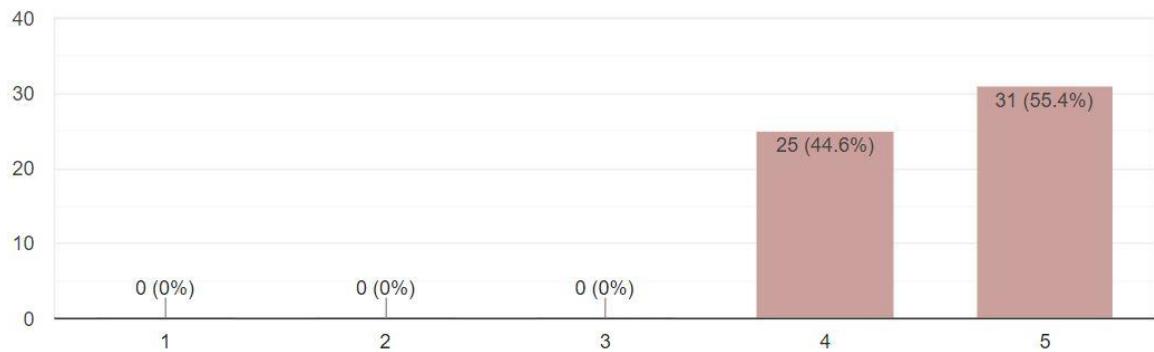


Figure 532: Result of User Feedback Form – 5

Rate the report feature

56 responses

 Copy

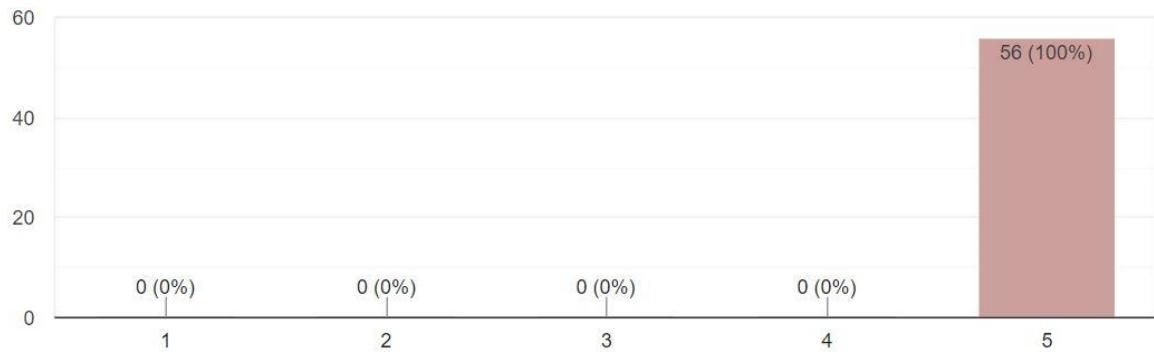


Figure 533: Result of User Feedback Form - 6

Rate the browsing doctor feature

 Copy

56 responses

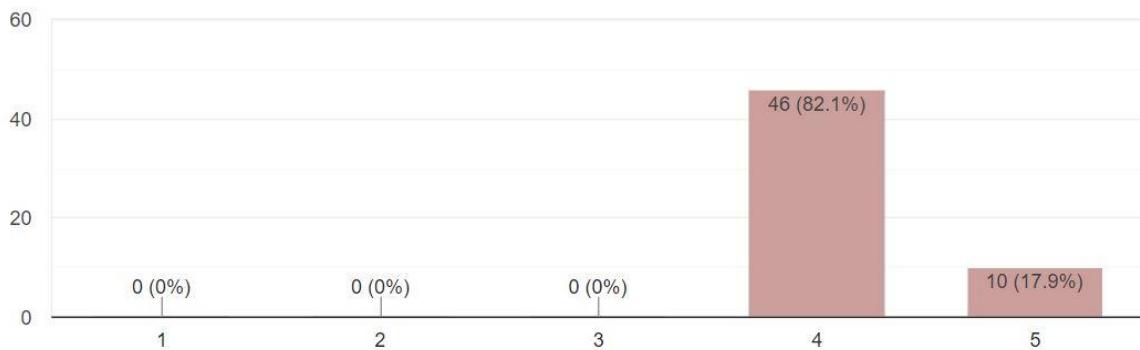


Figure 534: Result of User Feedback Form – 7

Rate the search feature

 Copy

56 responses

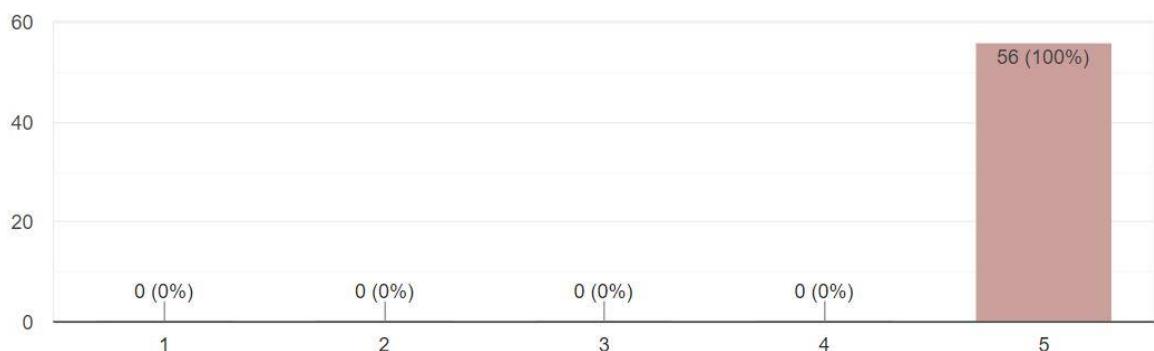


Figure 535: Result of User Feedback Form - 8

Rate the download report feature

 Copy

56 responses

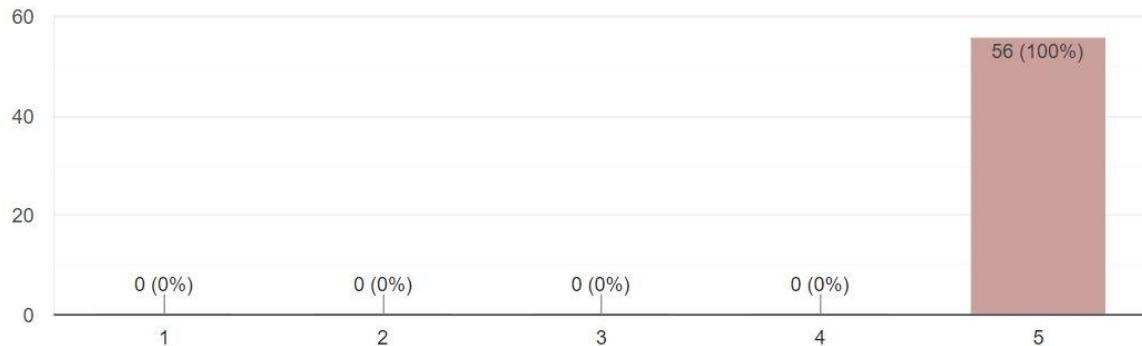


Figure 536: Result of User Feedback Form - 9

Will you use this application?

56 responses

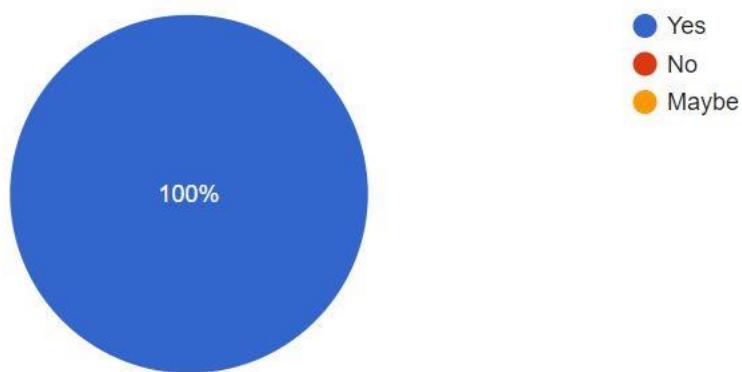


Figure 537: Result of User Feedback Form - 10

Will you recommend this application to other people?

56 responses

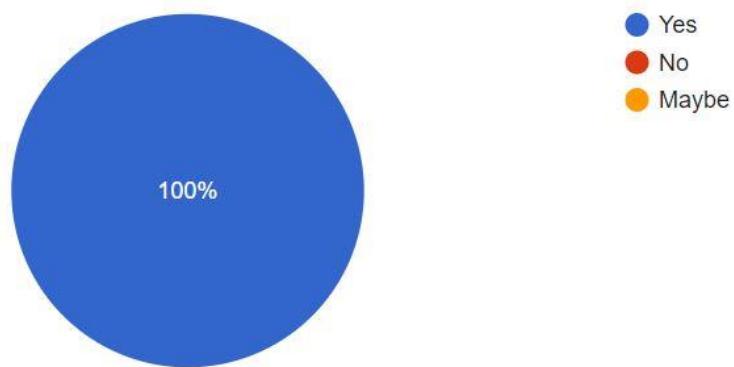
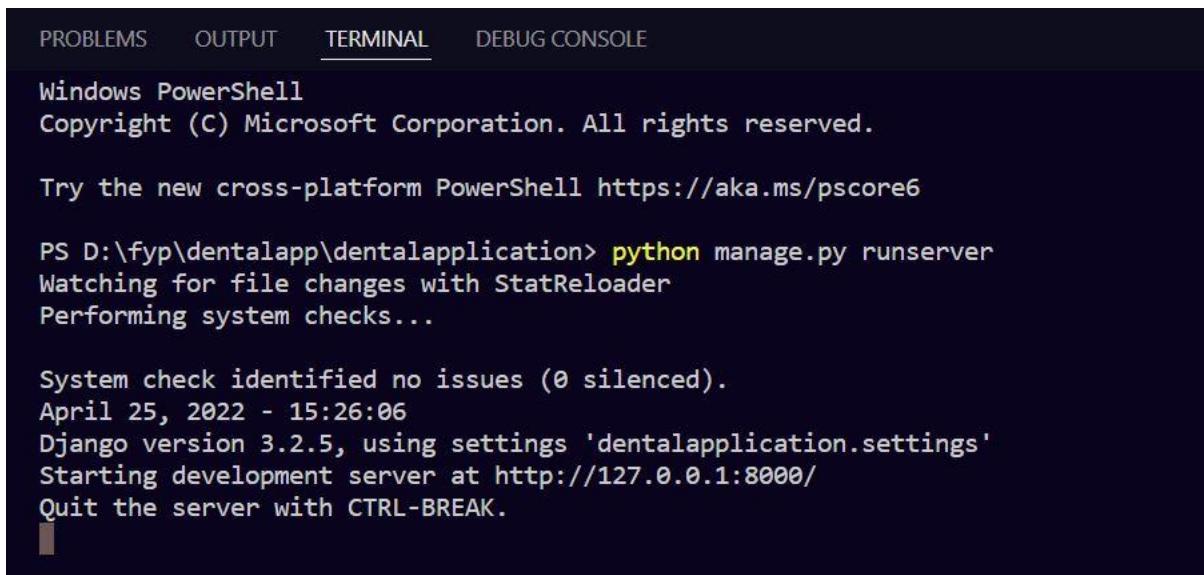


Figure 538: Result of User Feedback Form - 11

7.10 Appendix J - User Manual

To start the program:

In order to start the program. The command 'python manage.py runserver' is entered in the VS Code.



The screenshot shows the VS Code interface with the 'TERMINAL' tab selected. The terminal window displays the following output:

```
PROBLEMS    OUTPUT    TERMINAL    DEBUG CONSOLE

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\fyp\dentalapp\dentalapplication> python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
April 25, 2022 - 15:26:06
Django version 3.2.5, using settings 'dentalapplication.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

Figure 539: User Manual – 1

After doing this, open any web browser and enter the url '127.0.0.1:8000'. This will redirect to the home page.

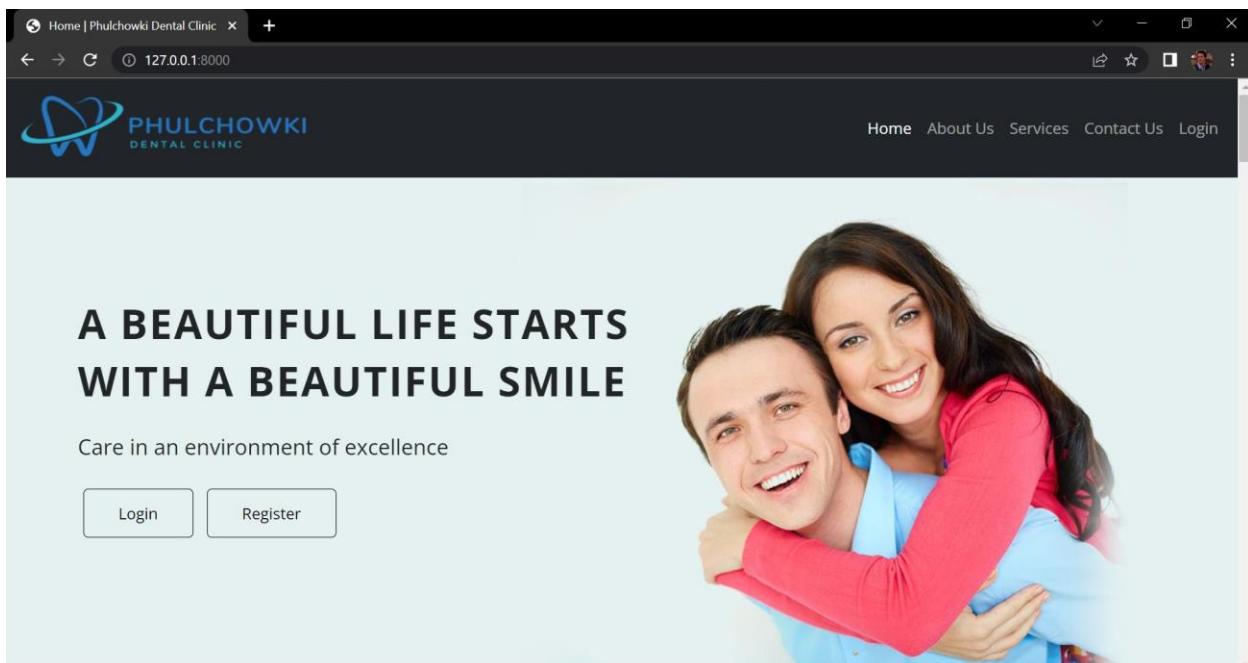


Figure 540: User Manual – 2

If you're not registered and have any queries. You can fill out the contact us form in the Contact Us section of the home page.

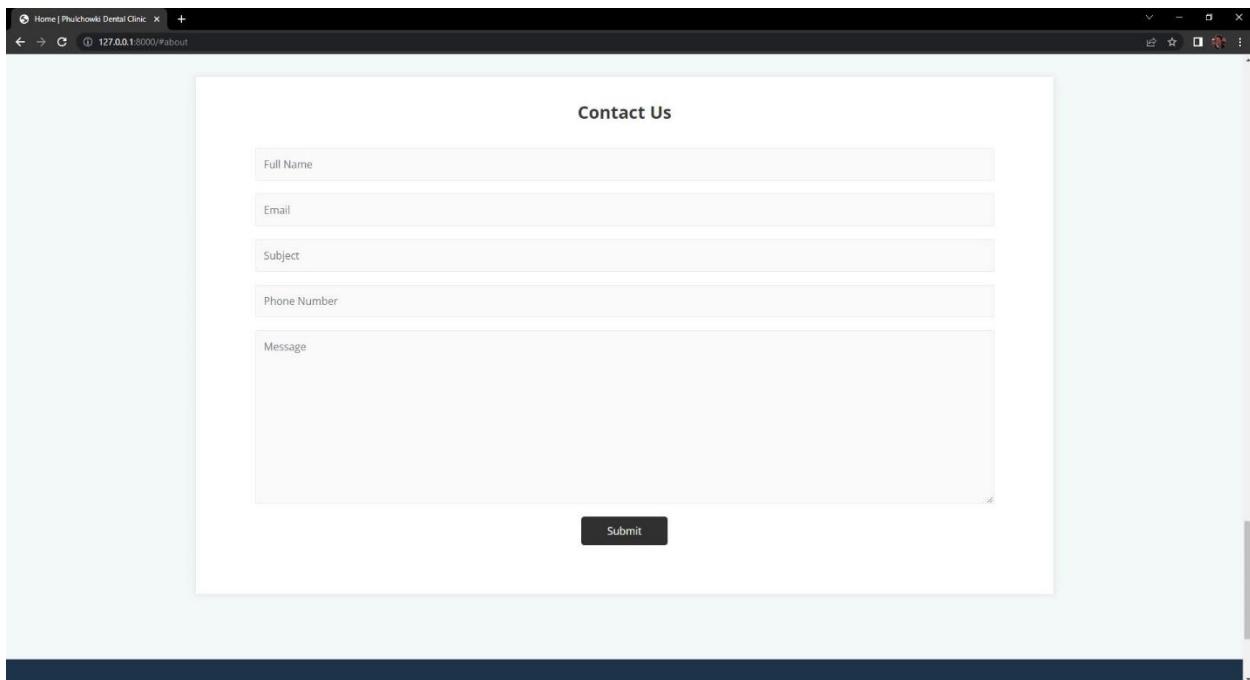


Figure 541: User Manual – 3

If you are not registered yet, click the 'Register' button and you will redirected to the Registration Page.

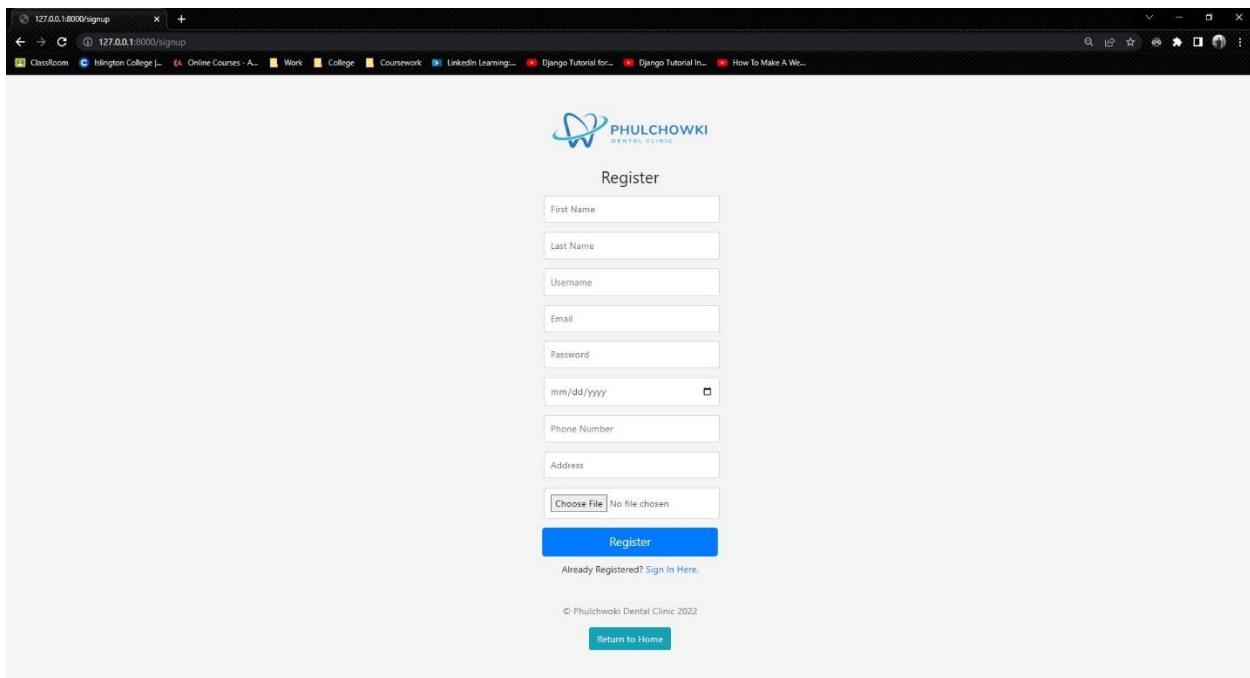
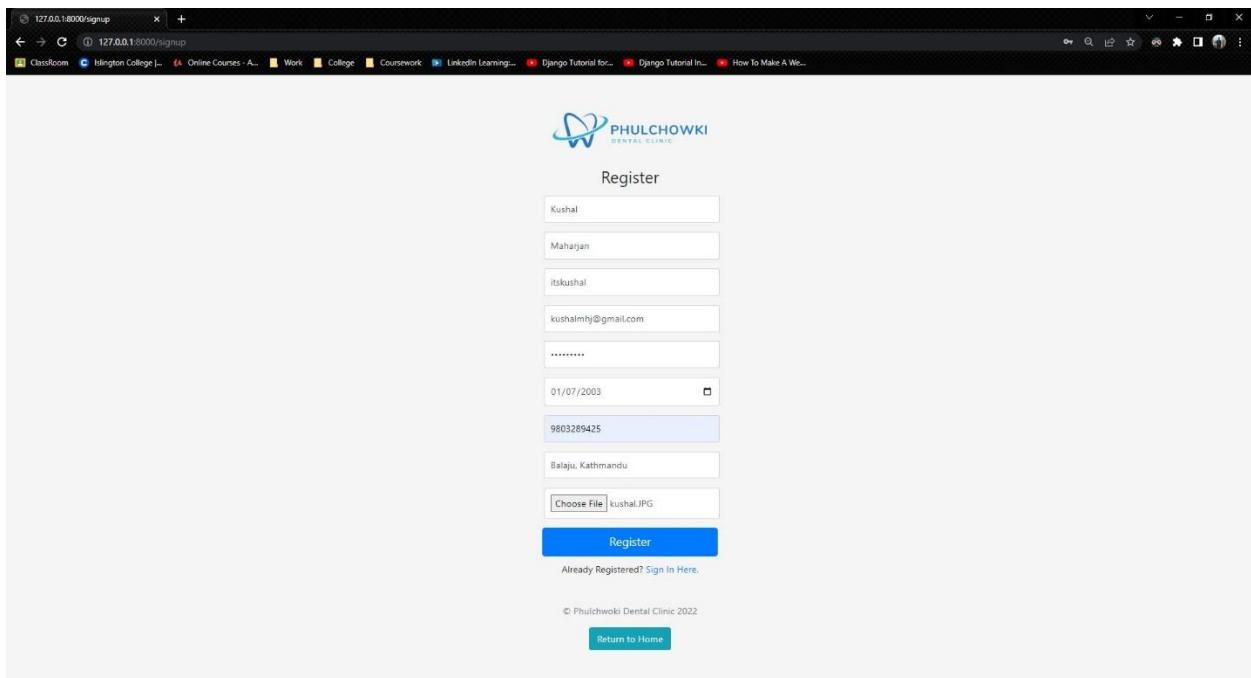
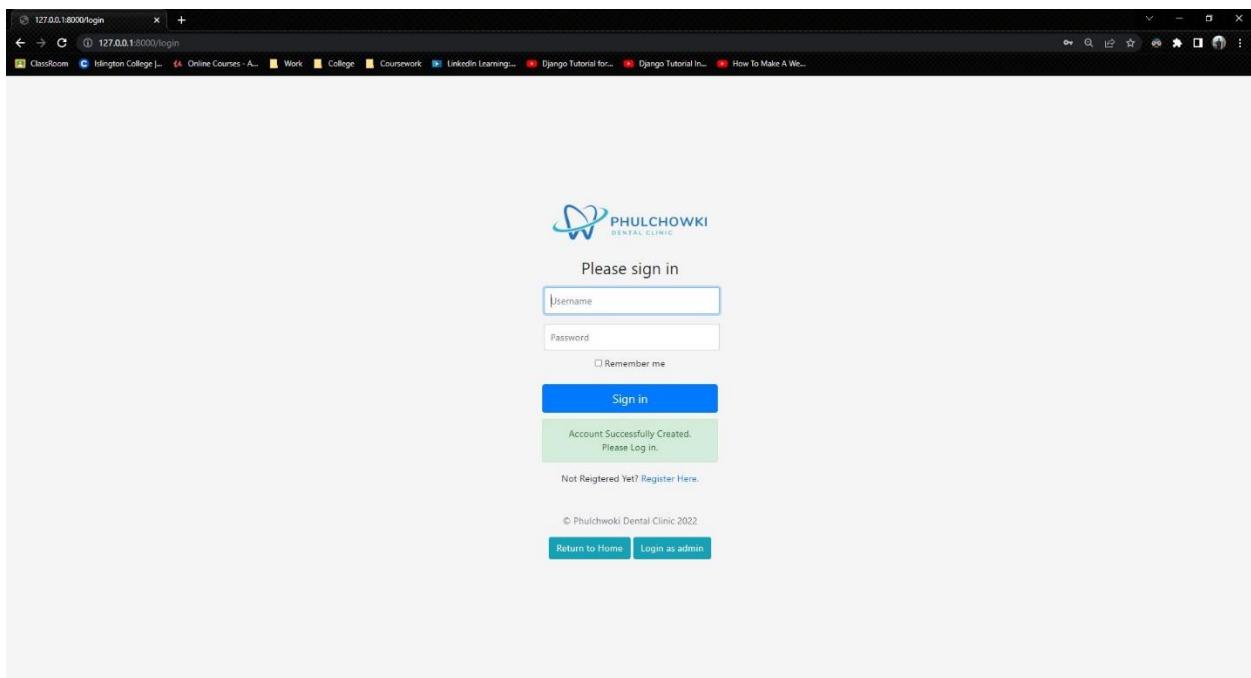


Figure 542: User Manual – 4

Fill out the form with appropriate details and you will be registered to the system.

*Figure 543: User Manual – 5**Figure 544: User Manual – 6*

After registering, you can then proceed to log in. Please enter valid credentials in order to get inside the system.

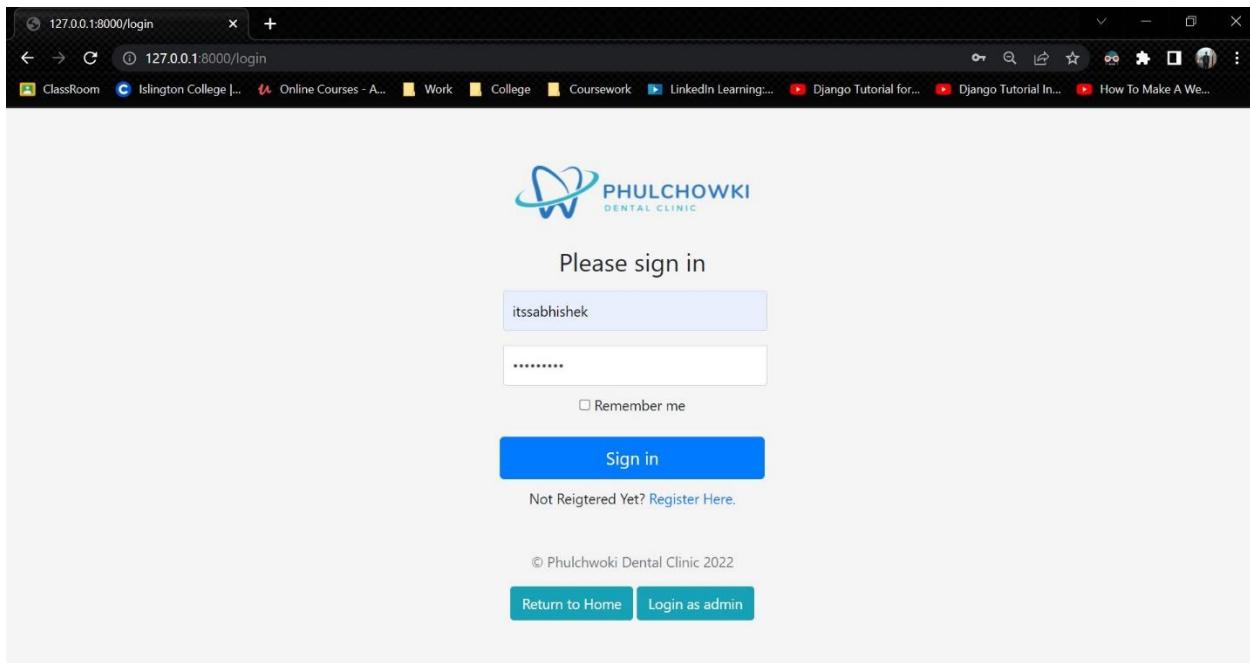


Figure 545: User Manual – 7

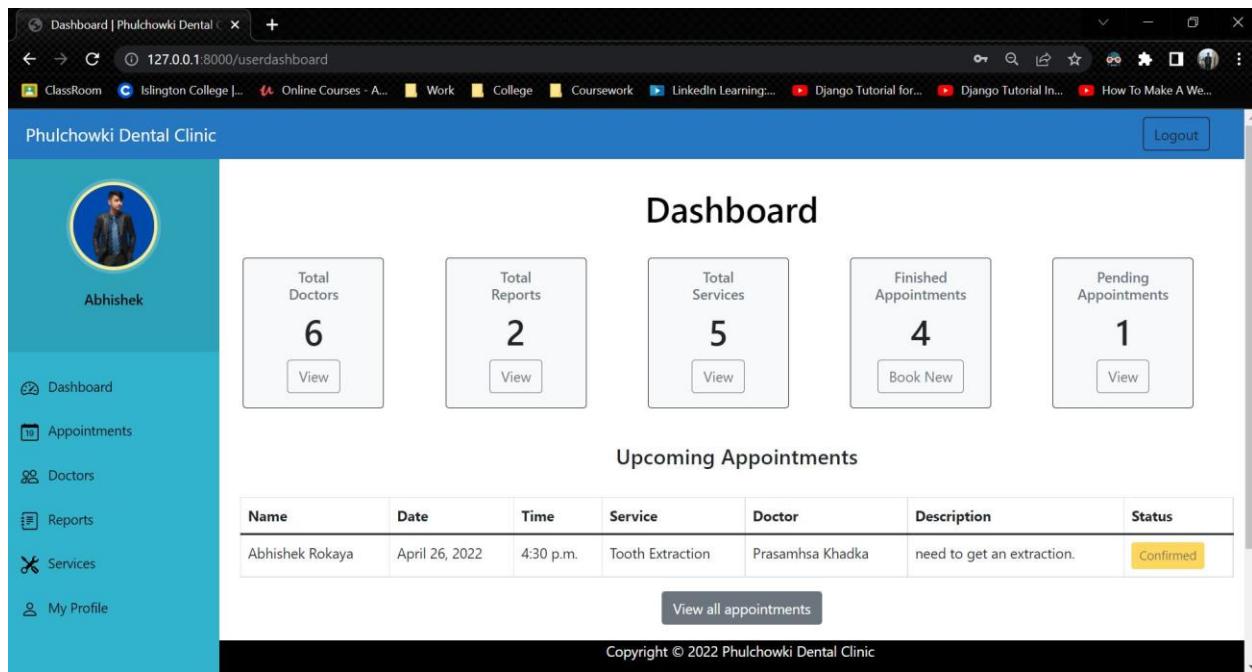


Figure 546: User Manual – 8

If you want to book an appointment, you can click the Appointment button in the side bar. Then you can click 'Book new appointment'.

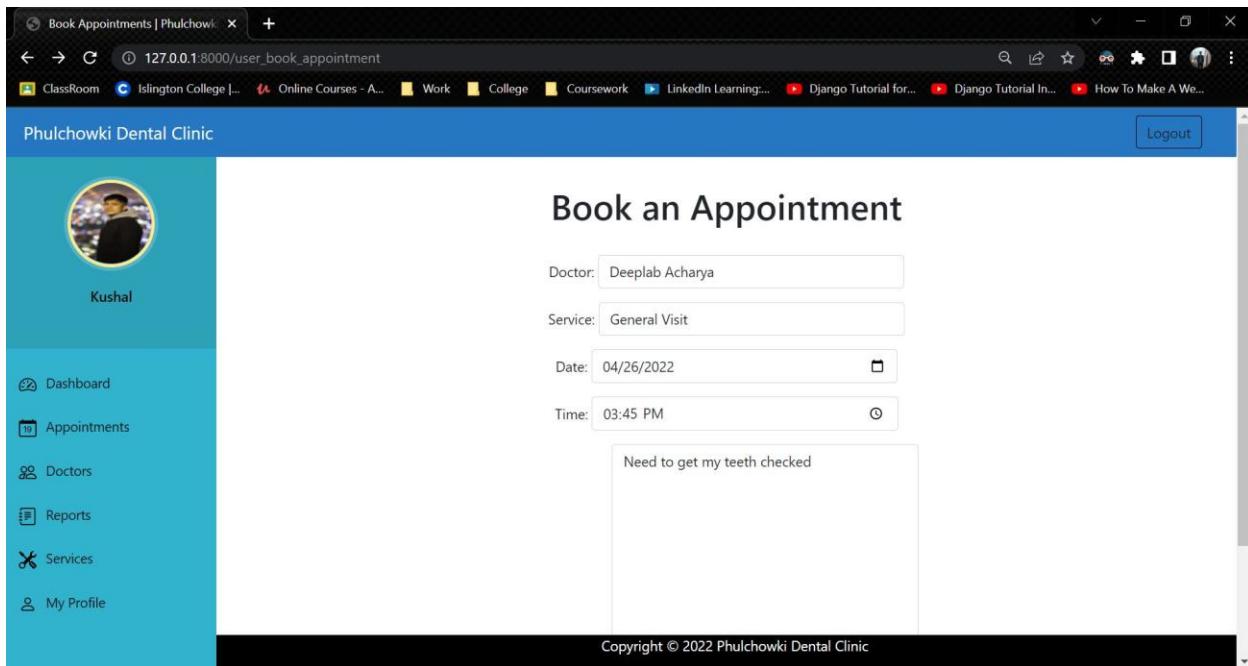


Figure 547: User Manual – 9

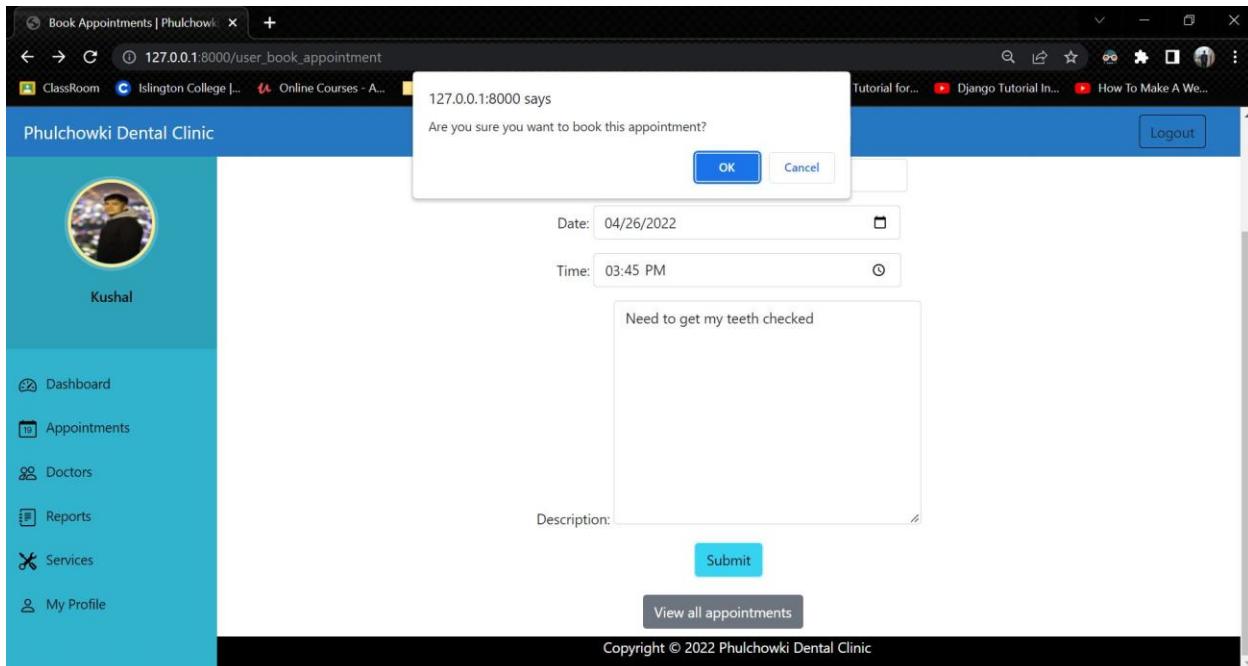


Figure 548: User Manual – 10

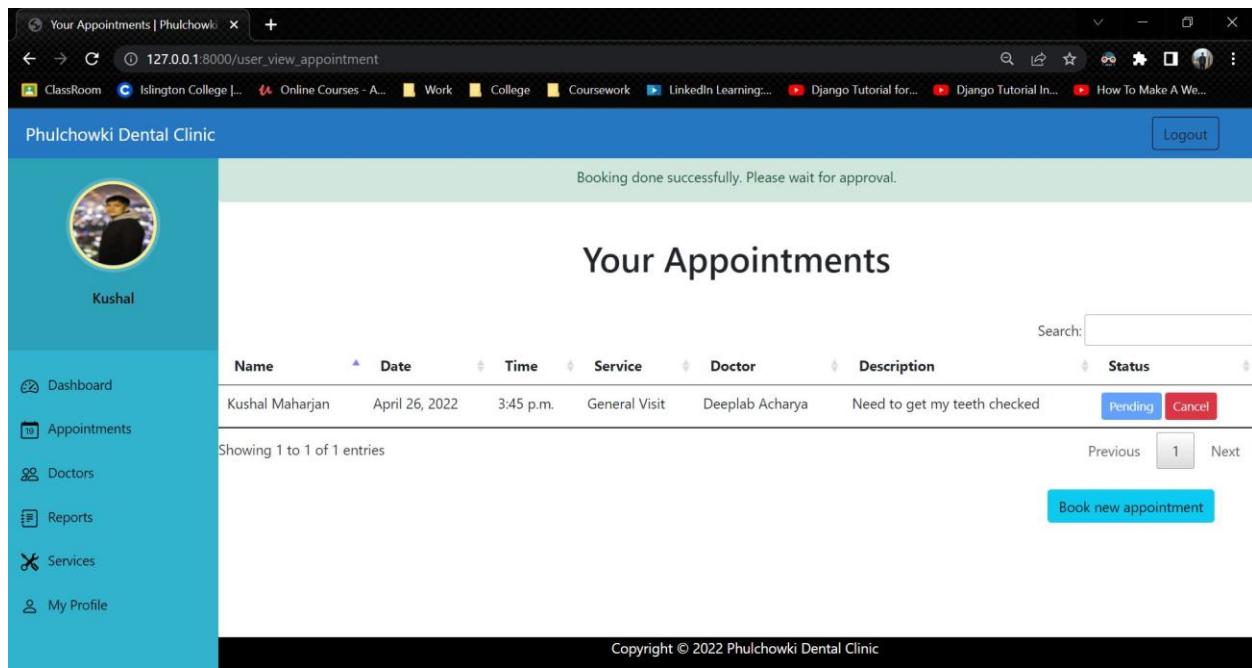


Figure 549: User Manual – 11

Please do mind that you can only book appointment from 9 am to 5pm any day of the week. After submitting the form, we can see that the appointment is pending. You can cancel the appointment as well. The admin reviews the appointment and approves the appointment.

Name	Date	Time	Service	Doctor	Description	Status
Abhishek Rokaya	April 28, 2022	2:45 p.m.	Tooth Extraction	Prasamhsa Khadka	I need to extract my wisdom tooth.	Finished
Abhishek Rokaya	April 29, 2022	12:50 p.m.	General Visit	Animesh Basnet	General checkup needed.	Finished
Abhishek Rokaya	May 2, 2022	2 p.m.	Scaling	Animesh Basnet	I need to get my teeth cleaned.	Finished
Abhishek Rokaya	May 6, 2022	4 p.m.	General Visit	Rabina Lama	Need a general visit.	Pending
Abhishek Rokaya	April 26, 2022	2:01 p.m.	General Visit	Prasamhsa Khadka	A general visit with the doctor	Finished
Abhishek Rokaya	April 26, 2022	4:30 p.m.	Tooth Extraction	Prasamhsa Khadka	need to get an extraction.	Confirmed

Figure 550: User Manual – 12

Here we can see the appointment is confirmed if the admin approves it. You can cancel the confirmed appointment as well. If the admin rejects the appointment, you will see the status as cancelled. If you want to cancel the appointment just press the cancel appointment and it will be marked as cancelled.

The screenshot shows a web browser window with the URL 127.0.0.1:8000/user_view_appointment. The page title is "Your Appointments | Phulchowki". The main content area displays a list of appointments under the heading "Your Appointments". A yellow banner at the top says "Booking cancelled.". The appointment list includes the following data:

Name	Date	Time	Service	Doctor	Description	Status
Abhishek Rokaya	May 6, 2022	4 p.m.	General Visit	Rabina Lama	Need a general visit.	Cancelled
Abhishek Rokaya	May 2, 2022	2 p.m.	Scaling	Animesh Basnet	I need to get my teeth cleaned.	Finished
Abhishek Rokaya	April 29, 2022	12:50 p.m.	General Visit	Animesh Basnet	General checkup needed.	Finished
Abhishek Rokaya	April 28, 2022	2:45 p.m.	Tooth Extraction	Prasamhsa Khadka	I need to extract my wisdom tooth.	Finished
Abhishek Rokaya	April 26, 2022	2:01 p.m.	General Visit	Prasamhsa Khadka	A general visit with the doctor	Finished
Abhishek Rokaya	April 26, 2022	4:30 p.m.	Tooth Extraction	Prasamhsa Khadka	need to get an extraction.	Confirmed

Copyright © 2022 Phulchowki Dental Clinic

Figure 551: User Manual – 13

You can sort the appointment list according to any of the headers. You can also search for it using date, time, service, doctor or status.

To view the doctor's details, you can click the Doctors button in the side bar.

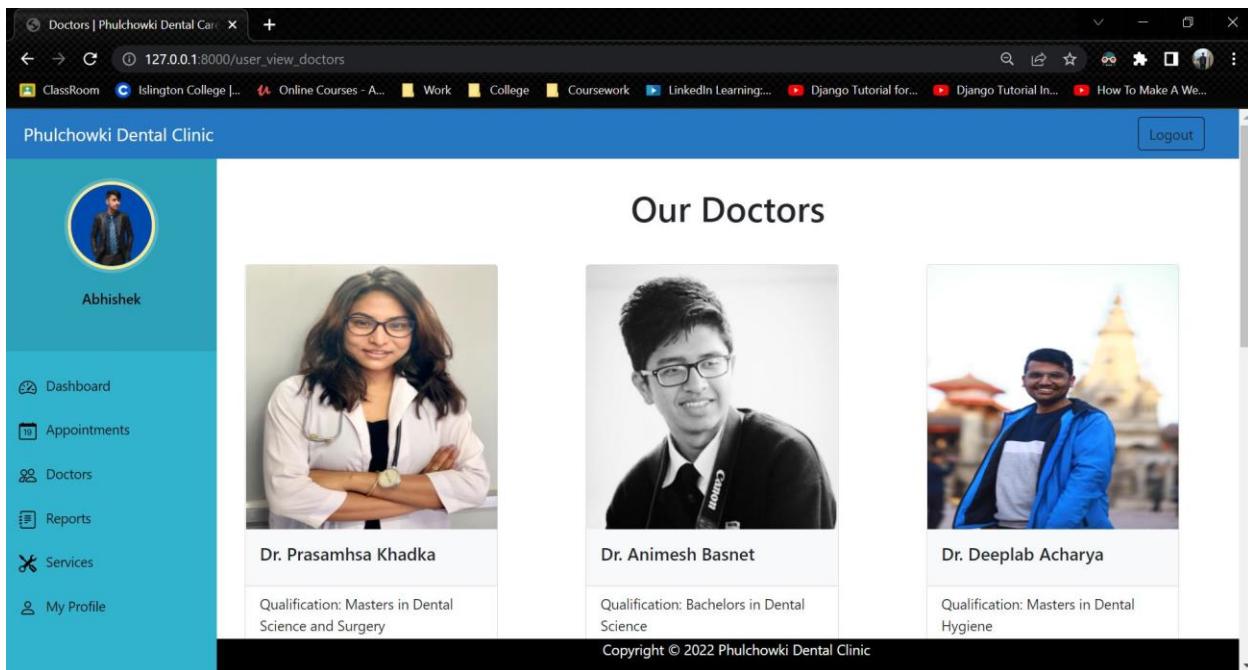


Figure 552: User Manual – 14

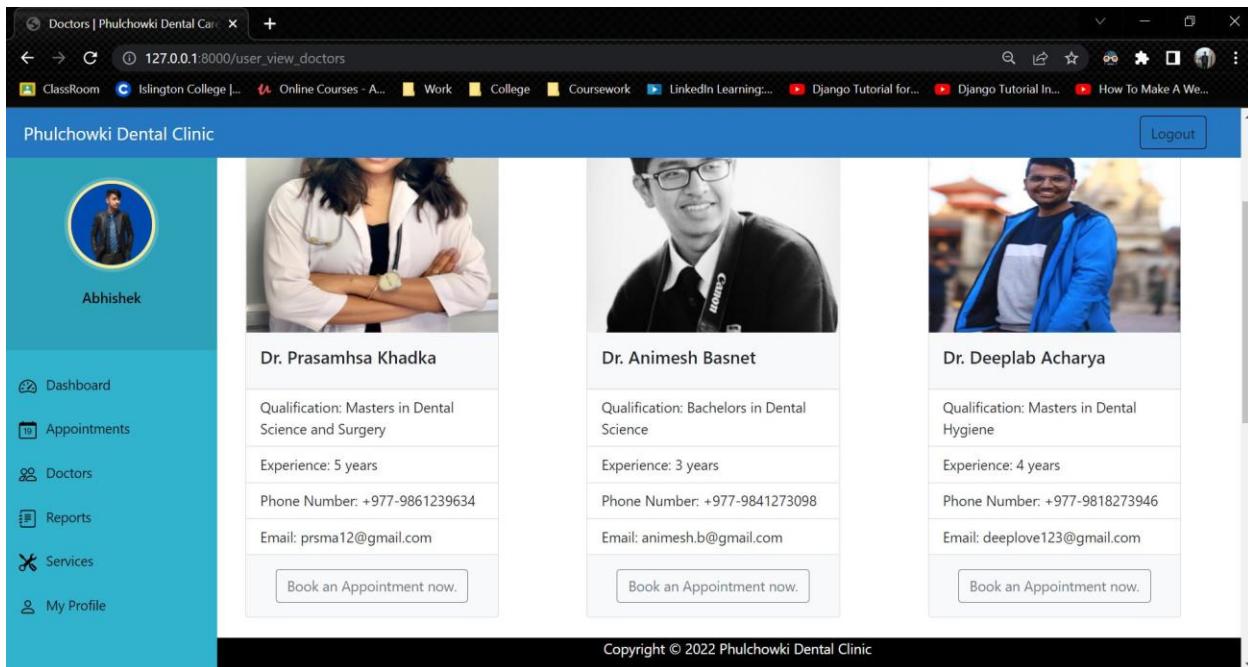


Figure 553: User Manual – 15

Here, you can see the list of doctors in a card form. You can book an appointment with any of the listed doctor.

In order to view your reports, you can click the Reports button in the side bar. The reports will only be available if you appointment has finished. The admin uploads you report after your appointment has finished.

Name	Appointment Date	Doctor	Service	Problems	Prescription	Other Details	Total Cost
Lionel Messi	April 27, 2022	Animesh Basnet	General Visit	There is a cavity in the upper right canal teeth.	The teeth needs to be filled. For now, please brush at least 2 times a day.	Avoid drinking cold drinks.	Rs.350

Showing 1 to 1 of 1 entries

Copyright © 2022 Phulchowki Dental Clinic

Figure 554: User Manual – 16

You can download it as pdf, excel and csv as well. You can sort it as well as search it using the search bar.

The screenshot shows a web-based application for managing dental reports. The main title is "Your Reports". On the left, there is a sidebar with a user profile icon and the name "Lionel". Below the profile are several menu items: Dashboard, Appointments, Doctors, Reports (which is currently selected), Services, and My Profile. At the bottom of the sidebar, there is a link to a PDF file named "Reports Phulchowki Dental Care.pdf". The main content area displays a table of reports. The table has columns for Name, Appointment Date, Doctor, Service, Problems, Prescription, and Other Details. One entry is shown: Lionel Messi, April 27, 2022, Animesh Basnet, General Visit, There is a cavity in the upper right canal teeth, The teeth needs to be filled. For now, please brush at least 2 times a day, Avoid drinking cold drinks, and Rs.350. Below the table, it says "Showing 1 to 1 of 1 entries". At the bottom right of the main content area, there are buttons for "Previous", "1", and "Next". The footer of the page includes the copyright notice "Copyright © 2022 Phulchowki Dental Clinic".

Name	Appointment Date	Doctor	Service	Problems	Prescription	Other Details	Total Cost
Lionel Messi	April 27, 2022	Animesh Basnet	General Visit	There is a cavity in the upper right canal teeth.	The teeth needs to be filled. For now, please brush at least 2 times a day.	Avoid drinking cold drinks.	Rs.350

Figure 555: User Manual – 17

In order to view the services available in the clinic. You can click on the Services button in the side bar.

The screenshot shows a web-based user interface for a dental clinic. On the left, a sidebar features a circular profile picture of a man, the name "Abhishek", and a list of navigation links: Dashboard, Appointments, Doctors, Reports, Services, and My Profile. The main content area is titled "Our Services" and displays a table of services with columns for Name, Cost, and Description. The services listed are Braces, General Visit, RCT, Scaling, and Tooth Extraction. Each service entry includes a brief description. A search bar is located at the top right of the main content area. At the bottom right, there are buttons for Previous, Next, and a page number (1). The footer contains the copyright notice "Copyright © 2022 Phulchowki Dental Clinic".

Name	Cost	Description
Braces	28000	We do all kinds of braces implantation to align your teeth. The base price starts from Rs.28000.
General Visit	350	We do a thorough check up on this and recommend necessary treatments.
RCT	4500	Root Canal Treatment (RCT) is a procedure done when the tooth decay reaches the pulp (innermost layer of the tooth) causing inflammation.
Scaling	1500	Scaling is the cleaning of the teeth.
Tooth Extraction	15000	Tooth extraction is done when the teeth needs to be removed. The costs started from 15000 and can be raised according to the complexity.

Figure 556: User Manual – 18

Here, you can view the list of services available in the clinic along with the cost and details. To view the details of your profile, click on the My Profile button in the side bar.

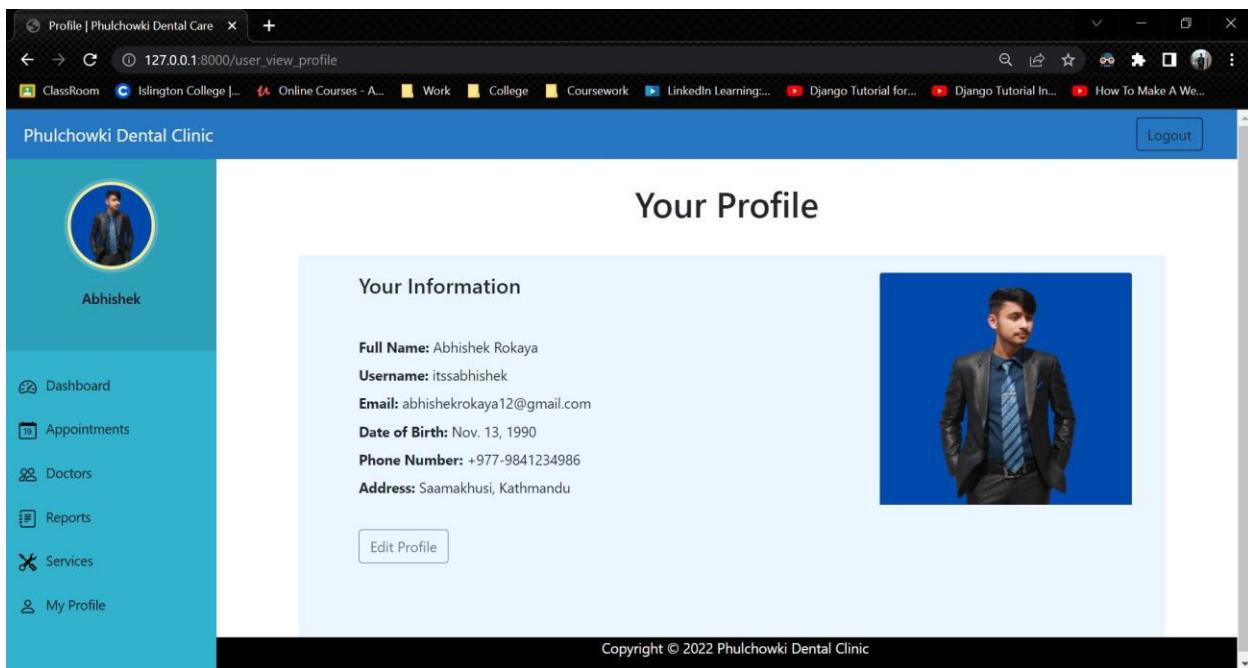


Figure 557: User Manual – 19

You can edit your details by clicking the Edit Profile button there.

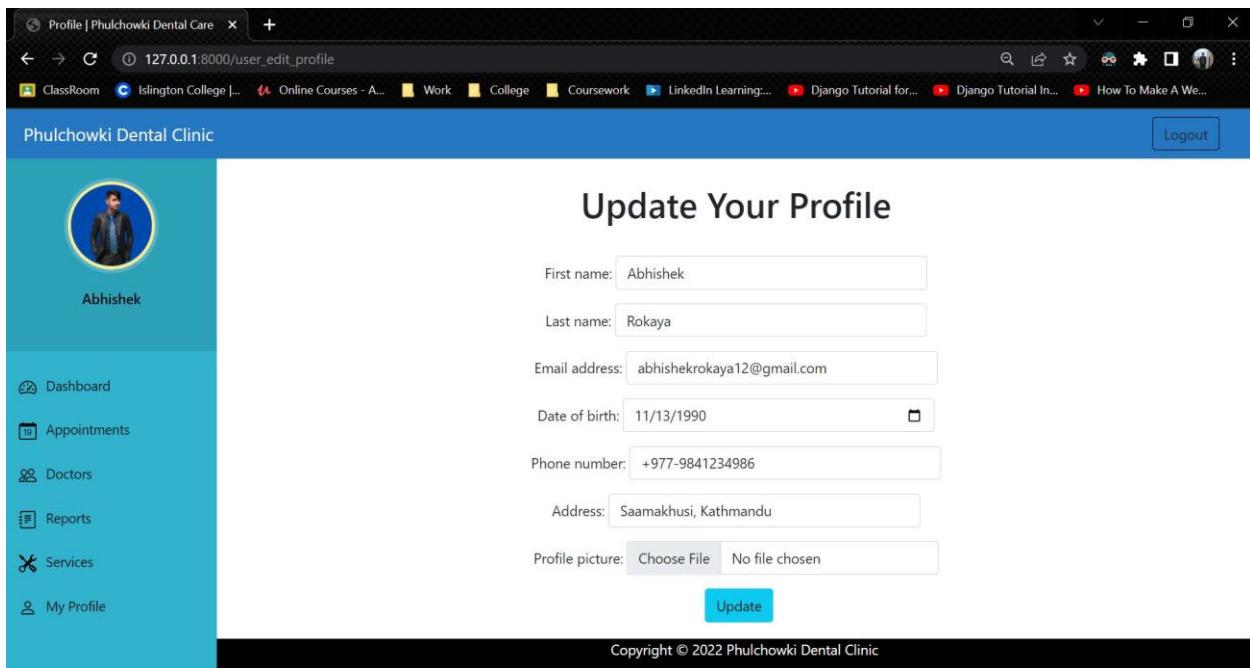


Figure 558: User Manual – 20

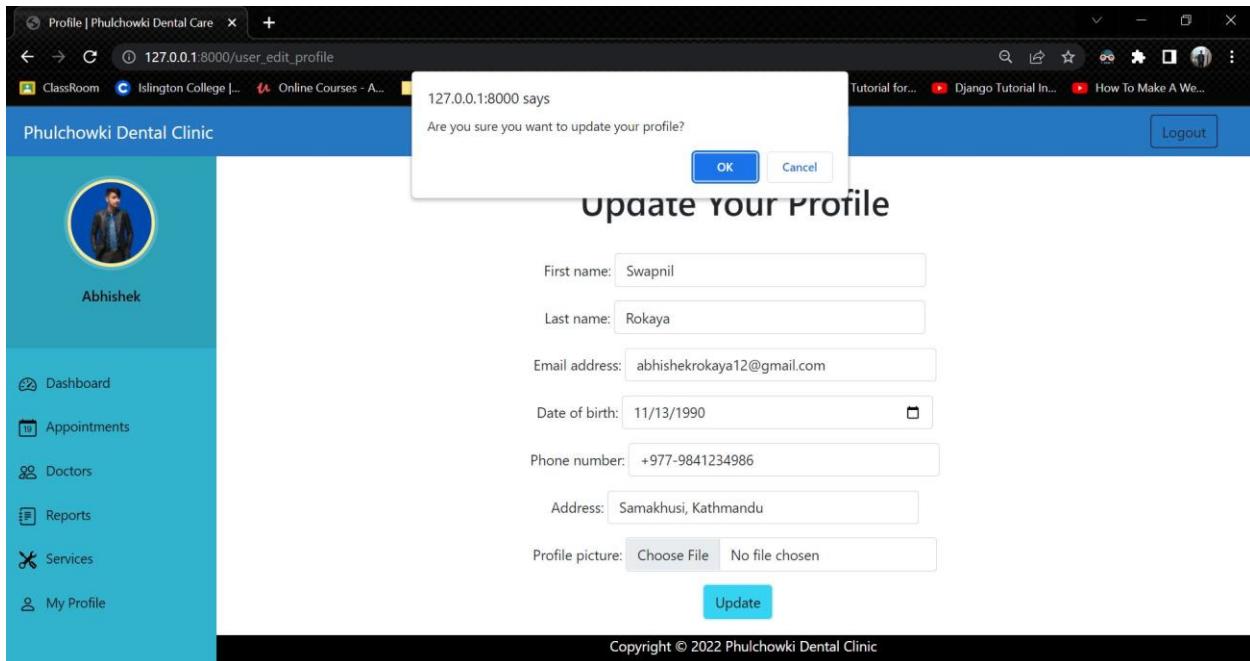


Figure 559: User Manual – 21

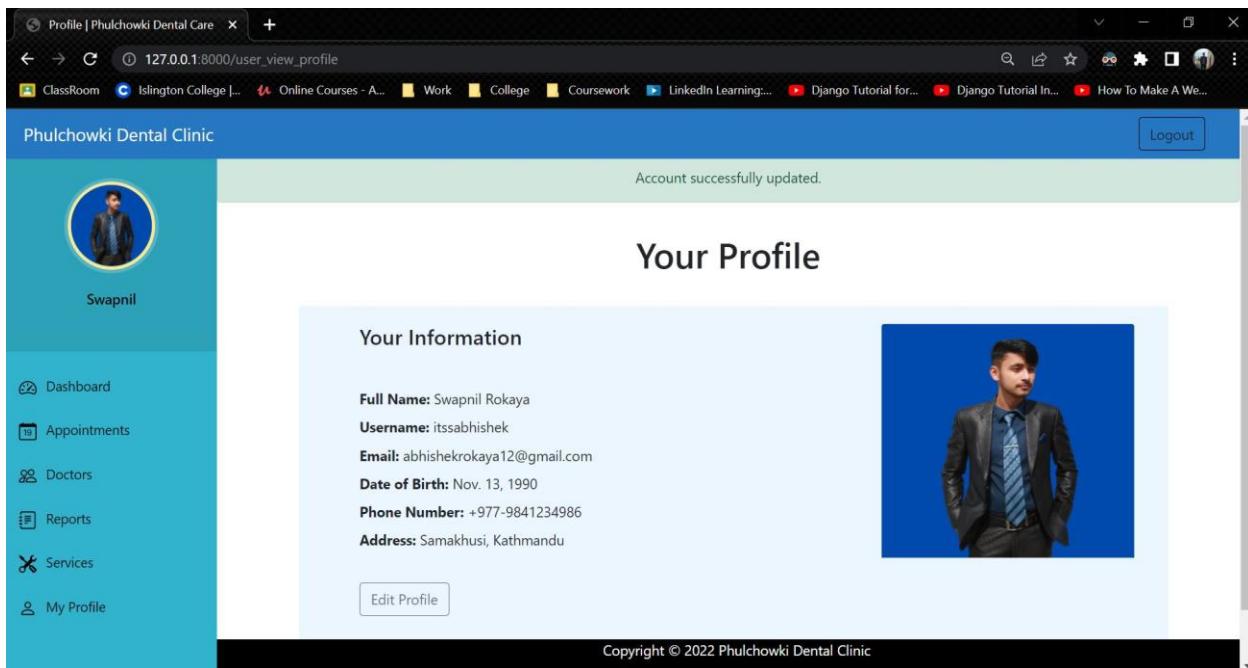


Figure 560: User Manual – 22

In order to log out, just click the Logout button in the top right corner.

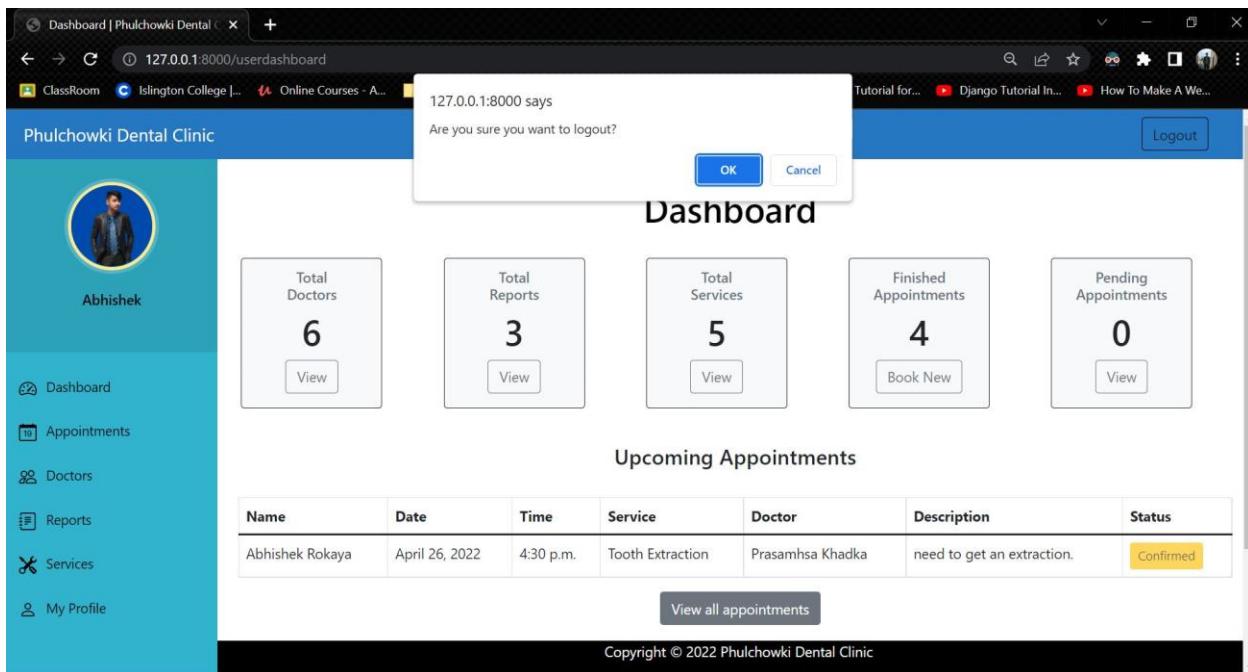


Figure 561: User Manual – 23

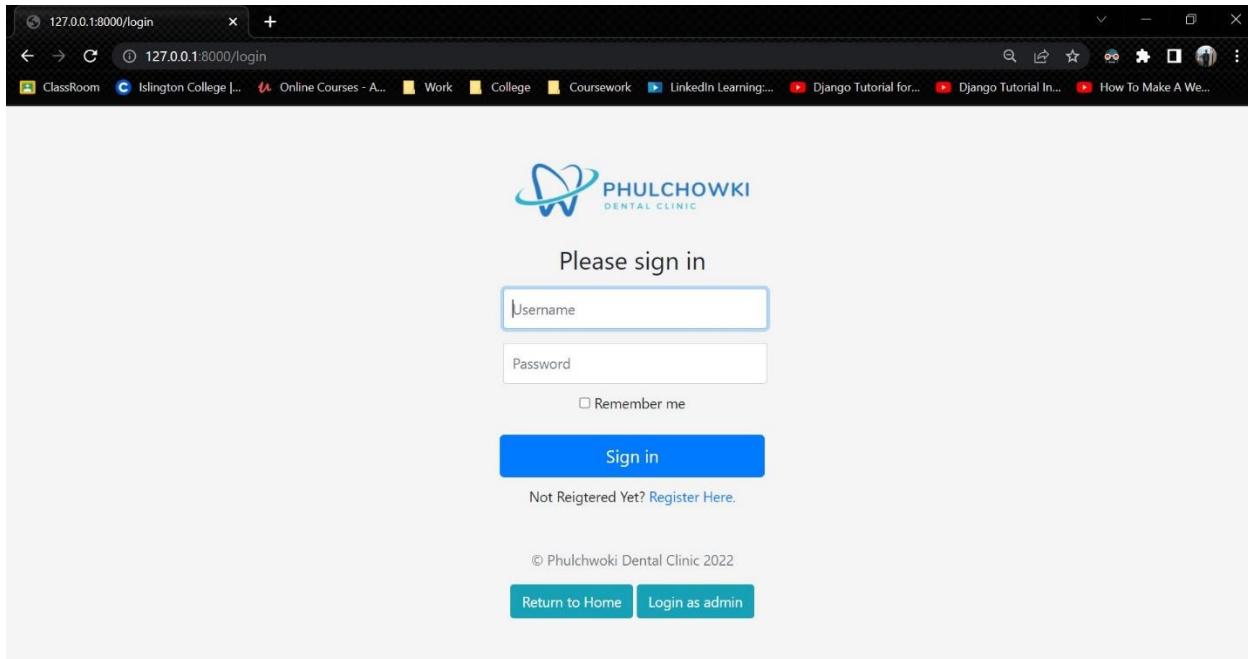


Figure 562: User Manual – 24

You will then be redirected to login page. In this way you can use the system with ease.

7.11 Appendix K: Future Work

7.11.1 Readings for future work

1. Payment with Khalti

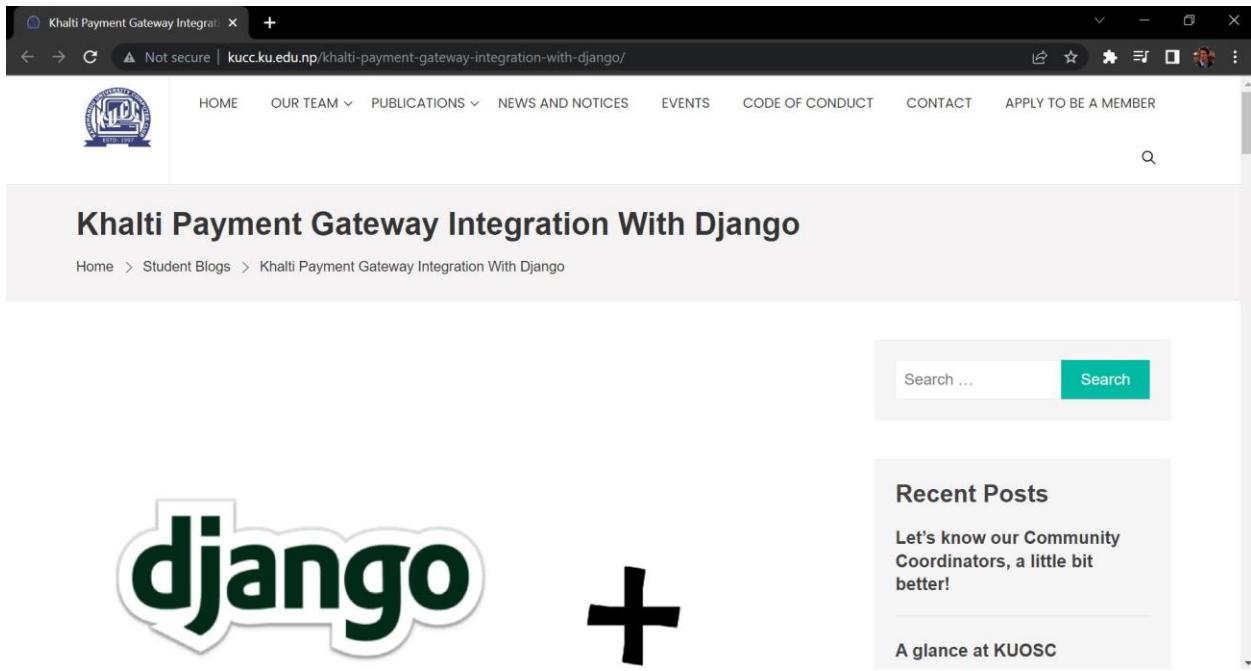


Figure 563: Readings for Future Work – 1

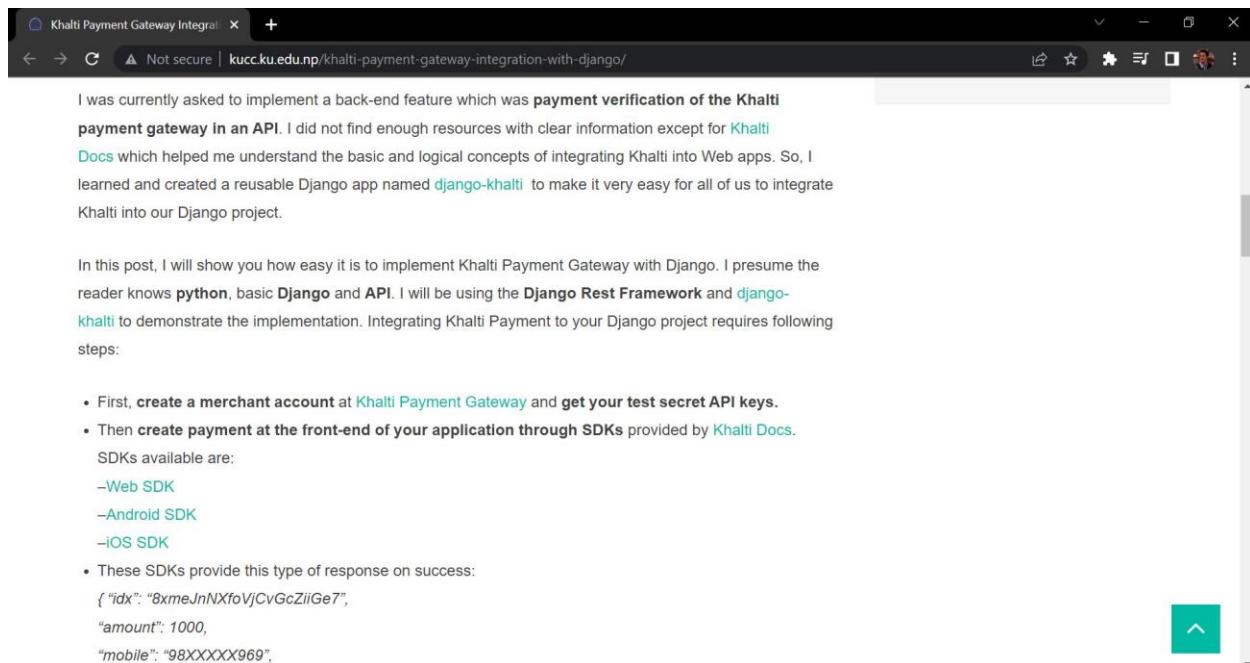


Figure 564: Readings for Future Work – 2

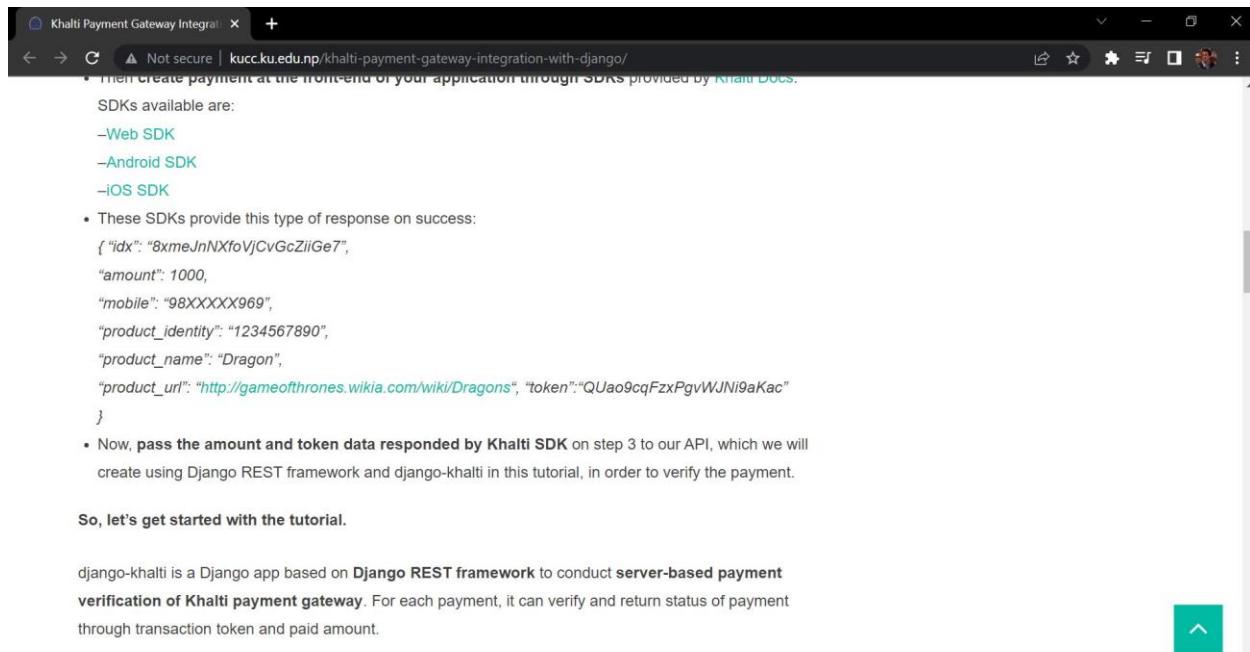


Figure 565: Readings for Future Work – 3

2. Payment with eSewa

The screenshot shows a web browser window titled "eSewa Document". The URL in the address bar is "developer.esewa.com.np/#/?id=introduction". The page content is the "Introduction" section of the eSewa Payment API documentation. On the left, there is a sidebar with a search bar and a list of links: "eSewa Payment API", "Introduction", "Epay", "Android", "iOS", "Flutter", "WooCommerce", and "Token Payment". The main content area starts with a heading "Introduction" and a welcome message: "Welcome to eSewa Payment API documentation." It explains that eSewa enables customers to make payments from associated merchant applications in a secure environment, with the payment amount deposited into their eSewa wallet or bank account. Below this is a section titled "Whom this is for" which states that business houses and organizations interested in adding online payment facilities to their systems and providing value added to customers. There is also a section titled "Intent of Document" which describes the document's purpose as being intended for partner merchants seeking to integrate and transact with eSewa, providing guidelines for integration, transaction flow, and validation processes.

Figure 566: Readings for Future Work – 3

The screenshot shows a web browser window for the eSewa Document at developer.esewa.com.np/#/id=introduction. The left sidebar lists various API endpoints: Introduction, Epay, Android, iOS, Flutter, WooCommerce, and Token Payment. The main content area starts with a search bar and an 'Intent of Document' section. It explains that the document is intended for partners merchant seeking to integrate and transact with the eSewa. It provides specific implementation details, general guidelines, transaction flow, and validation process. It also notes that the document does not define specific ways to implement the guidelines and procedures, mentioning vendor, platform, and architectural considerations. The focus is on establishing connectivity to eSewa and outlining the transaction process with or without verification. A link to 'Click here to become a merchant' is provided. Below it, there's a 'For any queries:' section with two bullet points: Email: merchant@esewa.com.np and Call At: 977-14414210.

Figure 567: Readings for Future Work – 4

The screenshot shows a web browser window for the ePay - eSewa Document at developer.esewa.com.np/#/epay?id=implementation. The left sidebar lists various API endpoints, including Implementation, which is highlighted. The main content area has a section titled 'Implementation'. It describes the eSewa ePay system integration process as the process of implementing eSewa as payment option, divided into two phases: Development & Staging Environment and Production Environment. The Development & Staging Environment phase involves testing process flow, transactions, and other integration results using a test eSewa ID with password. A note states that the test server may be taken off-line for maintenance or other technical reasons, during which users will receive "Service Temporarily Unavailable" message or any other message. In such case, please try again later or contact eSewa staff regarding issue. The primary objective of this stage are: Whether merchant requests are getting posted onto eSewa ePay system or not, Whether merchant is getting response of either a successful or failure transaction on their site, and Implementation and testing of verification process. The Production Environment section describes the production transaction carried out on URL: https://esewa.com.np. After successful testing phase, the partner merchant will be provided with merchant code for production environment credentials on request. Partner merchant's have to modify the request URL and point toward production server and use production environment merchant credentials. Then, partner merchant will be able to start accepting payments using eSewa ePay payment.

Figure 568: Readings for Future Work – 5

The screenshot shows a web browser window with the URL developer.esewa.com.np/#/epay?id=implementation. The page title is "Epay - eSewa Document". On the left, there is a sidebar with a navigation menu for the "eSewa Payment API" including sections like Introduction, Epay, Transaction Flow, System Interaction, Implementation, Integration, Payment, Payment Verification, Credentials & URLs, and FAQ. Below the sidebar are icons for Android, iOS, Flutter, WooCommerce, and Token Payment. The main content area has a heading "Integration" and a sub-section "Pre-requisite" which states: "It is required for merchant to provide eSewa with two email ids. Those emails will be used for creating merchant and end user in test environment." Another section "Payment" contains a note: "During this phase, merchant will use test user credentials to login in eSewa and process the transaction. Adequate balance will be updated to test user account." At the bottom of the main content, there is a link "Document Sample Endpoint".

Figure 569: Readings for Future Work – 6

This screenshot is similar to Figure 569 but includes code snippets in the main content area. In the "Payment" section, there is a "PYTHON" code block:

```
document.body.appendChild(form);
form.submit();
}


```

Below it, there is a "PHP" code block:

```
#!/usr/bin/python3
import requests as req

url = "https://uat.esewa.com.np/epay/main"
d = {'amt': 100,
     'pdc': 0,
     'pac': 0,
     'txAmnt': 0,
     'tAmt': 100,
     'pid': 'ee2c3ca1-696b-4cc5-a6be-2c40d929d453',
     'scd': 'EPAYTEST',
     'su': 'http://merchant.com.np/page/esewa_payment_success?q=su',
     'fu': 'http://merchant.com.np/page/esewa_payment_failed?q=fu'}
resp = req.post(url, d)
```

At the bottom, there is another "PHP" code block:

```
$url = "https://uat.esewa.com.np/epay/main";
```

Figure 570: Readings for Future Work – 7

3. Chat System

The screenshot shows a web browser window with the title "Realtime chat app using Django" and the URL "smazee.com/blog realtime-chat-app-using-python". The main content of the page is titled "Realtime chat app using Django Channels - Python". It describes Django as a high-level Python web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel. It's free and open source.

Django channel:

Channels is a project that takes Django and extends its abilities beyond HTTP - to handle WebSockets, chat protocols, IoT protocols, and more.

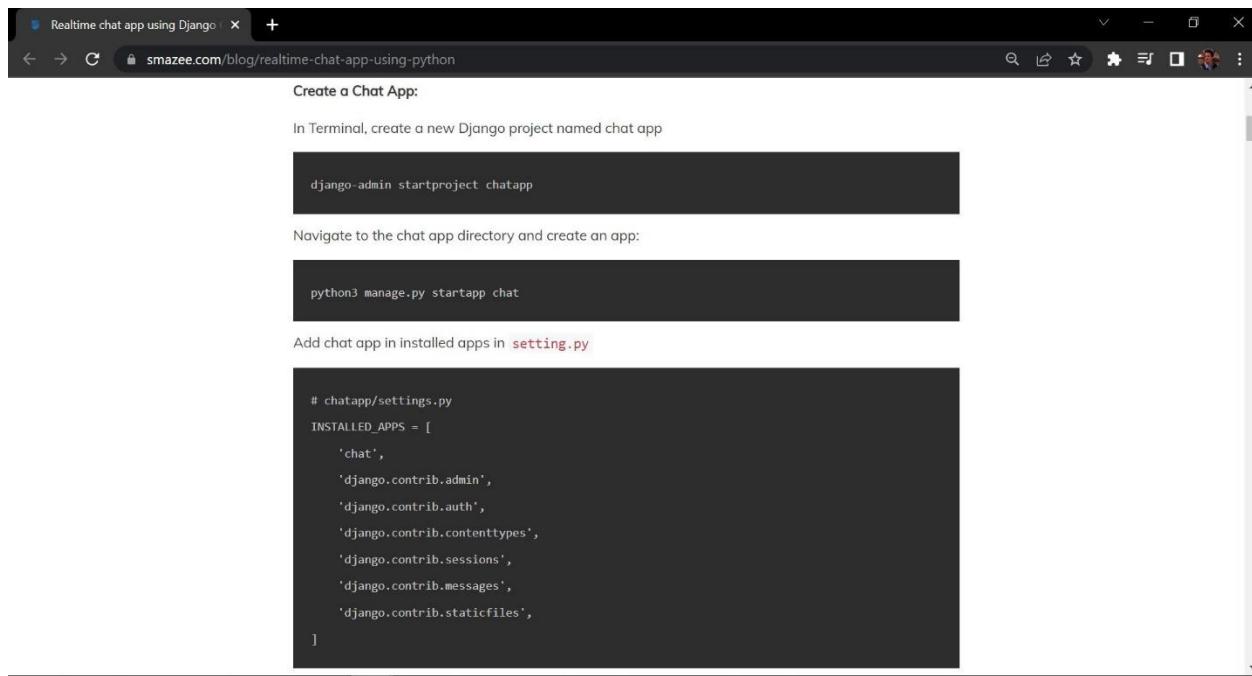
Pre-requesting :

1. Python and Pip
2. Django
3. Basic Knowledge of creating Django projects.

Create a Chat App:

In Terminal, create a new Django project named chat app

Figure 571: Readings for Future Work – 8



The screenshot shows a browser window with the title "Realtime chat app using Django". The URL is smazee.com/blog/realtime-chat-app-using-python. The content of the page is a step-by-step guide:

- Create a Chat App:**
- In Terminal, create a new Django project named chat app

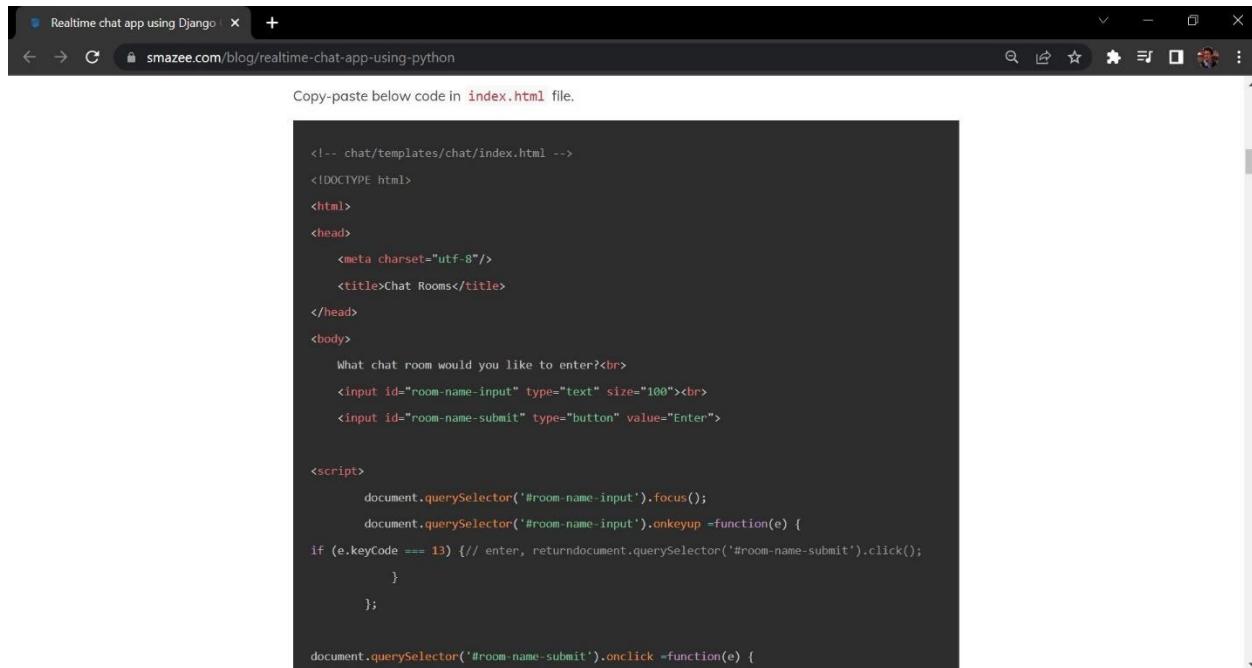
```
django-admin startproject chatapp
```

- Navigate to the chat app directory and create an app:

```
python3 manage.py startapp chat
```

- Add chat app in installed apps in `setting.py`

```
# chatapp/settings.py
INSTALLED_APPS = [
    'chat',
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
]
```

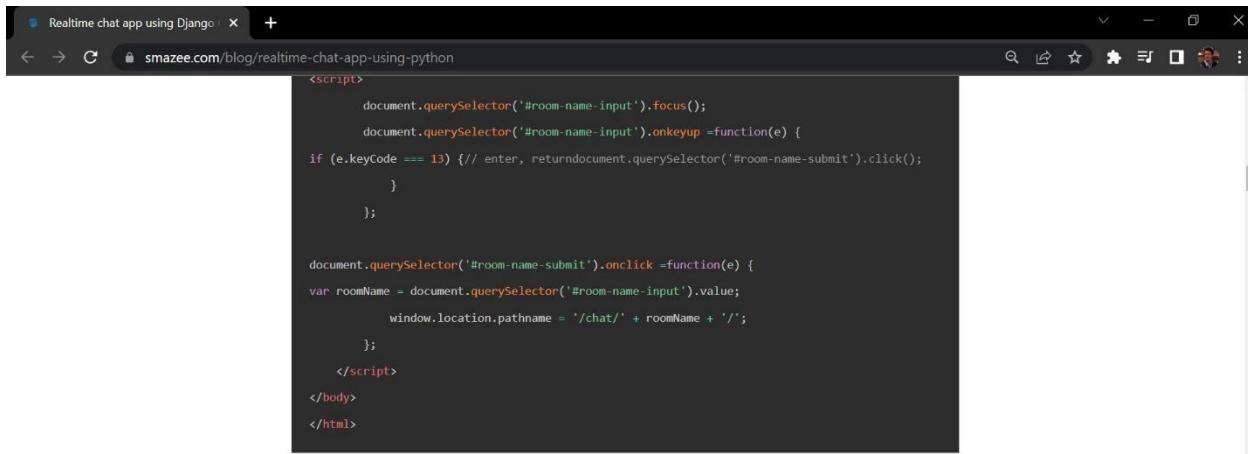

Figure 572: Readings for Future Work – 9


The screenshot shows a browser window with the title "Realtime chat app using Django". The URL is smazee.com/blog/realtime-chat-app-using-python. The content of the page is a snippet of HTML code:

```
<!-- chat/templates/chat/index.html -->
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8"/>
    <title>Chat Rooms</title>
</head>
<body>
    What chat room would you like to enter?<br>
    <input id="room-name-input" type="text" size="100"><br>
    <input id="room-name-submit" type="button" value="Enter">

    <script>
        document.querySelector('#room-name-input').focus();
        document.querySelector('#room-name-input').onkeyup =function(e) {
            if (e.keyCode === 13) { // enter, return
                document.querySelector('#room-name-submit').click();
            }
        };
        document.querySelector('#room-name-submit').onclick =function(e) {
```

Figure 573: Readings for Future Work – 10



The screenshot shows a browser window with the title "Realtime chat app using Django" and the URL "smazee.com/blog/realtime-chat-app-using-python". The page displays the HTML and JavaScript code for a room creation form. The code includes a script block that handles keydown events for the room name input field, focusing it and triggering a click on the submit button if the enter key is pressed. It also includes logic to capture the room name from the input field and update the window location's pathname to include the room name.

```
<script>
    document.querySelector('#room-name-input').focus();
    document.querySelector('#room-name-input').onkeyup =function(e) {
        if (e.keyCode === 13) { // enter, return
            document.querySelector('#room-name-submit').click();
        }
    };

    document.querySelector('#room-name-submit').onclick =function(e) {
        var roomName = document.querySelector('#room-name-input').value;
        window.location.pathname = '/chat/' + roomName + '/';
    };
</script>
</body>
</html>
```

Explanation:

This HTML code is to create a form with one input field to name the room and submit button. Once the submit button is pressed, the new room is created and redirected to that page.

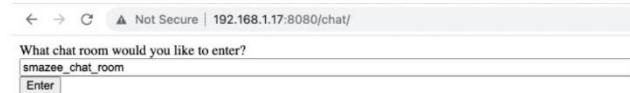


Figure 574: Readings for Future Work – 11