



islington college
(इस्लिंगटन कॉलेज)

Module Code & Module Title

CS6P05NI Final Year Project

Assessment Weightage & Type

40% FYP Final Report

Semester

2023-24 Autumn

PROJECT TITLE: Scrapequest

Student Name: Sharams Kunwar

London Met ID: 21049701

College ID: np01nt4a210112

Internal Supervisor: Raman Pradhananga

External Supervisor: Dikshant Koirala

Assignment Due Date: 24th April 2024

Assignment Submission Date: 24th April 2024

Word Count (Where Required): 8768

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

Abstract

This report discusses the development and implementation of "Scrapequest," a web-based scraping tool designed to streamline project discovery processes within IT organizations. Addressing the common inefficiencies associated with manual project searches, Scrapequest leverages automated data collection, cleaning, and processing to enhance efficiency and accuracy. The methodology employed is the Rational Unified Process (RUP), chosen for its iterative approach and emphasis on user requirements, ensuring a structured development and implementation phase. The report details the progression of the project, comparing Scrapequest's features and benefits with similar systems, and underscores its potential to significantly improve project discovery in the IT sector. Future work aims to expand its capabilities and integrate further enhancements to maintain its competitive advantage and adaptability in a dynamic industry landscape.

Acknowledgement

I would like to express my deepest gratitude by stating that I have managed to complete my Final Year Project Final Report within the given time frame. This report would not have been completed without the effort and co-operation of my supervisors Mr. Dikshant Koirala and Mr. Raman Pradhananga. Furthermore, I would like to sincerely thank all the helping hands that worked together directly or indirectly for the completion of this report.

The guidance from my supervisors helped me to work me out on my flaws and guided me for making my report more attractive and logical. Moreover, I express my sincerest gratitude to Islington College for providing a platform for better learning and developing creative skills through various coursework and assignments.

Lastly, a special acknowledgement also goes to my friends, family, seniors for helping me learn about the confusions and tackle them that occurred during the given timeframe.

Sincerely Yours,

Sharams Kunwar

Table of Contents

1	Chapter 1: Introduction	1
1.1	Project Description.....	1
1.2	Problem Domain	2
1.3	Current Scenario.....	3
1.4	Project as a Solution.....	4
1.5	Aim and Objectives.....	5
1.5.1	Aim	5
1.5.2	Objectives	5
1.6	Structure of the Report	6
1.6.1	Chapter 2: Background	6
1.6.2	Chapter 3: Development	6
1.6.3	Chapter 4: Testing and Analysis.....	6
1.6.4	Chapter 5: Conclusion.....	6
2	Chapter 2: Background	7
2.1	Targeted User Type.....	7
2.2	End User.....	8
2.2.1	Client Information.....	8
2.3	Understanding the Solution.....	9
2.3.1	Overview of the System.....	9
2.3.1.1.1	System Architecture	9
2.3.2	Technical terms and definitions	12
2.4	Review of similar projects.....	14
2.4.1	TenderNotice	14
2.4.2	TradeNep.....	15

2.4.3	BolpatraNepal	16
2.5	Comparison Between Systems	17
2.5.1	Analysis of the comparison.....	18
3	Chapter 3: Development	19
3.1	Considered Methodologies.....	19
3.2	Selected Methodology	19
3.2.1	Rational Unified Process (RUP)	19
3.2.2	Why RUP?	19
3.3	Phases of RUP methodology	20
3.4	Survey Results.....	26
3.4.1	Pre-Survey Results.....	26
3.4.2	Post-Survey Results	27
3.5	Requirement Analysis	28
3.5.1	Hardware Requirements.....	28
3.5.2	Software Requirements	28
3.6	Design.....	29
3.6.1	Logo	29
3.6.2	System Design (Web Application).....	30
3.6.2.1	Use Case Diagram	30
3.6.2.2	High Level Use Case Diagram	30
3.6.2.3	Entity Relationship Diagram (ERD).....	31
3.6.2.4	Data Flow Diagram (DFD).....	41
3.6.2.5	Activity Diagram	42
3.6.3	Feature Design (Web Application).....	44
3.6.3.1	Login/Register/Logout.....	44

3.6.3.1.1	Use Case Diagram	44
3.6.3.1.2	Data Flow Diagram	45
3.6.3.1.3	Activity Diagram.....	46
3.6.3.2	Search Projects	48
3.6.3.2.1	Use Case Diagram	48
3.6.3.2.2	Data Flow Diagram	49
3.6.3.2.3	Activity Diagram.....	50
3.6.3.3	Save Projects.....	51
3.6.3.3.1	Use Case Diagram	51
3.6.3.3.2	Data Flow Diagram	52
3.6.3.3.3	Activity Diagram.....	53
3.6.3.4	View Saved Projects	54
3.6.3.4.1	Use Case Diagram	54
3.6.3.4.2	Data Flow Diagram	55
3.6.3.4.3	Activity Diagram.....	56
3.6.3.5	View Project Details	57
3.6.3.5.1	Use Case Diagram	57
3.6.3.5.2	Data Flow Diagram	58
3.6.3.5.3	Activity Diagram.....	59
3.6.3.6	View Project Details	60
3.6.3.6.1	Use Case Diagram	60
3.6.3.6.2	Data Flow Diagram	61
3.6.3.6.3	Activity Diagram.....	62
3.6.3.7	Modify Predefined Keywords	63
3.6.3.7.1	Use Case Diagram	63

3.6.3.7.2	Data Flow Diagram	64
3.6.3.7.3	Activity Diagram.....	65
3.6.3.8	Manage User.....	66
3.6.3.8.1	Use Case Diagram.....	66
3.6.3.8.2	Data Flow Diagram	67
3.6.3.8.3	Activity Diagram.....	68
3.6.3.9	View and Export Saved Project Details.....	69
3.6.3.9.1	Use Case Diagram.....	69
3.6.3.9.2	Data Flow Diagram	70
3.6.3.9.3	Activity Diagram.....	71
3.7	Implementation.....	72
3.7.1	Inception	72
3.7.2	Project Elaboration.....	73
3.7.3	Project Construction.....	74
3.7.3.1	Important Screenshots (Sprint 8-11):.....	75
3.7.3.2	Important Screenshots (Sprint 12):.....	78
3.7.3.3	Important Screenshots (Sprint 13-14).....	80
3.7.3.4	Important Screenshots (Sprint 15).....	83
3.7.3.5	Important Screenshots (Sprint 16).....	86
3.7.3.6	Important Screenshots (Sprint 17):.....	87
3.7.3.7	Important Screenshots (Sprint 18).....	90
3.7.3.8	Important Screenshots (Sprint 19).....	94
3.7.4	Project Transition	96
4	Chapter 4: Testing and Analysis.....	97
4.1	Test Plan	97

4.1.1	Unit Testing, Test Plan	97
4.1.2	System Testing, Test Plan	99
4.2	Unit Testing	100
4.3	System Testing	150
4.4	Critical Analysis	163
4.4.1	Test Summary	163
4.4.2	Evaluation	163
4.4.2.1	Evaluation of Project Deliverables	163
4.4.2.2	System Evaluation	164
5	Conclusion	165
5.1	Legal, Social and Ethical Issues	166
5.1.1	Legal Issues.....	166
5.1.2	Social Issues.....	167
5.1.3	Ethical Issues	168
5.2	Advantages	169
5.3	Limitations	170
5.4	Future Work.....	171
5.4.1	Cloud-Based Auto-Scaling.....	171
5.4.2	Automated Compliance Checking System	171
5.4.3	Expand Project Scope	172
5.4.4	Multilingual Data Handling Capability	172
5.4.5	Resource Optimization Framework	173
5.4.6	Adaptive Scraping System for Dynamic Websites	173
5.4.7	User Experience and System Responsiveness Enhancement	174
5.4.8	Data Accuracy and Verification System	174

6	Chapter 6: References	175
7	Chapter 7: Bibliography.....	181
8	Chapter 8: Appendix	183
8.1	Appendix A: Pre-Survey	183
8.1.1	Pre-Survey Form.....	183
8.1.2	Sample of Filled Pre-Survey Forms.....	188
8.1.3	Pre-Survey Result	193
8.2	Appendix B: Post-Survey.....	200
8.2.1	Post-Survey Form	200
8.2.2	Sample of Filled Post-Survey Forms	206
8.2.3	Post-Survey Result.....	212
8.3	Appendix C: Sample Codes	218
8.3.1	Sample Code of the UI.....	218
8.3.1.1	Web Application	218
8.3.1.1.1	User Login/Signup Page.....	218
8.3.1.1.2	Home Page	224
8.3.1.1.2.1	Navbar	225
8.3.1.1.2.2	Search Button	226
8.3.1.1.2.3	Dashboard with Analytics.....	231
8.3.1.1.3	Filter via Organizations	236
8.3.1.1.3.1	Save Project	243
8.3.1.1.4	Filter Via Keywords	247
8.3.1.1.4.1	Save Project Button	252
8.3.1.1.5	Saved Projects	254
8.3.1.2	Admin Panel	256

8.3.1.2.1	Manage Users	257
8.3.1.2.2	Modify Keywords	257
8.3.1.2.3	View Saved Projects and Export	260
8.3.2	Sample Code for the Automation Script	264
8.3.2.1	Scrape Bots	264
8.3.2.1.1	Notable Modules	264
8.3.2.1.2	Web Driver Setup	264
8.3.2.1.3	Scraping Logics	265
8.3.2.2	Containerization	273
8.4	Appendix D: Designs	275
8.4.1	Gantt Chart	275
8.4.1.1	Gantt Chart Iterations	278
8.4.2	Work Breakdown Structure	280
8.4.3	Data Flow Diagram	281
8.4.3.1	Data Flow Diagram Iterations	282
8.4.4	Use Case Diagram	285
8.4.4.1	Use Case Diagram Iterations	287
8.4.5	Wireframes	288
8.4.6	Milestone Chart	293
8.5	Appendix E: Screenshots of the System	294
8.6	Appendix F: User Feedback	301
8.6.1	User Feedback Form	301
8.6.2	Sample of Filled User Feedback Forms	307
8.6.3	User Feedback Results	313
8.6.3.1	Summary of Results	318

8.7	Appendix G: Future Work.....	319
8.7.1	Readings for Future Work.....	319
8.7.1.1	Cloud-Based Auto-Scaling	319
8.7.1.2	Automated Compliance Checking System	319
8.7.1.3	Multilingual Data Handling Capability	320
8.7.1.4	Resource Optimization Framework.....	320
8.7.1.5	Adaptive Scraping System for Dynamic Websites	320
8.8	Appendix H: Requirements from Client	321
8.8.1	Client requirements	321
8.8.2	Client Approval Letter	322
8.9	Appendix I: Considered Methodology.....	323
8.9.1	Waterfall Methodology	323
8.9.2	Prototype Methodology	324
8.9.3	Spiral Model.....	325
8.9.4	Unified Process	326
8.10	Appendix J: SRS Document	327
8.10.1	Project Title: Scrapequest.....	327
8.10.2	Category: Web Application	327
8.10.3	Introduction	327
8.10.3.1	Purpose	327
8.10.3.2	Intended Audiences and Reading Suggestions	327
8.10.3.3	Project Scope	327
8.10.3.4	Existing System	328
8.10.3.5	Proposed System.....	329
8.10.4	Overall Description	330

8.10.4.1	System Features	330
8.10.4.1.1	Login/ Register	330
8.10.4.1.1.1	Authentication and Authorization.....	330
8.10.4.1.2	Project Discovery (User)	330
8.10.4.1.2.1	Project Search and Discovery	330
8.10.4.1.2.2	Personalized Project Management.....	331
8.10.4.1.3	Admin Management (Admin)	331
8.10.4.1.3.1	User Management.....	331
8.10.4.1.3.2	Predefined Field Configuration	332
8.10.4.1.3.3	Analytics.....	332
8.10.4.2	User Class and Characteristics	333
8.10.4.3	Operating Environment	334
8.10.4.4	Assumptions and Dependencies	334
8.10.4.5	Functional Requirements	335
8.10.4.5.1	Authentication and Authorization	335
8.10.4.5.2	Search Projects	336
8.10.4.5.3	Filter Projects	337
8.10.4.5.4	Seamless User Interaction	338
8.10.4.5.5	Project Details	339
8.10.4.5.6	Save Projects	340
8.10.4.5.7	Retrieve Saved Projects.....	341
8.10.4.5.8	Admin User Management	342
8.10.4.5.9	Modify Predefined Fields	343
8.10.4.5.10	Monitor Trends	344
8.10.4.5.11	Track Saved Projects and Export Details	345

8.10.4.6	External Interfaces Requirements.....	345
8.10.4.6.1	User Interface (UI)	345
8.10.4.6.2	Hardware	345
8.10.4.6.3	Software	345
8.10.4.7	Other Non-Functional Requirements.....	346
8.10.4.7.1	Performance Requirements	346
8.10.4.7.2	Safety Requirements	347
8.10.4.7.3	Other Software Quality Attributes.....	348
8.11	Appendix K: Legal, Social and Ethical Issues (Detailed)	349
8.11.1	Legal Issues.....	349
8.11.1.1	Global Context.....	349
8.11.1.1.1	Data Privacy and Protection	349
8.11.1.1.2	Copyright and Intellectual Property	349
8.11.1.1.3	Robots.txt	349
8.11.1.2	Context of Nepal.....	350
8.11.2	Social Issues.....	351
8.11.2.1	Privacy Concerns	351
8.11.2.2	Data Accuracy and Misrepresentation	351
8.11.2.3	Impact on Website Operations	351
8.11.3	Ethical Issues	352
8.11.3.1	Layoffs	352
8.11.3.2	Impact on Website Operations	352
8.11.3.3	Dependency and Over-reliance.....	352
8.11.3.4	Market Distortion	352
8.12	Appendix L: Advantages of the System (Detailed)	354

8.13	Appendix M: Limitations (upon scaling)	357
8.14	Appendix N: User Guide	359
8.15	Appendix O: Context Survey	370
8.15.1	Context Survey Form	370
8.15.2	Sample of Filled Context Survey Form.....	378
8.15.3	Context Survey Form Results	382
8.15.3.1	Summary of Results.....	386
8.16	Appendix P: Progress Table.....	387

Table of Figures

Figure 1 System Overview Diagram.....	9
Figure 2 Similar System, i.e., TenderNotice (TenderNotice, 2023).....	14
Figure 3 Similar System, i.e., TradeNep (TradeNep, 2023)	15
Figure 4 Similar System, i.e., BolpatraNepal (BolpatraNepal, 2023)	16
Figure 5 RUP Methodology (Levin, 2022).....	19
Figure 6 Timeline of tasks carried out during Inception and Elaboration Phase	22
Figure 7 Timeline of tasks carried out during Construction Phase	24
Figure 8 Timeline of tasks carried out during Transition Phase	25
Figure 9 Application Logo.....	29
Figure 10 Overall System Use Case Diagram of Scrapequest.....	30
Figure 11 ERD of collections in Scrapequest DB.....	31
Figure 12 ERD of models in DB SQLite	40
Figure 13 User Interaction Activity Diagram	42
Figure 14 Admin Interaction Activity Diagram.....	43
Figure 15 High Level Use-Case Diagram of Login/Register	44
Figure 16 DFD Level 2 - (Login/Register).....	45
Figure 17 Activity Diagram for Registration	46
Figure 18 Activity Diagram for Login	47
Figure 19 High Level Use Case Diagram of Searching Projects.....	48
Figure 20 DFD Level 2 - (Searching Projects)	49
Figure 21 Activity Diagram for Searching Projects.....	50
Figure 22 High Level Use Case Diagram of Saving Projects.....	51
Figure 23 DFD Level 2 - (Saving Projects)	52
Figure 24 Activity Diagram for Saving Projects.....	53
Figure 25 High Level Use Case Diagram of Viewing Saved Projects	54
Figure 26 DFD Level 2 - (Viewing Saved Projects).....	55
Figure 27 Activity Diagram for Viewing Saved Projects	56
Figure 28 High Level Use Case Diagram of Viewing Project Detail	57
Figure 29 DFD Level 2 - (Viewing Project Details – Filtered via Organizations)	58
Figure 30 DFD Level 2 - (Viewing Project Details – Filtered via Predefined Keywords).....	58

Figure 31 Activity Diagram for Viewing Project Details	59
Figure 32 High Level Use Case Diagram of Viewing Analytics	60
Figure 33 DFD Level 2 - (Viewing Analytics)	61
Figure 34 Activity Diagram for Viewing Analytics	62
Figure 35 High Level Use Case of Modifying Predefined Keywords	63
Figure 36 DFD Level 2 – (Modify Predefined Keywords).....	64
Figure 37 Activity Diagram for Modifying Predefined Keywords.....	65
Figure 38 High Level Use Case Diagram of Managing Users	66
Figure 39 DFD Level 2 - (Manage User).....	67
Figure 40 Activity Diagram for Managing Users	68
Figure 41 High Level Use Case Diagram of Viewing and Exporting Saved Project Details	69
Figure 42 DFD Level 2 - (View and Export Saved Project Details)	70
Figure 43 Activity Diagram for Viewing and Exporting Saved Project Details	71
Figure 44 Sprint 1 - Backlogs (Project Inception)	72
Figure 45 Sprint 2 - Backlogs (Project Inception)	72
Figure 46 Sprint 3 - Backlogs (Project Inception)	72
Figure 47 Sprint 4 - Backlogs (Project Elaboration)	73
Figure 48 Sprint 5 - Backlogs (Project Elaboration)	73
Figure 49 Sprint 6 - Backlogs (Project Elaboration)	73
Figure 50 Sprint 7 - Backlogs (Project Elaboration)	73
Figure 51 Sprint 8 - Backlogs (Project Construction)	74
Figure 52 Sprint 9 - Backlogs (Project Construction)	74
Figure 53 Sprint 10 - Backlogs (Project Construction)	74
Figure 54 Sprint 11 - Backlogs (Project Construction).....	75
Figure 55 Sample Scraping Logic for ADB	75
Figure 56 Data Cleaning Script.....	77
Figure 57 Sprint 12 - Backlogs (Project Construction)	78
Figure 58 Migration Script.....	78
Figure 59 Sprint 13 - Backlogs (Project Construction)	79
Figure 60 Sprint 14 - Backlogs (Project Construction)	79
Figure 61 Django Project Directory.....	80

Figure 62 Integrating Django with MongoDB.....	81
Figure 63 Fetching List of Project Filtered Via Organization	81
Figure 64 Fetching List of Project Filtered Via Organization	82
Figure 65 Sprint 15 - Backlogs (Project Construction)	83
Figure 66 Function to Sort Projects by Date.....	83
Figure 67 Function to Filter Projects Via Keywords	84
Figure 68 Function to handle seamless custom search.....	85
Figure 69 Sprint 16 - Backlogs (Project Construction)	86
Figure 70 Function to handle visualization charts.....	86
Figure 71 Sprint 17 - Backlogs (Project Construction)	87
Figure 72 Function to handle Login and Signup	87
Figure 73 Model to handle User-Saved Projects	88
Figure 74 Function to save projects and view saved projects.....	88
Figure 75 View User Saved Projects (Admin).....	89
Figure 76 Sprint 18 - Backlogs (Project Construction)	90
Figure 77 Created config file.....	90
Figure 78 Function to modify Keywords List.....	91
Figure 79 Function to handle Exporting to CSV	92
Figure 80 Function to handle exporting to PDF.....	92
Figure 81 Function to handle Logout.....	93
Figure 82 Sprint 19 - Backlogs (Project Construction)	94
Figure 83 WhiteNoise Configuration.....	94
Figure 84 Dockerfile Configuration.....	94
Figure 85 Sprint 20 - Backlogs (Project Construction)	95
Figure 86 Sprint 21 - Backlogs (Project Construction)	95
Figure 87 Sprint 22 - Backlogs (Project Transition)	96
Figure 88 Sprint 23 - Backlogs (Project Transition)	96
Figure 89 Sample of Project Data in ADB Website	100
Figure 90 Scrapped Data (ADB).....	101
Figure 91 Sample of Project Data in JICA Website	102
Figure 92 Scrapped Data (JICA).....	102

Figure 93 Sample of Project Data in SDC Website	103
Figure 94 Scraped Data (SDC)	104
Figure 95 Sample of Project Data in UKAID Website	105
Figure 96 Scraped Data (UKAID)	105
Figure 97 Sample of Project Data in UNDP Website	106
Figure 98 Scraped Data (UNDP)	106
Figure 99 Sample of Project Data in UNFAO Website.....	107
Figure 100 Scraped Data (UNFAO)	108
Figure 101 Sample of Project Data in USAID Website.....	109
Figure 102 Scraped Data (USAID).....	109
Figure 103 Sample of Project Data in WB Website.....	110
Figure 104 Scraped Data (WB).....	111
Figure 105 Script to clean the data.	112
Figure 106 Displayed Output in Terminal	113
Figure 107 Script to check connection.....	114
Figure 108 Displayed Output.....	114
Figure 109 Script to Migrate Data to DB.....	115
Figure 110 Output in Terminal.....	116
Figure 111 Screenshot of Mongo Atlas Cluster	116
Figure 112 Error upon submitting empty fields.....	117
Figure 113 Email Validation	117
Figure 114 Password Validation.....	117
Figure 115 Form Validation	118
Figure 116 Error on Empty Form Submission.....	118
Figure 117 User with Active Status	119
Figure 118 Credentials Validation.....	119
Figure 119 User with no Active Status	120
Figure 120 Authorization Validation.....	120
Figure 121 Navigating to home page without logging in.	121
Figure 122 User Recognized by the System	122
Figure 123 Search Results Container Appearing	123

Figure 124 Results Populated in Container	123
Figure 125 Results Container Disappearing	123
Figure 126 Documents Inserted in Database	124
Figure 127 Results Displayed in Dashboard.....	124
Figure 128 Projects Sorted via Date	125
Figure 129 JICA project information being displayed upon clicking JICA button.	126
Figure 130 Project information related to Migration being displayed upon clicking Migration button.	127
Figure 131 Accurately showing Filtered Projects based on Keyword.....	128
Figure 132 Accurately Displaying Related Results for selected keyword.....	129
Figure 133 UNFAO projects.....	130
Figure 134 Projects Displayed Upon Clicking UNFAO button.....	130
Figure 135 Clicking Save Project Button	131
Figure 136 Confirmation Message.....	131
Figure 137 Clicking on "View Project" Button	132
Figure 138 Page Redirection.....	133
Figure 139 Clicking on Logout Button.....	134
Figure 140 Navigated to Login Page	134
Figure 141 Navigating to Previous Page	135
Figure 142 Prompted to Login Again	135
Figure 143 Displayed Error Message	136
Figure 144 Admin was displayed an error message.....	137
Figure 145 Deleting the user.....	138
Figure 146 Confirmation dialogue.....	138
Figure 147 Confirmation message.....	138
Figure 148 Manually Adding User	139
Figure 149 Confirmation Message.....	139
Figure 150 Modifying User Account Permissions.....	140
Figure 151 Confirmation Message.....	140
Figure 152 Login Successful	140
Figure 153 Modifying Keywords.....	141

Figure 154 Confirmation Message.....	141
Figure 155 Changes Reflected in Filter via Keywords Page	142
Figure 156 Changes Reflected in Dashboard Page.....	142
Figure 157 Buttons to Export.....	143
Figure 158 Downloaded Files	143
Figure 159 Exported CSV.....	144
Figure 160 Exported PDF	144
Figure 161 Admin Logs	145
Figure 162 Admin Logged in as User.....	146
Figure 163 Creating Docker Image.....	147
Figure 164 Image ran without any error.	148
Figure 165 Export Successful	148
Figure 166 Exported file.	148
Figure 167 Loading Docker Image into another system.....	149
Figure 168 Windows Platform Running Scrapequest System	150
Figure 169 Linux Platform Running Scrapequest System.....	151
Figure 170 Displayed Analytics Dashboard	153
Figure 171 Accurate and Relevant Project Details being displayed.	154
Figure 172 Projects Filtered Via Organization being displayed.	155
Figure 173 Projects Filtered Via Keywords being displayed.....	156
Figure 174 List of Admin's Saved Projects.....	157
Figure 175 Admin Dashboard to Manage Users.....	158
Figure 176 Admin Dashboard to Modify Keywords	159
Figure 177 Admin Dashboard to View User Saved Projects and Export them.	159
Figure 178 An Error Page	160
Figure 179 Responsiveness of the App.....	161
Figure 180 Responsiveness of the App.....	162
Figure 181 Pre-Survey Form	187
Figure 182 Sample of Filled Pre-Survey Form.....	192
Figure 183 Post Survey Form	205
Figure 184 Sample of Filled Post-Survey Form	211

Figure 185 Login UI	218
Figure 186 Signup Page	219
Figure 187 Login/Signup Forms	220
Figure 188 Login/Signup Backend	221
Figure 189 Styled Login Form for error validation.	222
Figure 190 Styled Signup Form for error validation.	222
Figure 191 Homepage.....	224
Figure 192 Screenshot of Navbar	225
Figure 193 HTML & CSS for Navbar	225
Figure 194 Backend function to handle Logout.	226
Figure 195 Search Button	226
Figure 196 Search Button in Action.....	227
Figure 197 HTML Page for Search Button.....	228
Figure 198 CSS for toggling Search Button	228
Figure 199 JS for handling Search Query	229
Figure 200 Backend for Search Functionality	230
Figure 201 Dashboard with Analytics.....	231
Figure 202 HTML code for Dashboard.	232
Figure 203 JS for Generating and Styling Charts	233
Figure 204 Backend for generating data in Chart.	234
Figure 205 JSON file to load Keywords.....	235
Figure 206 Page to Display Projects Filtered Via Organizations.....	236
Figure 207 HTML Page for Filtered Via Organizations Page	237
Figure 208 HTML Page for Filtered Via Organizations Page	238
Figure 209 HTML Page for Filtered Via Organizations Page	239
Figure 210 JS to toggle Organization Buttons.....	240
Figure 211 Backend Function for Filter Via Organization Page.....	241
Figure 212 Backend Function for Filter Via Organization Page	242
Figure 213 Save Project Button	243
Figure 214 Save Project Button in Action	243
Figure 215 HTML & CSS code for "Save Project Button."	243

Figure 216 JS to handle Form Submission	244
Figure 217 JS to handle Confirmation Message	245
Figure 218 Backend Function to Save Projects	246
Figure 219 Page to Display Projects Filtered Via Keywords.....	247
Figure 220 HTML Page to Display Projects Filtered Via Keywords (Full Matches).....	248
Figure 221 HTML Page to Display Projects Filtered Via Keywords (Partial Matches).....	248
Figure 222 JS function to toggle Keywords buttons.....	249
Figure 223 Backend Function for Filter Via Keywords Page.....	250
Figure 224 Backend Function for Filter Via Keywords Page.....	251
Figure 225 Save project Button.	252
Figure 226 Save Project Button in Action	252
Figure 227 JS to handle form submission.....	253
Figure 228 Saved Projects Page.....	254
Figure 229 HTML Page for Saved Projects	255
Figure 230 Backend Function Display Saved Projects.....	255
Figure 231 Admin Panel	256
Figure 232 Admin Interface to Manage Users	257
Figure 233 Admin Panel to Modify Keywords.....	257
Figure 234 Form to Modify Keywords	258
Figure 235 Backend Function to Modify Keywords	258
Figure 236 Admin Panel for Saved Projects	260
Figure 237 Backend Function to View Saved Projects.....	261
Figure 238 Backend function to export to CSV	262
Figure 239 Backend function to export to PDF	263
Figure 240 Screenshots of Notable Modules/Packages/Libraries used for Scraping.	264
Figure 241 Setting up WebDriver for Scraping	264
Figure 242 Sample Scraping Logic for ADB.....	265
Figure 243 Scraping Logic for JICA	266
Figure 244 Scraping Logic for SDC	267
Figure 245 Scraping Logic for UKAID	268
Figure 246 Scraping Logic for UNDP	269

Figure 247 Scraping Logic for UNFAO	270
Figure 248 Scraping Logic for USAID.....	271
Figure 249 Scraping Logic for WB	272
Figure 250 Container in Action.....	273
Figure 251 Docker Image	273
Figure 252 Dockerfile.....	274
Figure 253 Gantt Chart Timeline of Project Inception	275
Figure 254 Gantt Chart Timeline of Project Elaboration.....	275
Figure 255 Gantt Chart Timeline of Project Construction (1)	276
Figure 256 Gantt Chart Timeline of Project Construction (2)	277
Figure 257 Gantt Chart Timeline of Project Transition	278
Figure 258 Gantt Chart Iteration 1	278
Figure 259 Gantt Chart Iteration 2.....	279
Figure 260 Work Breakdown Structure	280
Figure 261 Level 0 DFD for Scrapequest	281
Figure 262 Level 1 DFD for Scrapequest	281
Figure 263 Level 0 DFD for Scrapequest Iteration 1	282
Figure 264 Level 1 DFD (Users) Iteration 1.....	283
Figure 265 Level 1 DFD (Admin) Iteration 1.....	284
Figure 266 Overall Use Case Diagram Iteration 1.....	287
Figure 267 Wireframe for Login Page	288
Figure 268 Wireframe for Signup Page	288
Figure 269 Wireframe for Homepage	289
Figure 270 Wireframe for Filter via Organizations Page.....	290
Figure 271 Wireframe for Filter via Keyword Page	291
Figure 272 Wireframe for Saved Projects Page	292
Figure 273 Milestone Chart	293
Figure 274 Login UI	294
Figure 275 Signup Page	295
Figure 276 Homepage.....	296
Figure 277 Page to Display Projects Filtered Via Organizations.....	297

Figure 278 Page to Display Projects Filtered Via Keywords.....	298
Figure 279 Saved Projects Page.....	299
Figure 280 Admin Panel	300
Figure 281 User Feedback Form.....	306
Figure 282 Sample of Filled User Feedback Form	312
Figure 283 User Feedback Form Results	317
Figure 284 Waterfall Methodology (Lutkevich, 2023)	323
Figure 285 Prototyping Model (geeksforgeeks, 2023)	324
Figure 286 Spiral Process Model (Gladun, 2023)	325
Figure 287 Unified Process (Abaroa, 2019)	326
Figure 288 Loading Docker Image	359
Figure 289 Running Docker Image.....	359
Figure 290 Landing Page	360
Figure 291 Registration Page.....	361
Figure 292 Admin Panel	362
Figure 293 Guide to manage User.	362
Figure 294 Homepage Guide	363
Figure 295 Filter Via Organizations Page Guide	364
Figure 296 Filter Via Keywords Page Guide	365
Figure 297 Saved Projects Page Guide	366
Figure 298 Logging out of the system.....	367
Figure 299 Scraping Guide	367
Figure 300 Manage Users Guide	368
Figure 301 Modify Keywords Guide	368
Figure 302 Guide to View Saved Projects and Export Details	369
Figure 303 Context Survey Form	377
Figure 304 Sample of Filled Context Survey Form.....	381
Figure 305 Results of Context Survey Form	385

Table of Tables

Table 1 Comparison of Scrapequest with Similar Systems	17
Table 2 Goals & Outcomes (Inception Phase).....	20
Table 3 Goals & Outcomes (Elaboration Phase)	21
Table 4 Goals & Outcomes (Construction Phase)	23
Table 5 Goals & Outcomes (Transition Phase).....	25
Table 6 Unit Testing Plan.....	98
Table 7 System Testing Plan	99
Table 8 Unit Testing Test Case 1.....	100
Table 9 Unit Testing Test Case 2.....	112
Table 10 Unit Testing Test Case 3.....	114
Table 11 Unit Testing Test Case 4.....	115
Table 12 Unit Testing Test Case 5.....	117
Table 13 Unit Testing Test Case 6.....	118
Table 14 Unit Testing Test Case 7.....	119
Table 15 Unit Testing Test Case 8.....	120
Table 16 Unit Testing Test Case 9.....	121
Table 17 Unit Testing Test Case 10.....	122
Table 18 Unit Testing Test Case 11.....	123
Table 19 Unit Testing Test Case 12.....	124
Table 20 Unit Testing Test Case 13.....	125
Table 21 Unit Testing Test Case 14.....	126
Table 22 Unit Testing Test Case 15.....	128
Table 23 Unit Testing Test Case 16.....	130
Table 24 Unit Testing Test Case 17.....	131
Table 25 Unit Testing Test Case 18.....	132
Table 26 Unit Testing Test Case 19.....	134
Table 27 Unit Testing Test Case 20.....	136
Table 28 Unit Testing Test Case 21.....	137
Table 29 Unit Testing Test Case 22.....	138
Table 30 Unit Testing Test Case 23.....	139

Table 31 Unit Testing Test Case 24.....	140
Table 32 Unit Testing Test Case 25.....	141
Table 33 Unit Testing Test Case 26.....	142
Table 34 Unit Testing Test Case 27.....	143
Table 35 Unit Testing Test Case 28.....	145
Table 36 Unit Testing Test Case 29.....	146
Table 37 Unit Testing Test Case 30.....	147
Table 38 Unit Testing Test Case 31.....	148
Table 39 Unit Testing Test Case 32.....	148
Table 40 Unit Testing Test Case 33.....	149
Table 41 System Testing Test Case 1	150
Table 42 System Testing Test Case 2	151
Table 43 System Testing Test Case 3	152
Table 44 System Testing Test Case 4	154
Table 45 System Testing Test Case 5	155
Table 46 System Testing Test Case 6	156
Table 47 System Testing Test Case 7	157
Table 48 System Testing Test Case 8	158
Table 49 System Testing Test Case 9	160
Table 50 System Testing Test Case 10	161
Table 51 Identification of Actors	285
Table 52 High Level Use Case of Registration.....	285
Table 53 High Level Use Case of Login.....	285
Table 54 High Level Use Case of Logout.....	285
Table 55 High Level Use Case of Searching Projects	285
Table 56 High Level Use Case of Saving Projects	285
Table 57 High Level Use Case of Viewing Saved Projects	286
Table 58 High Level Use Case Diagram of Viewing Project Details	286
Table 59 High Level Use Case of Viewing Analytics.....	286
Table 60 High Level Use Case of Modifying Predefined Keywords	286
Table 61 High Level Use Case of Managing Users	286

Table 62 High Level Use Case of Viewing and Exporting Saved Project Details.....	286
Table 63 Functional Requirement Description of Authentication and Authorization.....	335
Table 64 Functional Requirement Description of Searching Projects	336
Table 65 Functional Requirement Description of Filtering Projects	337
Table 66 Functional Requirement Description of Seamless User Interface	338
Table 67 Functional Requirement Description of Project Details	339
Table 68 Functional Requirement Description of Saving Projects	340
Table 69 Functional Requirement Description of Retrieving Saved Projects	341
Table 70 Functional Requirement Description of Admin User Management.....	342
Table 71 Functional Requirement Description of Modifying Predfine Fields	343
Table 72 Functional Requirement Description of Monitoring Trends	344
Table 73 Functional Requirement Description of Tracking Saved Projects and Exporting Details	345
Table 74 Non-functional requirement description of Performance Requirements	346
Table 75 Description of Safety Requirements	347
Table 76 Description of Other Software Quality Attributes	348
Table 77 Progress Table	387

Scrapequest: Corporate Project Search

1 Chapter 1: Introduction

1.1 Project Description

The project, titled "Scrapequest," focuses on automating the project discovery process using web scraping techniques. Project discovery is crucial for organizations to gather important information about a project's vision, objectives, and scope, which is fundamental for creating high-quality software that meets the needs of end-users and stakeholders. This phase is critical for mitigating risks, fostering clarity, and setting the project's direction, enabling organizations to source projects that align with their expertise and strategic goals. However, neglecting or delaying this phase can lead to missed deadlines, budget overruns, and a failure to deliver the intended value of the product. According to McKinsey research, large IT projects often exceed their budgets by 45% and see a 56% shortfall in realized benefits (Bloch & Blumberg, 2012). Therefore, the project discovery phase is essential for ensuring that projects align with organizational goals and deliver substantial value to stakeholders.

1.2 Problem Domain

In today's competitive business environment, identifying projects aligned with corporate goals is challenging, with significant resources like time and capital overused in manual searches. A survey by Wellington found that only 22% of organizations use project management software, with 50% spending days manually collating projects (workamajig, 2022). This inefficient process not only drains resources and leads to inaccurate matches but also results in high opportunity costs, including missed chances to earn foreign revenue from international development organizations, particularly impactful in revenue-driven economies like Nepal. Furthermore, manual, and repetitive tasks contribute to low morale and increased attrition risks, as reported by 90% of respondents in an InformationWeek survey (InformationWeek, 2022). This inefficiency causes nearly 9.9% of every dollar to be wasted due to poor project performance (teamstage, 2023), placing companies at a competitive disadvantage.

The aforementioned issues represent just the surface of a much larger set of problems. The manual project discovery process presents several other challenges that can hinder a company's competitiveness:

- It requires significant time and personnel, diverting resources from core business activities.
- Manual searches often yield inaccurate results and projects that don't align with the company's capabilities or objectives.
- Delays in project identification can lead to missed business opportunities and potential revenue.
- Inefficiency in securing projects can cause companies to fall behind competitors.
- The abundance of online data can overwhelm decision-makers, complicating the separation of relevant from irrelevant information.
- Traditional methods may necessitate expensive database or subscription services, further straining budgets.

Given these issues, leveraging high-quality data and automating data collection, cleaning, and processing is essential for improving efficiency and maintaining a competitive edge. (Munappy, et al., 2020).

1.3 Current Scenario

A survey of 35 companies involved in tenders and development projects was conducted to understand the current situation of 'Project Discovery' in Nepal which revealed significant insights into 'Project Discovery' practices in Nepal, with responses from 20 companies. The majority (76.2%) of these companies operate in the Miscellaneous sector, followed by the IT, Printing, Advertising & Stationery sector at 47.6%. Notably, 86% of respondents still rely on manual searches and existing networks (71.4%) to find projects, dedicating over 20 hours per week to this task. Over half of the companies find that the projects discovered do not align well with their strategic goals, with 57% lacking a specific method for project discovery.

The challenges include time-consuming manual searches and difficulty in verifying data accuracy and relevance. Consequently, 29% of companies experience missed opportunities, and nearly half report decreased morale. Despite these issues, a strong majority (85%) are willing to invest in solutions that offer comprehensive data filtering and real-time updates, with 81% seeking to better match projects with their strategic objectives.

[Note: Click here to view [survey results](#).]

1.4 Project as a Solution

An innovative web scraping project discovery tool could significantly reduce headaches and resource drain for corporations in their search for suitable projects. Using web scraping is cost-effective, and reduces non-response errors in project discovery, making it an attractive alternative to traditional methods (Barcaroli & Nurra, 2014). Studies highlight its efficacy in delivering accurate and consistent user behavior data (Kaur & Himanshu, 2017). Similarly, web scraping provides timely information and overcomes limitations such as non-response errors found in traditional methods (Dongo & Cardinale, 2021). Web scraping is a highly useful tool in the information age, and an essential one in the modern fields (Simajuntak & Traigan, 2022).

In Nepal, platforms like Bolpatra Nepal already utilize scraping to identify government tenders, easing the discovery process (BolpatraNepal, 2023). Scrapequest builds on this foundation with features specifically tailored to the needs of organizations addressing forementioned problem in ways below:

- Utilizes algorithms to align projects with organizational goals and expertise, simplifying alignment with strategic priorities.
- Automation reduces the time and resources spent on manual searches, making project discovery more cost-effective.
- Provides near real-time updates, minimizing missed opportunities and missed revenues.
- Offers solutions for efficient data filtration and management to handle sifting huge pool of data sources.
- Automates repetitive tasks, boosting team morale which is often dampened by manual project discovery processes.

1.5 Aim and Objectives

1.5.1 Aim

The Scrapequest project aims to revolutionize IT organization project discovery by utilizing web scraping techniques, and automation, thereby enhancing project discovery processes.

1.5.2 Objectives

The S.M.A.R.T objectives of the project are:

- Implement web scraping using Python libraries like Selenium, Scrapy, and BeautifulSoup to automate the collection of project data from various websites.
- Utilize MongoDB for data management to organize and process large volumes of scraped project data efficiently.
- Use Django to create robust web applications that are scalable and efficient, ensuring quick deployment and easier maintenance, which will help in managing and displaying scraped data effectively.
- Implement Docker for containerization to standardize the application environment across different development and production platforms, enhancing reliability and ease of deployment.
- Through practical application of the above technologies, author will enhance his technical skills in web development, data management, and application deployment, preparing him for complex projects in the future.

1.6 Structure of the Report

This section outlines the comprehensive framework of the report.

1.6.1 Chapter 2: Background

This section provides an in-depth review of project-related areas, including client information, similar projects, and analyses to aid understanding.

1.6.2 Chapter 3: Development

This section discusses the methodologies employed for project development, covering the development phases, implementation, survey analysis, and design elements.

1.6.3 Chapter 4: Testing and Analysis

This section focuses on testing, including unit and system tests to verify that the system meets its specifications.

1.6.4 Chapter 5: Conclusion

This section summarizes the project, covering legal, social, and ethical issues it touches, its advantages and limitations.

2 Chapter 2: Background

Scrapequest is a project discovery platform that aids in identifying and exploring international development projects in fields like food science, digitalization, and education. It uses web scraping and automation techniques to improve efficiency and enhance project discovery stages. The platform extracts project information from prominent donors like UNDP, UNFAO, USAID, World Bank, ADB, JICA, SDC, and UK Aid.

2.1 Targeted User Type

Scrapequest is exclusively tailored for Vertex Special Technologies, serving as a dedicated tool for their specific needs. The platform is designed to be utilized solely by the client, Vertex Technologies, providing them with a streamlined solution for project discovery. However, with further customization and scaling, Scrapequest has the potential to extend its utility to a broader audience, including individuals and organizations involved in international development, project management, and research, as it provides comprehensive information on projects related to food science, digitalization, and education, etc., **tailored to specific need** of the client. It could help users efficiently discover relevant projects, aligning with their goals, interests, and expertise.

2.2 End User

2.2.1 Client Information

Vertex Special Technologies



Description of Client

The client for the project is Vertex Special Technologies. They are a leading digital transformation company offering services such as staffing, software development, digital transformation, and AI services located in Sanepa, Lalitpur. They focus on delivering exceptional service and staying up to date with the latest trends and technologies to help businesses achieve their goals (VertexSpecial, 2023).

They work in international development projects as well. In their pursuit of discovering international development projects, they found that the traditional methods employed to identify projects aligned with their organizational goals were cost ineffective. The sheer volume of projects further exacerbated the situation, resulting in significant time consumption and, consequently, missed deadlines.

Thereafter, a proposal for a web application which would scrape information from donors' websites, making it faster and more cost-effective to shortlist projects aligned with their organizational objectives was sent to the company.

[Note: More about Client Requirements in [Appendix](#)]

2.3 Understanding the Solution

Scrapequest is a digital solution for project discovery and information gathering in international development, transforming the process from manual data collection to digitalization. It offers automated scraping and advanced search functionalities, promoting quicker access to relevant information, strategic decision-making, and a heightened user experience. Focusing on sectors like food science, digitalization, and education, the platform provides secure user authentication, robust authorization mechanisms, and data privacy regulations. It automates scraping project information from organizations like the United Nations and World Bank, offering a user-friendly web interface. **Scrapequest** ensures efficiency, scalability, and industry best practices. Upon completion, it will exceed expectations, allowing clients to search, discover, and analyze international development projects with ease and efficiency.

2.3.1 Overview of the System

2.3.1.1 System Architecture

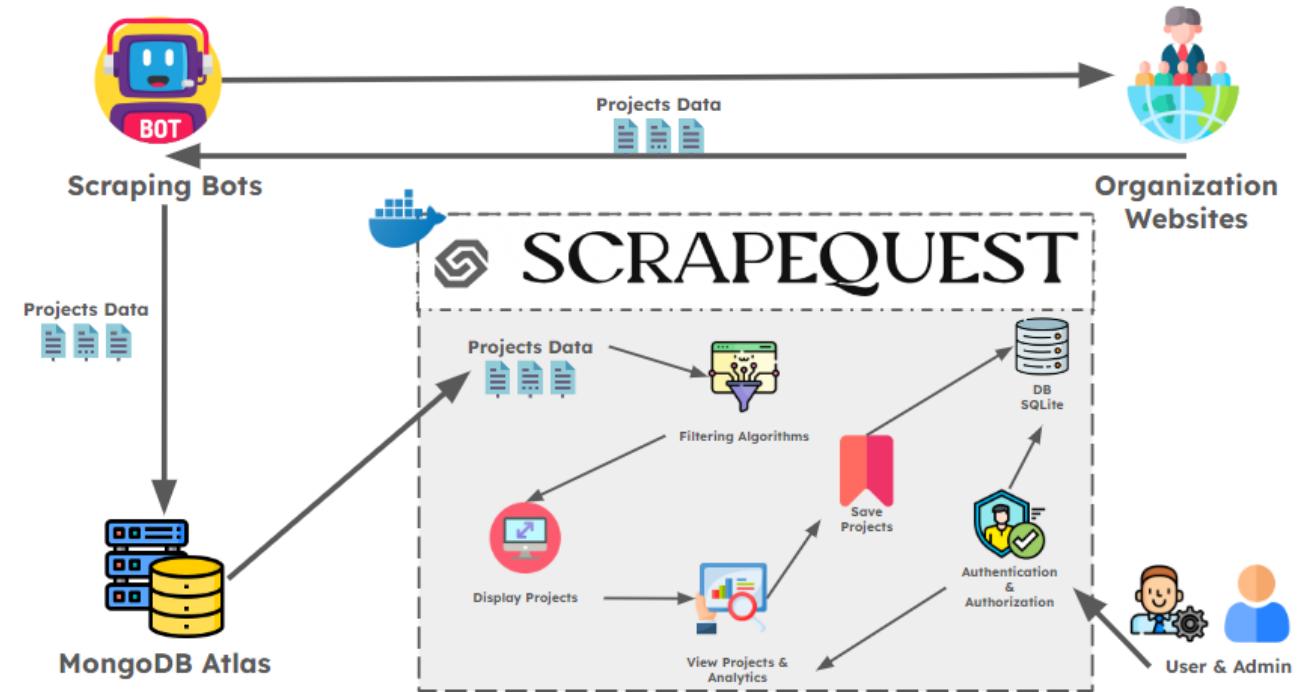


Figure 1 System Overview Diagram

The following are the major module components of the system that paved the way for the complete framework of the project.

Scraping Bots

Scraping Bots are automated programs designed to extract data from organization websites as per the requirements from the client. They navigate through web pages, identify the relevant data, and collect it for further processing.

MongoDB Atlas

This is a fully managed cloud database service provided by MongoDB. Here, it is used as the storage solution to keep the scraped projects data. It's known for its scalability and flexibility in handling large volumes of data.

SCRAPEQUEST System:

- Filtering Algorithms: These algorithms are used to process the raw scraped data to tailor according to client's needs.
- Save Projects: This would take the filtered projects data and store it in a structured format, within the SCRAPEQUEST system's own database.
- Display Projects: This would be responsible for presenting the filtered project data or saved project data in a user-friendly format, through a web interface.
- View Projects & Analytics: This feature allows users and admins to not only view the listed projects but also perform analytical operations to extract insights.
- Authentication & Authorization: This ensures that only authenticated and authorized users and admins can access the system and its data.

[Note: A comprehensive User Guide is in [appendix](#) to understand the system better.]

Working Mechanism:

1. The **Scraping Bots** begin the process by visiting various Organization Websites and collecting Projects Data.
2. The collected data is then stored in **MongoDB Atlas**, which serves as an initial repository.
3. Within the SCRAPEQUEST system, **Filtering Algorithms** process the raw data from **MongoDB Atlas**, ensuring that only relevant and accurate project data is retained.
4. After filtering, the data is saved into a structured database, **SQLite DB**, where Projects can be easily managed and queried.
5. The **Display Projects** function retrieves data from both the **SQLite DB** and **MongoDB Atlas** and presents it in a format that is accessible to users and admins, taking care to respect the Authentication & Authorization protocols.
6. **Users & Admins** can then **View Projects & Analytics**, which implies that they have the ability to not only see the project listings but also gain insights, using built-in functions within **SCRAPEQUEST**.

2.3.2 Technical terms and definitions

- **IDE: Virtual Studio Code**

VS Code is a popular free and open-source code editor developed by Microsoft, known for its cross-platform compatibility, rich feature set, extensibility, lightweight design, active community support, and accessibility. It offers syntax highlighting, IntelliSense, debugging support, Git integration, and built-in terminal, making it an ideal choice for developers across various domains (Heller, 2023).

- **Programming Language: Python**

Python is a high-level programming language with dynamic semantics, making it ideal for Rapid Application Development and scripting to connect existing components (Python, 2023). It offers powerful web scraping libraries, data processing and manipulation through libraries like Pandas, web application development with frameworks like Django, Pymongo for integration with MongoDB, automation and scripting capabilities, and a robust community of developers for ongoing support and development resources. Henceforth, it has been chosen for development of **Scrapequest**.

- **Python Libraries: BeautifulSoup, Selenium**

Selenium and BeautifulSoup are Python libraries used in web scraping, which also has been used in the development of **Scrapequest**. Selenium automates user interactions with web browsers, enabling the extraction of project information from dynamic websites (WebscrapeAI, 2023). BeautifulSoup, on the other hand, parses HTML and XML files, simplifying the process (compucademy, 2023). Together, they allow for seamless extraction of project details from dynamic content, making them a powerful tool for scraping dynamic websites.

- **Web Application Framework: Django**

Django is a high-level Python web framework that simplifies web application development by focusing on reusability, less code, and the "Don't Repeat Yourself" principle (Mozilla, 2023). Django is used in development of **Scrapequest** for efficient web application development, scalability, database management, admin interface, and security compliance.

- **Database: MongoDB**

MongoDB is a NoSQL document-oriented database used in development of **Scrapequest** due to its flexible, JSON-like structure, scalability, efficient data retrieval, JSON-like queries, and geospatial capabilities (MongoDB, 2023). Its design allows for easy adaptation to changing data requirements, scalability across multiple servers, and compatibility with diverse data from international development projects.

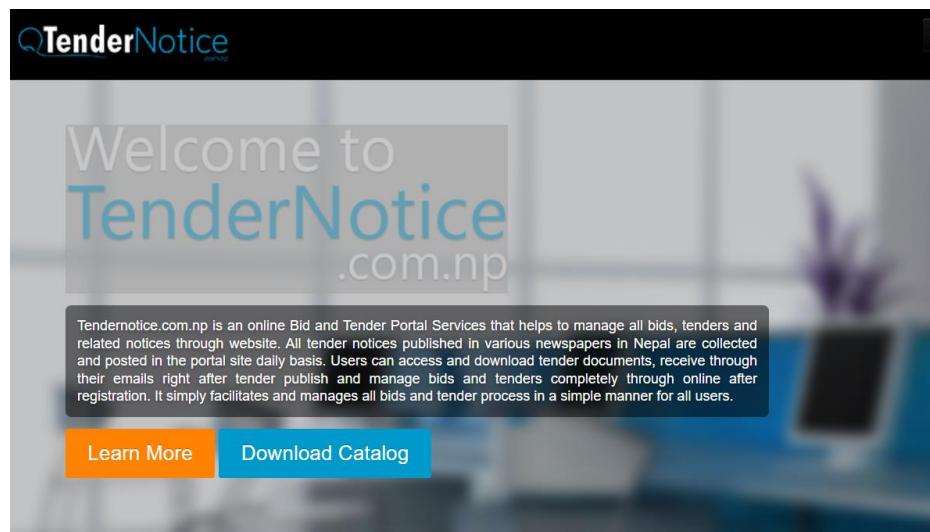
- **Containerization: Docker**

Docker is a platform that automates application deployment in lightweight containers, ensuring consistency across different environments (Abdo, 2023). This makes it ideal for projects like **Scrapequest** due to its ability to manage dependencies, providing a consistent environment across different stages. It simplifies deployment by encapsulating the entire application and its dependencies into a container, streamlining the process, and making the development, testing, and deployment of **Scrapequest** more efficient and reliable.

2.4 Review of similar projects

2.4.1 TenderNotice

TenderNotice is an online platform for Nepali bid and tender notices, offering features such as daily collection and posting of notices, profile creation and update, filter options, and tender management. Tender Notices from Government, Private and Non-Government and any other institutions are posted online, and users can access and download tender documents (TenderNotice, 2023). They can also create their own profiles and keep information on bidding, competitors, and relevant documents. The system alerts users as the tender deadline approaches. Users can also view tender notices in one place, facilitating the bidding process and providing information on invitations, quotations, requests, expressions of interest, and enlistment.



Latest Tenders / Notices

Nepali Date English Date

S.No.	Notice Publisher	Description	Published Date	Last date of Submission	Notice Category	Industry	Product/S
1	Alternative Energy Promotion	AEPC/DKTI/SIPS/NCB/G/2023/24-01 Supply, Delivery, Installation, Testing and Commissioning of 2 KWP and 3 KWP Solar	2023-12-29	2024-01-29 12:00	Tender	Government/ Ministries/ Departments	GOODS

Figure 2 Similar System, i.e., TenderNotice (TenderNotice, 2023)

2.4.2 TradeNep

TradeNep is a platform for finding and publishing tender notices in Nepal, covering all sectors including government, non-government, private, and institute. It offers features such as search and filter, subscription, and alert, bid and tender, and access to news and research related to tendering and procurement in Nepal and abroad. Users can search and filter tender notices by category, location, organization, deadline, and keyword. They can also subscribe to the platform via email or SMS to receive alerts when new notices are published or updated (TradeNep, 2023).

नेपाल भरीबाट प्रकाशित सबै प्रकारका टेन्डर तथा बोलपत्र सँग सम्बन्धित सुचनाहरु : Tender Notice in Nepal

S.NO	CALLER'S NAME & ADDRESS SUBMISSION DATE	ITEMS / DESCRIPTION	PUBLISHED DATE
1	Ministry of Home Affairs, Department of Immigration, Kathmandu <i>29/01/2024</i>	Supply, Delivery, Installation , Support and Upgradation of DC/NLDC/DR Infrastructure	<i>31/12/2023</i>

Figure 3 Similar System, i.e., TradeNep (TradeNep, 2023)

2.4.3 BolpatraNepal

BolpatraNepal is a public procurement portal in Nepal that provides information on various tenders and projects across various sectors and regions. Users can search for tenders by keywords, categories, locations, dates, and other criteria, sort results by relevance, status, and budget. They can view and download tender details, such as tender ID, description, budget, deadline, eligibility, and contact information, and download tender documents like notice, invitation, and bidding forms. Users can subscribe to the portal and receive email alerts for new tenders, customize their profile and dashboard, and provide feedback and support to portal administrators and technical support. They can also access FAQ and help sections for more information (BolpatraNepal, 2023).

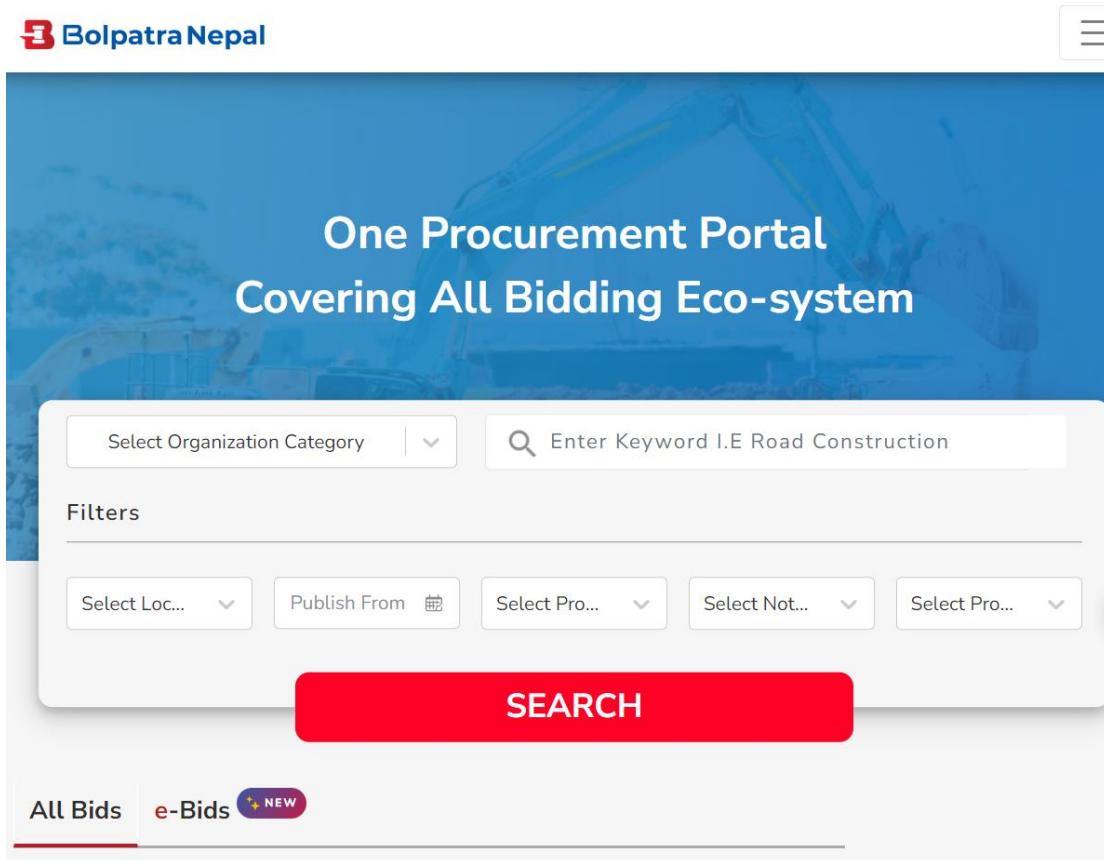


Figure 4 Similar System, i.e., BolpatraNepal (BolpatraNepal, 2023)

2.5 Comparison Between Systems

Features	Scrapequest	BolpatraNepal	TradeNep	TenderNotice
Scope	International Development Projects	Public Procurement in Nepal	Tender notices in Nepal	Nepali bid and Tender Notices
Donors Covered	Multiple International Organizations	Nepali public procurement	All Sectors	Government, Private and Non-Government Sector
Fields Aligned	Specific sectors as per Client's Requirements	Various Sectors	All Sectors	Not Specified
Search And Filter	Predefined Keywords tailored according to client's needs	Keywords, Categories, Locations, Dates, etc.	Categories, Location, etc.	Date
Data Collection using Automated Scraping	Yes	No	No	No
Export Data	Yes	Yes	Yes	Yes
Visualization	Yes	No	No	No

Table 1 Comparison of Scrapequest with Similar Systems

2.5.1 Analysis of the comparison

After comparing Scrapequest with BolpatraNepal, TradeNep, and TenderNotice, several conclusions can be drawn. Scrapequest stands out for its international focus, targeting development projects from various global organizations as per the client's specific needs, while BolpatraNepal, TradeNep, and TenderNotice primarily focus on the Nepali public procurement sector. It specifically targets projects from well-known international organizations such as the UN, World Bank, and others, aligning with specific fields like food science, digitalization, and education, offering a more targeted approach compared to the broader scopes of BolpatraNepal, TradeNep, and TenderNotice. Scrapequest also utilizes automated scraping, ensuring a systematic and reliable data collection process. BolpatraNepal, TradeNep, and TenderNotice offer search and filter options but with a broader focus while Scrapequest narrows it down according to the client's requirements. It also includes an analytics dashboard for tracking project availability trends, keywords, and preferred organizations, providing valuable insights for strategic decision-making.

In short, Scrapequest offers an exclusive and specialized solution for international development project discovery with a strong emphasis on data accuracy and ethical considerations. Its features, such as a targeted focus, analytics dashboard, and comprehensive admin panel, distinguish it from platforms like BolpatraNepal, TradeNep, and TenderNotice, which are more tailored to the specific context of Nepali public procurement. The composition of all these features aims to cater to specific requirements and expectations of the client.

3 Chapter 3: Development

3.1 Considered Methodologies

[Note: Explained in detail in [Appendix](#).]

3.2 Selected Methodology

3.2.1 Rational Unified Process (RUP)

The methodology used for the project development is RUP methodology. It's an incremental and iterative approach to software development that involves adjusting and repeating cycles until objectives are met (Minott, 2022).

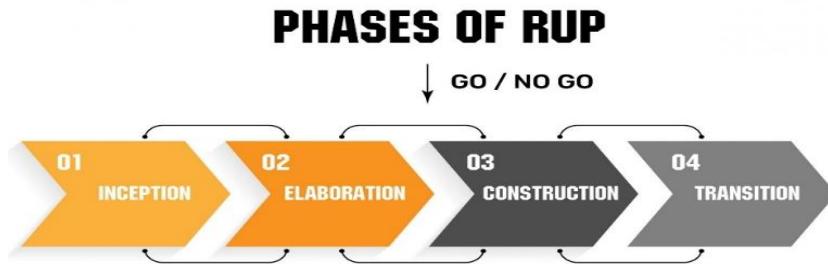


Figure 5 RUP Methodology (Levin, 2022)

3.2.2 Why RUP?

- **User Requirements:** Chosen for its strong focus on user requirements and ability to deliver high-quality software on a predictable schedule.
- **Iterative Approach:** Necessary for frequent progress meetings with supervisors, allowing for continuous refinement based on feedback.
- **Revisiting Phases:** RUP's iterative nature enables revisiting and modifying phases to continuously enhance the software.
- **Code Reuse:** Facilitates the reuse of code between components, optimizing both time and resources in a time-constrained project like this.
- **Continuous Testing:** Integral for controlling system changes; RUP incorporates continuous testing throughout development, aiding in early error detection and change management.
- **Documentation Emphasis:** Meets the project's heavy documentation requirements, making RUP highly suitable for academic projects.

3.3 Phases of RUP methodology

Implementation of this methodology in the project development involves the following four phases. According to the goal of each phase, tasks were broken down into manageable chunks and tracked in Jira Board.

1. Inception:

Overall Goal	<ul style="list-style-type: none"> - Establish a clear and shared understanding of the project's purpose, scope, and objectives among all stakeholders and develop a foundational framework for planning and executing the project
Expected Overall Outcome	<ul style="list-style-type: none"> - A document defining the project's vision, scope, objectives, stakeholders, and key success criteria. - Identification of client, their interests, and plans for their engagement and communication throughout the project. - An outline of the project's approach, phases, timelines, resources, and preliminary risk assessments, providing a roadmap for project execution. - Early identification of potential risks and initial strategies for mitigation, ensuring project resilience and adaptability.

Table 2 Goals & Outcomes (Inception Phase)

2. Elaboration:

Overall Goal	<ul style="list-style-type: none"> - To refine and expand upon the initial project framework by detailing the project proposal, developing a comprehensive project plan, and identifying potential risks, thereby setting a solid foundation for the project's execution and management.
Expected Overall Outcome	<ul style="list-style-type: none"> - A comprehensive document that elaborates on the project's objectives, scope, methodology, and expected outcomes, providing a clear roadmap for the project's execution. - An in-depth plan covering all aspects of project management, including detailed schedules, resource allocation, budgeting, quality control measures, communication strategies, and stakeholder engagement plans. - Identification and documentation of major project risks, along with their potential impacts and detailed mitigation strategies, ensuring preparedness for dealing with uncertainties.

Table 3 Goals & Outcomes (Elaboration Phase)

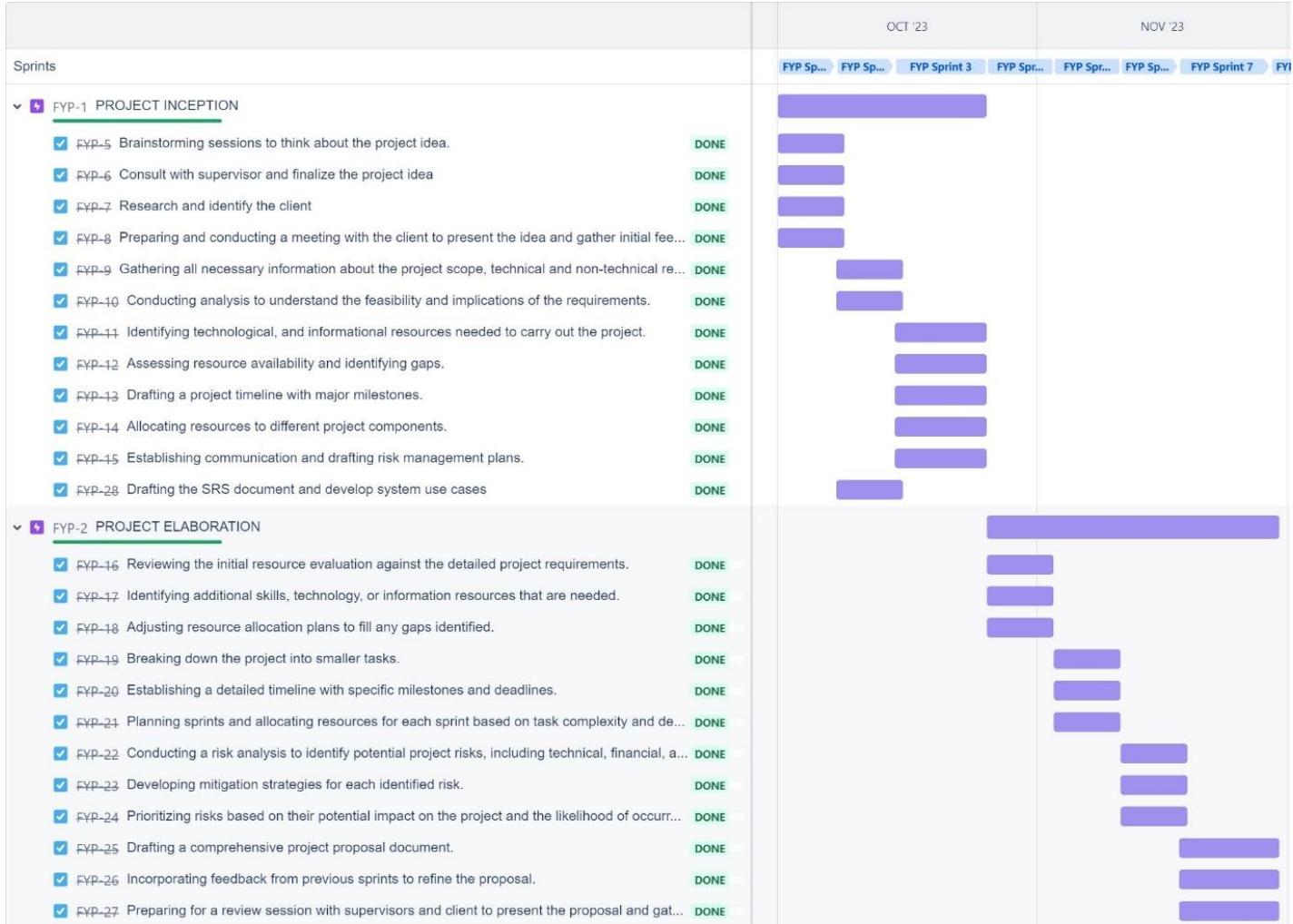


Figure 6 Timeline of tasks carried out during Inception and Elaboration Phase

3. Construction:

Overall Goal	<ul style="list-style-type: none"> - To methodically construct a system by completing the development of all necessary components, ensuring they function seamlessly together, and rigorously testing the final product for quality assurance before deployment.
Expected Overall Outcome	<p>Completed Data Collection:</p> <ul style="list-style-type: none"> - Successful scraping of necessary web data. - Fully developed databases ready for integration with system components. <p>Fully Designed System:</p> <ul style="list-style-type: none"> - A robust system architecture that serves as a blueprint for construction. - Detailed mock-ups guiding the user interface development. - Wireframe designs that provide a visual guide for the system's layout and navigation. <p>Constructed System Components:</p> <ul style="list-style-type: none"> - Front-end components built to specification, offering a user-friendly interface. - Back-end components developed to manage data processing and system operations effectively. <p>System Testing:</p> <ul style="list-style-type: none"> - Black Box Testing completed, ensuring the system meets all functional requirements without delving into internal code structure. - White Box Testing conducted, verifying the internal workings of the system, and ensuring code quality.

Table 4 Goals & Outcomes (Construction Phase)

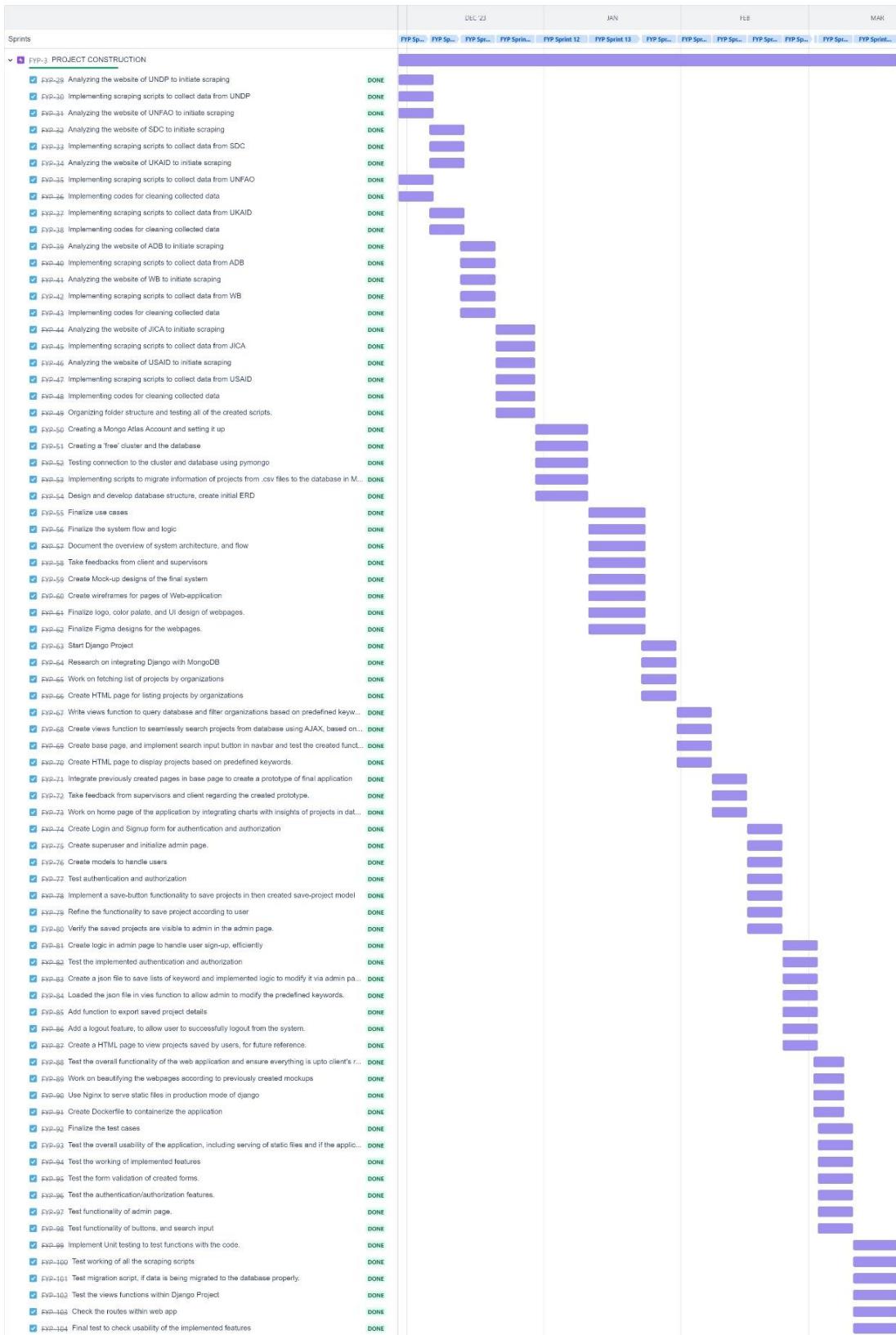


Figure 7 Timeline of tasks carried out during Construction Phase

4. Transition:

Overall Goal	To seamlessly transition the completed project to the client, in this case, Vertex Special Technologies, providing all necessary documentation, incorporating feedback through upgrades, and ensuring the system is fully tested and any issues are resolved for optimal operation.
Expected Overall Outcome	<ul style="list-style-type: none"> - Detailed documentation of the entire project, including system design, manuals, and maintenance guides, to be provided to the client for reference and future support. - The fully operational project is delivered to Vertex Special Technologies, ensuring they have everything needed to use and maintain the system effectively. - Analysis of initial feedback from Vertex Special Technologies is used to prioritize and implement system upgrades, enhancing functionality and user satisfaction. - Completion of extensive system testing to confirm stability and performance, along with the resolution of any outstanding issues, ensuring the system is reliable and ready for full-scale operation.

Table 5 Goals & Outcomes (Transition Phase)

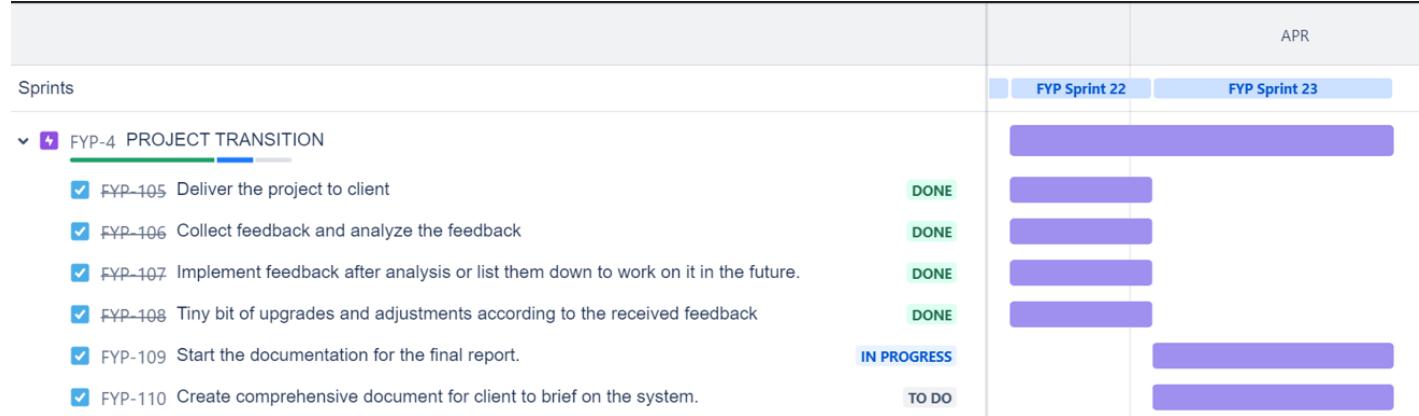


Figure 8 Timeline of tasks carried out during Transition Phase

3.4 Survey Results

3.4.1 Pre-Survey Results

The pre-survey form, which received 14 responses from employees of companies that might be potential clients of Scrapequest, highlighted various facets of the project discovery process within these organizations. Below are the insights:

- Majority of respondents were somewhat involved in the project discovery process, with an equal distribution among those who led, not involved, and very involved.
- A majority spend 5 to 10 hours weekly on project discovery tasks, with some feeling the time spent is just right (21.4%) or somewhat excessive (7.1%).
- The most common challenges include identifying projects aligned with the company's strategy and managing large volumes of data.
- All respondents use manual searches through websites and databases, while a significant number also attend industry networking events.
- Respondents unanimously value real-time updates and data filtering to match projects with strategy as the most beneficial features in a project discovery tool.
- The process impacts morale and workload somewhat negatively for most (42.9%), with a smaller number indicating positive (28.6%) or no significant impact (35.7%).
- Respondents are seeking for better project-strategy alignment and faster project identification.

[Note: Details of survey are placed in [appendix](#)]

3.4.2 Post-Survey Results

The post-survey form surveyed respondents of pre-survey form after thorough demonstration which gathered 13 responses, providing feedback on the Scrapequest tool:

- All respondents had a positive initial impression of Scrapequest.
- The interface was found to be very intuitive by the vast majority.
- Real-time project updates and integration with current IT systems were highly appreciated.
- All participants found Scrapequest's features innovative and beneficial.
- Some respondents suggested including projects from additional sources and improving app speed.
- Nearly all believe Scrapequest would save them a moderate amount of time in the project discovery process.
- Similarly, feel that Scrapequest would somewhat improve the accuracy of project discovery.
- Most participants are comfortable using Scrapequest following the demonstration.
- The most preferred training resources are in-depth user manuals and online tutorials.
- Almost all respondents are likely to recommend implementing Scrapequest.

[Note: Details of survey are placed in [appendix](#)]

3.5 Requirement Analysis

Requirement analysis is the process of gathering and documenting client specifications for a software product, identifying their desires, expectations, and objectives, and transforming them into functional and non-functional specifications which will serve as guidelines for the development process (Awati, 2023).

A complete specification and description of requirements for successful development of the system has been documented in SRS document.

[Note: Full description of [SRS document](#) is in Appendix section.]

Other requirements for the completion of project are listed below:

3.5.1 Hardware Requirements

A PC equipped with either a WIFI or Ethernet connection, operating on Windows 10 or Ubuntu as the OS, meeting the following requirements:

- Processor: Intel Core i5 or higher, or an equivalent
- Processor Speed: At least 2.00 GHz or higher
- RAM: 4 GB or more

3.5.2 Software Requirements

- **IDE: Virtual Studio Code**
- **Programming Language: Python**
- **Python Libraries: BeautifulSoup, Selenium**
- **Web Application Framework: Django**
- **Database: MongoDB/ SQLite**
- **Containerization: Docker**
- **VirtualBox: VMware Workstation Pro**
- **Version Control: GitHub**
- **Designing and Developing Wireframes: Balsamiq**
- **Developing Diagrams and Charts: Draw.io, Microsoft Visio**
- **Developing Gantt Chart: Instagantt**
- **Project Management Tool: Jira**

3.6 Design

Designing the development workflow is crucial before building the application. Understanding features ahead of time simplifies the development process. To provide a high-level overview of the project, UML diagrams were created highlighting detailed blueprint for its features.

3.6.1 Logo



Figure 9 Application Logo

The logo is purposely made minimalistic with modern, clean font with bold characters emphasizing strength and dependability. Its simplicity makes it scalable and easily identifiable. Also, the grayscale color palette used in the logo implies a sleek, modern, and professional look which highlights versatility, strengthening the brand's identity and connecting with the target audience.

3.6.2 System Design (Web Application)

3.6.2.1 Use Case Diagram

A use case diagram captures system users and interactions, using specific symbols and connectors to effectively visualize system functionality (TechTarget, 2023).

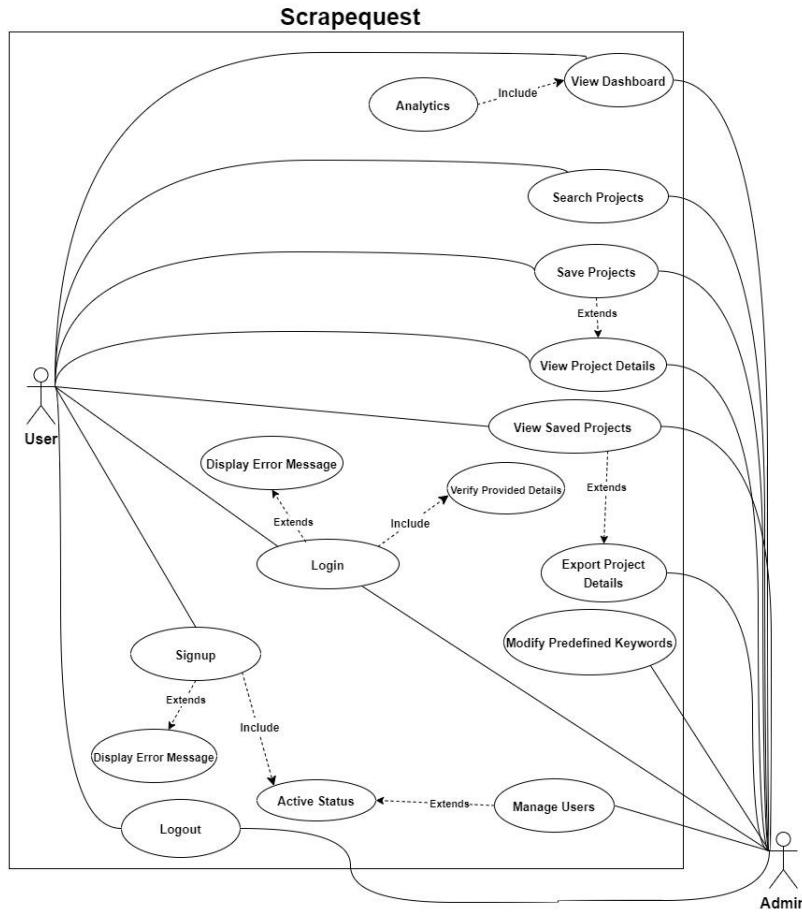


Figure 10 Overall System Use Case Diagram of Scrapequest

3.6.2.2 High Level Use Case Diagram

[Note: Full Description of High-Level Use Case Diagram is described in [appendix](#).]

3.6.2.3 Entity Relationship Diagram (ERD)

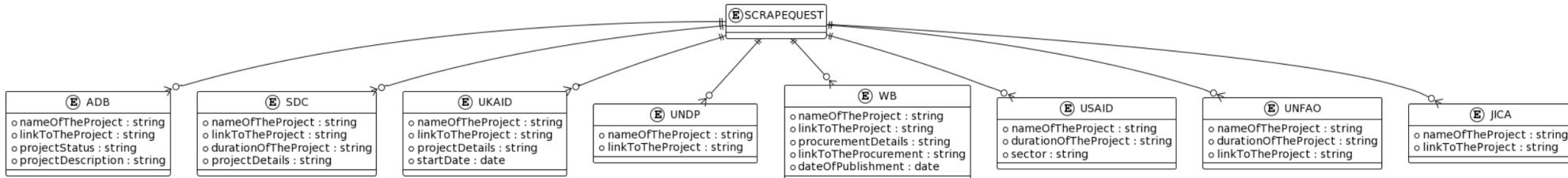


Figure 11 ERD of collections in Scrapequest DB

The MongoDB ERD represents collections within the SCRAPEQUEST database, where each entity corresponds to a different collection. These collections store information related to projects from various organizations like ADB, SDC, UKAID, UNDP, WB, USAID, UNFAO, and JICA.

Why MongoDB Was Used:

MongoDB, being a NoSQL database, is chosen for systems where the data model can vary and is not strictly structured. It is schema-less, i.e., various data types can be stored and accessed on the fly (PureStorage, 2024). This is beneficial when:

- The data does not fit well into tables and rows of traditional relational databases.
- The schema is expected to evolve over time.
- You need to handle large volumes of data with high performance and scalability.
- You want to leverage the flexibility of JSON-like documents that can store array and nested structures.

For the SCRAPEQUEST project, MongoDB is chosen only to store the project data because the data scraped from different sources have varying fields and a flexible schema, making MongoDB an excellent choice to accommodate these requirements without the constraints of a fixed schema.

Entities and Attributes:**Entity: 'ADB'****Attributes:**

- 'Name of the Project'
- 'Link to the project'
- 'Project Status'
- 'Project Description'

Entity: 'SDC'**Attributes:**

- Name of the Project
- Link to the project
- Duration of the Project
- Project Details

Entity: 'UKAID'**Attributes:**

- Name of the Project
- Link to the project
- Start Date
- Project Details

Entity: 'UNDP'**Attributes:**

- Name of the Project
- Link to the project

Entity: 'WB'**Attributes:**

- Name of the Project
- Link to the project
- Procurement Details
- Link to the Procurement
- Date of Publication

Entity: 'USAID'**Attributes:**

- Name of the Project
- Duration of the project
- Sector

Entity: 'UNFAO'**Attributes:**

- Name of the Project
- Duration of the Project
- Link to the project

Entity: 'JICA'**Attributes:**

- Name of the Project
- Link to the project

Relationships:

The ERD shows a one-to-many relationship from SCRAPEQUEST to each collection. This implies that SCRAPEQUEST can associate with many instances of these collections, although this is a conceptual representation. In MongoDB, collections don't have strict relationships like in relational databases; the ERD is using relational notation for the sake of understanding the logical grouping of collections under the database.

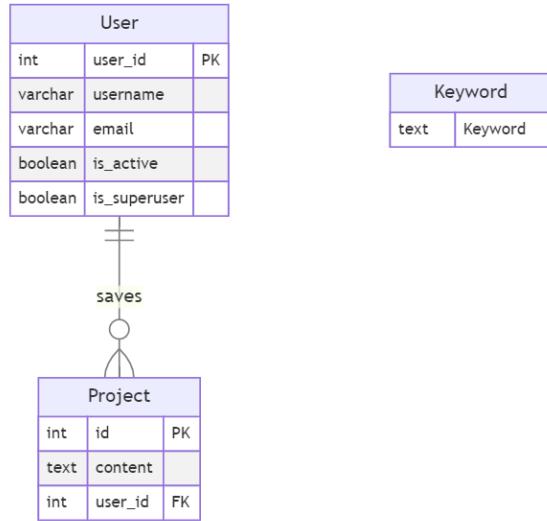


Figure 12 ERD of models in DB SQLite

Models and Attributes:

User: Represents the user of the system with attributes for identification and access control (id, username, email). The is_active and is_superuser fields indicate whether a user has active rights or admin rights, respectively.

Project: Represents the saved projects with a foreign key user_id linking it to a User. The content field stores the details of the project.

Relationship between User and Project: The relationship between User and Project is one-to-many, as one user can save many projects, but each project is associated with only one user.

Keyword: Keyword model stores keywords in the database, while it has subclass 'KeywordUploadProxy' which is a proxy model which allows behavior like uploading keywords without creating a new table, which proves to be useful when admin modifies predefined fields.

The MongoDB ERD is conceptual and represents collections and their attributes within a NoSQL database, while the SQLite ERD illustrates actual table relationships in a relational database managed by Django's ORM. MongoDB is used for its schema flexibility and scalability, while SQLite is used within Django for its simplicity and relational structure.

[Note: 'Django' library was also explored to integrate everything into a single database, but upon researching it further, many issues had been reported already, so it wasn't opted for.]

3.6.2.4 Data Flow Diagram (DFD)

A Data Flow Diagram (DFD) is a visual representation of how data moves through a system, illustrating processes, data stores, external entities, and data flows. It is used for system analysis and design to model the flow of data and provide a visual representation of the system's architecture (Nolle, 2023).

Here, Level 0 DFD offers a holistic view of the **Scrapequest** system, while Level 1 provides a more granular understanding of processes. The DFDs are attached in [appendix](#) section.

3.6.2.5 Activity Diagram

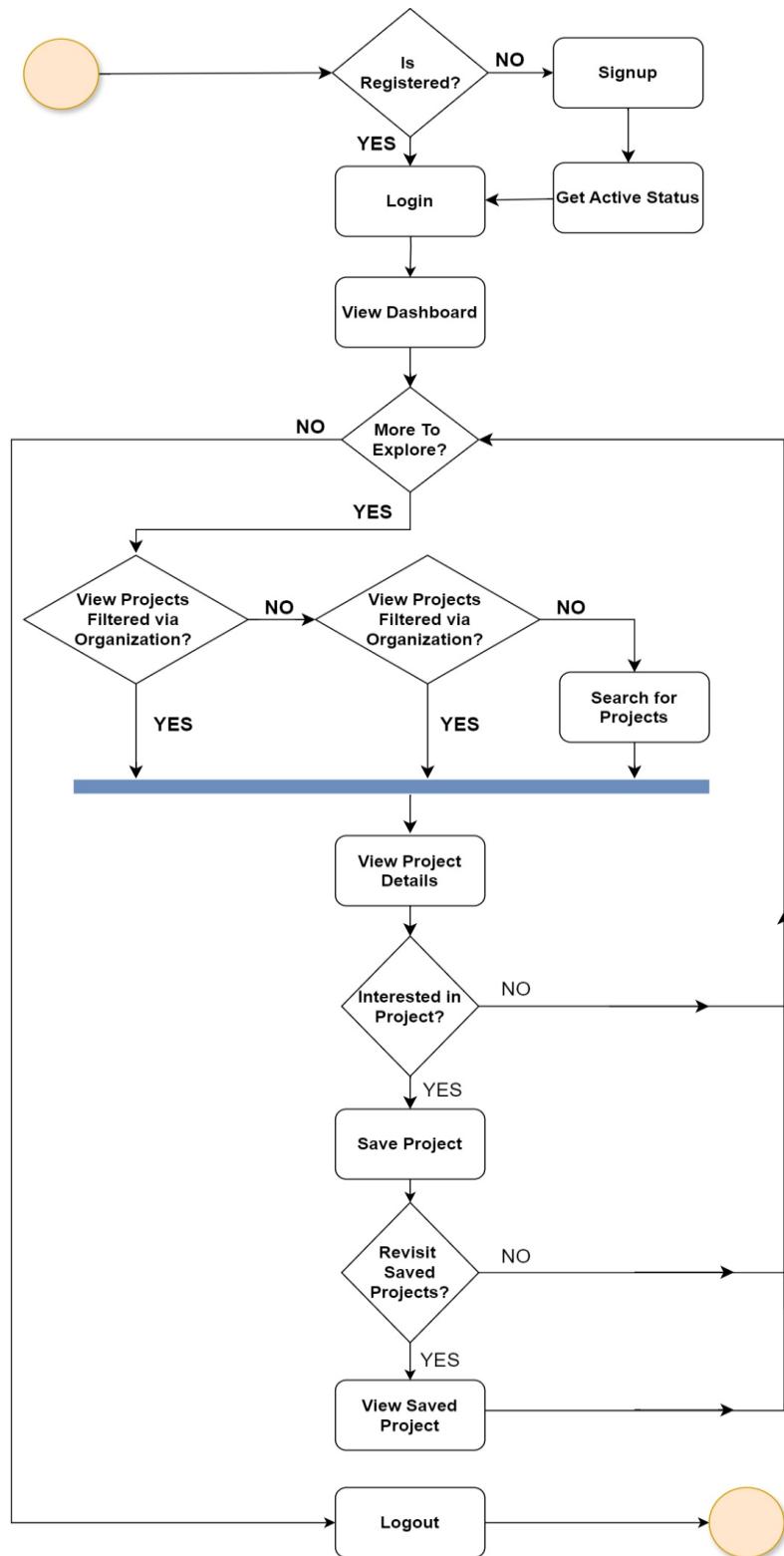


Figure 13 User Interaction Activity Diagram

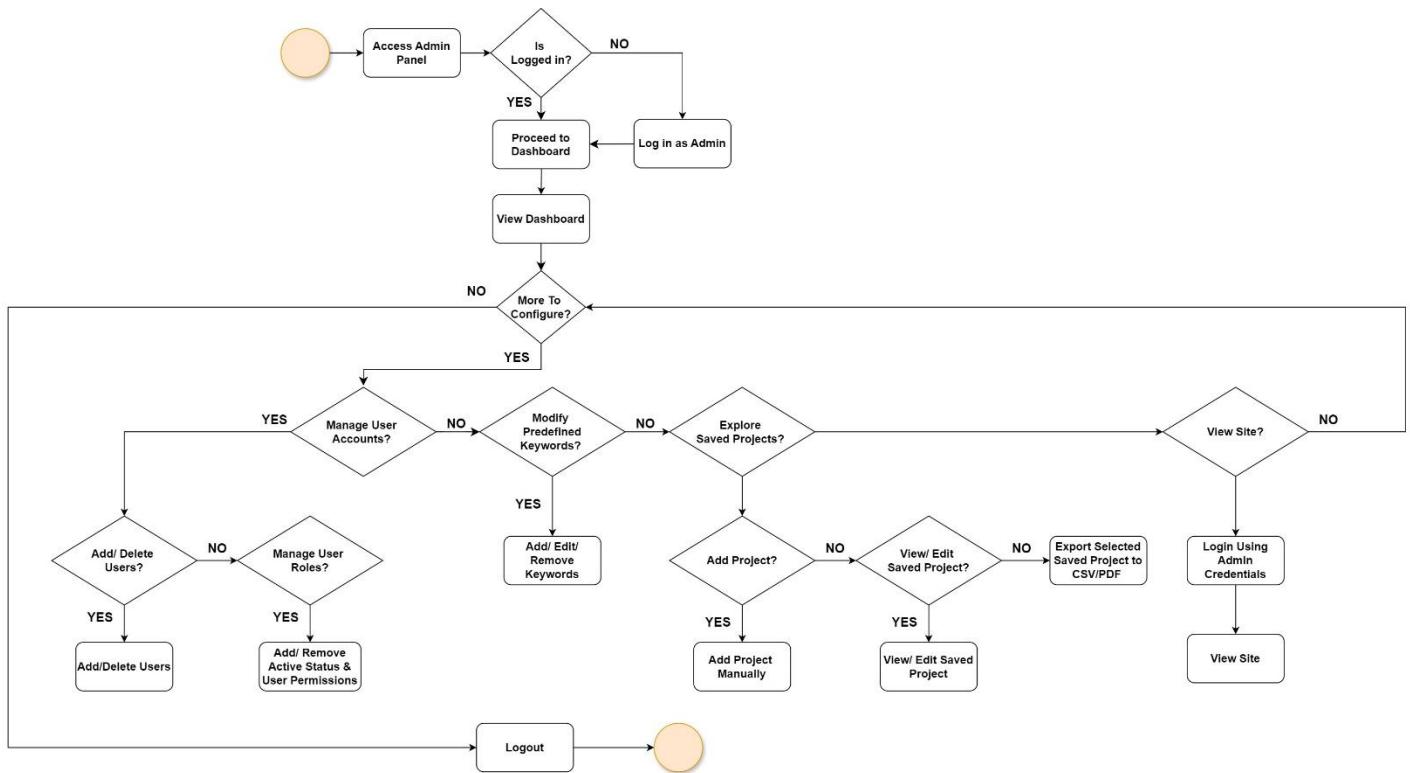


Figure 14 Admin Interaction Activity Diagram

3.6.3 Feature Design (Web Application)

Design for workings for individual core features of the system have been presented in the upcoming sections.

3.6.3.1 Login/Register/Logout

3.6.3.1.1 Use Case Diagram

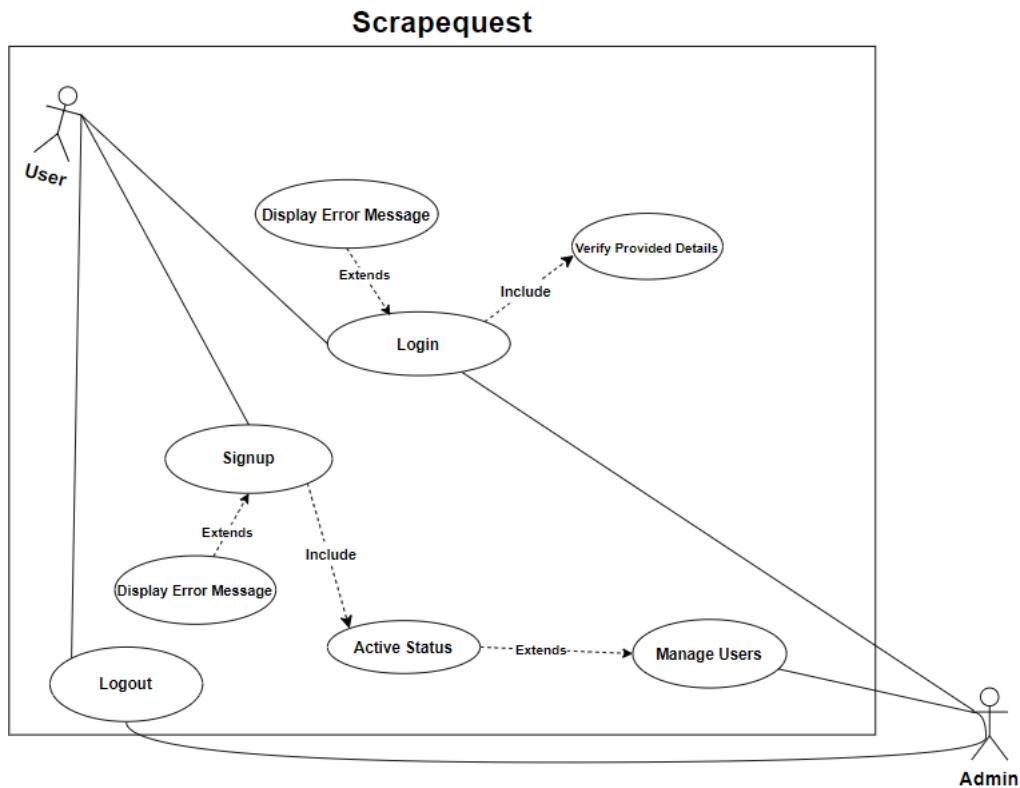


Figure 15 High Level Use-Case Diagram of Login/Register

3.6.3.1.2 Data Flow Diagram

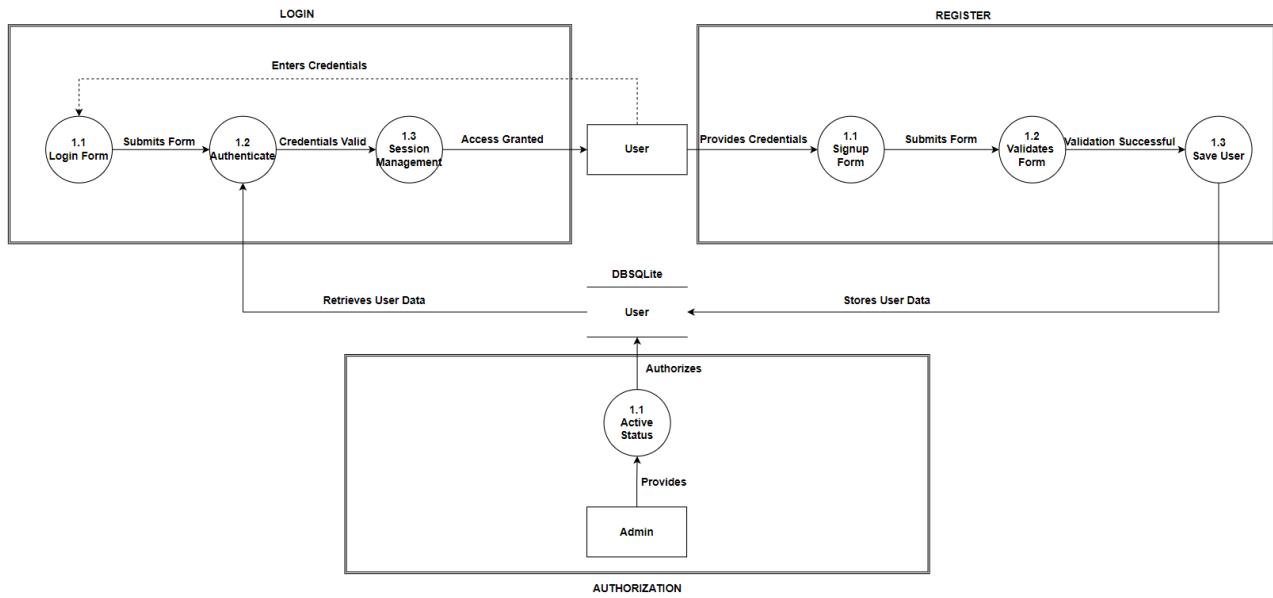


Figure 16 DFD Level 2 - (Login/Register)

3.6.3.1.3 Activity Diagram

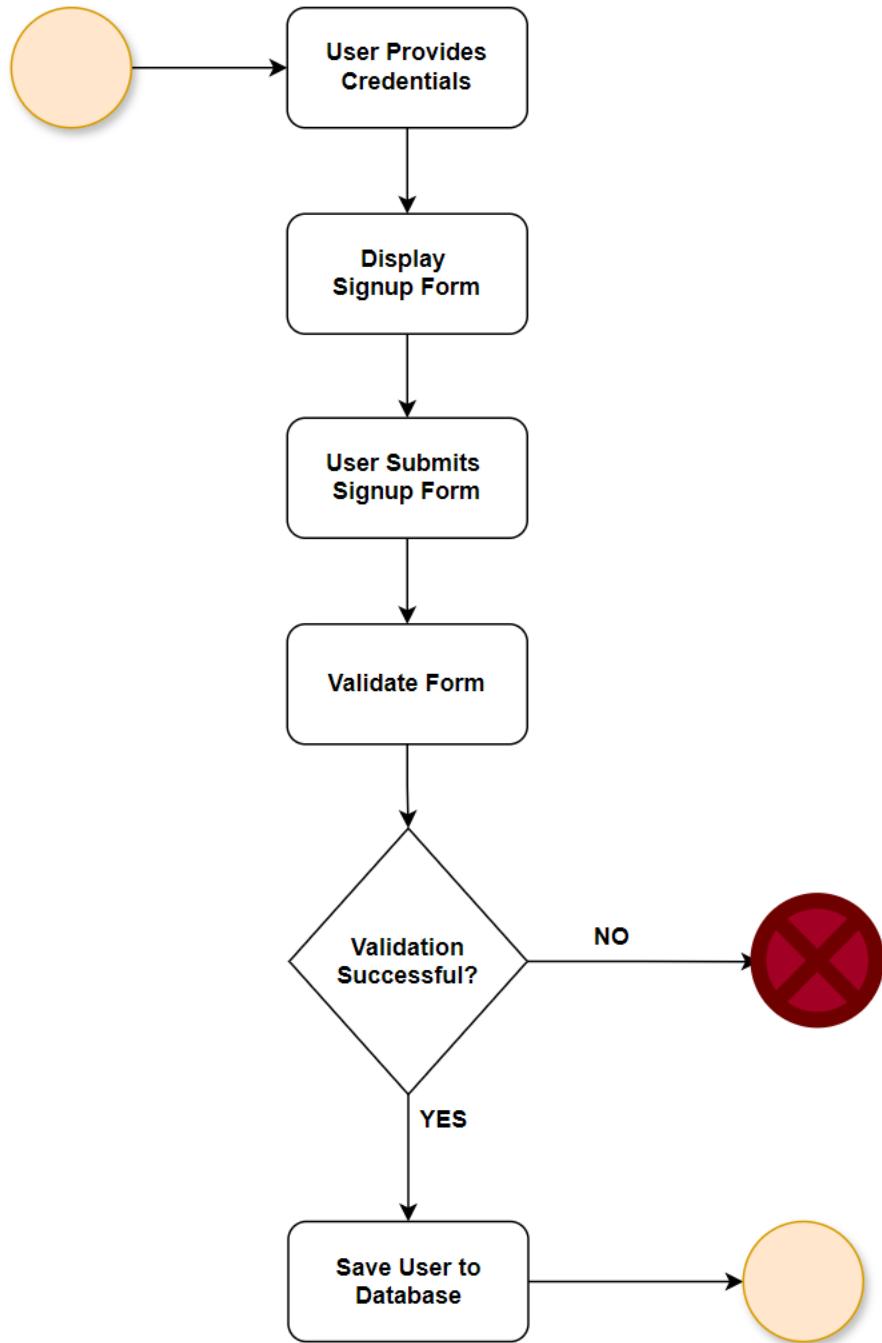


Figure 17 Activity Diagram for Registration

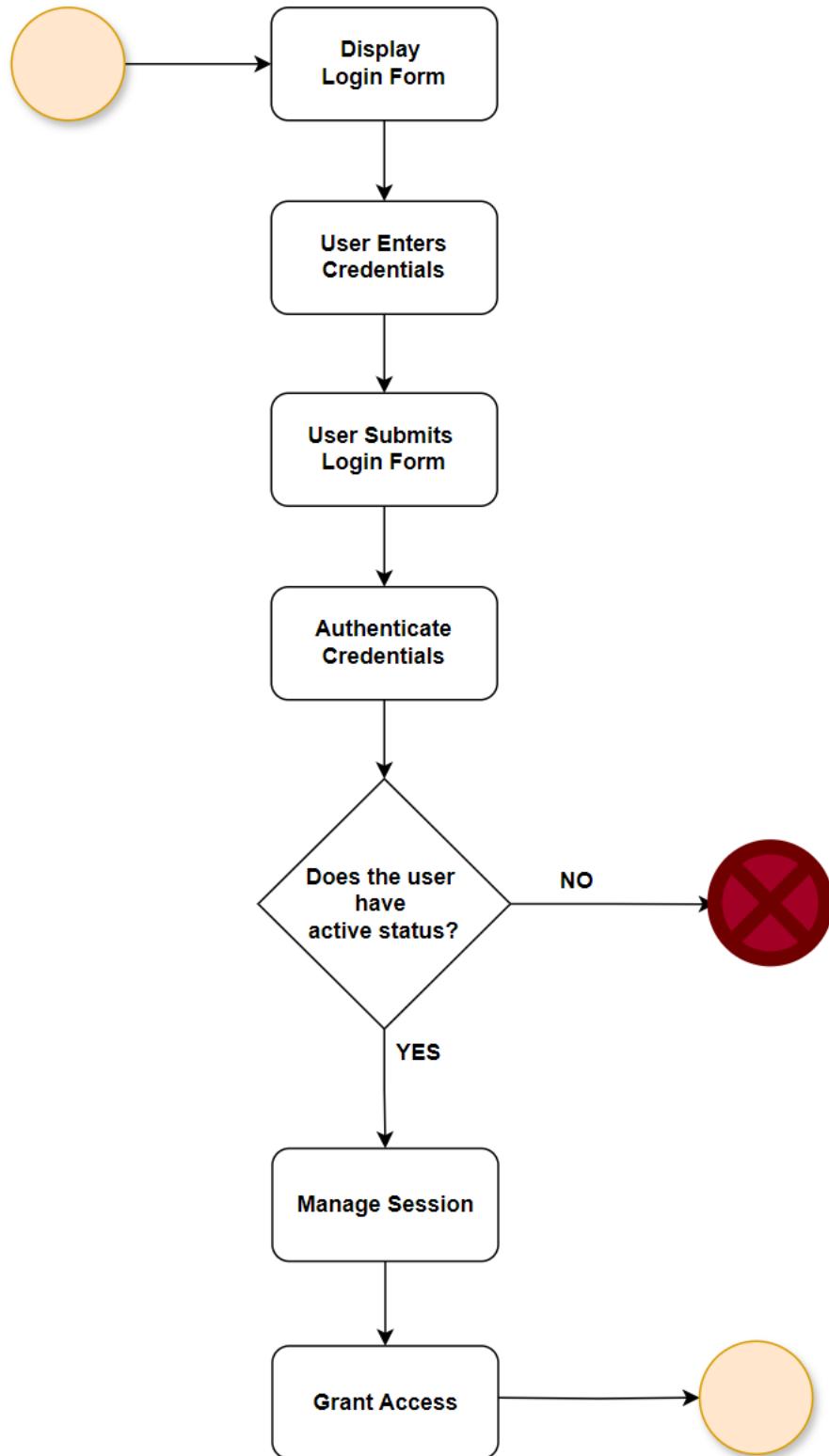


Figure 18 Activity Diagram for Login

3.6.3.2 Search Projects

3.6.3.2.1 Use Case Diagram

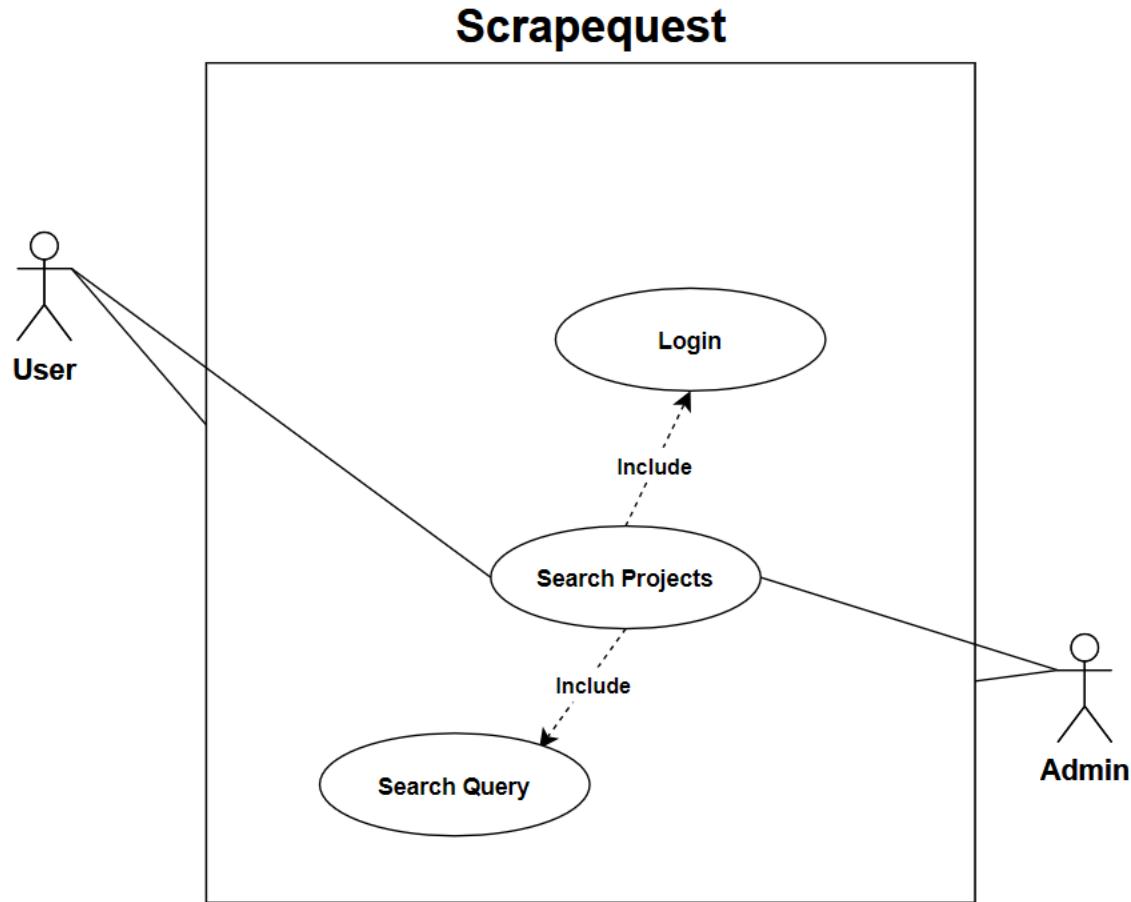


Figure 19 High Level Use Case Diagram of Searching Projects

3.6.3.2.2 Data Flow Diagram

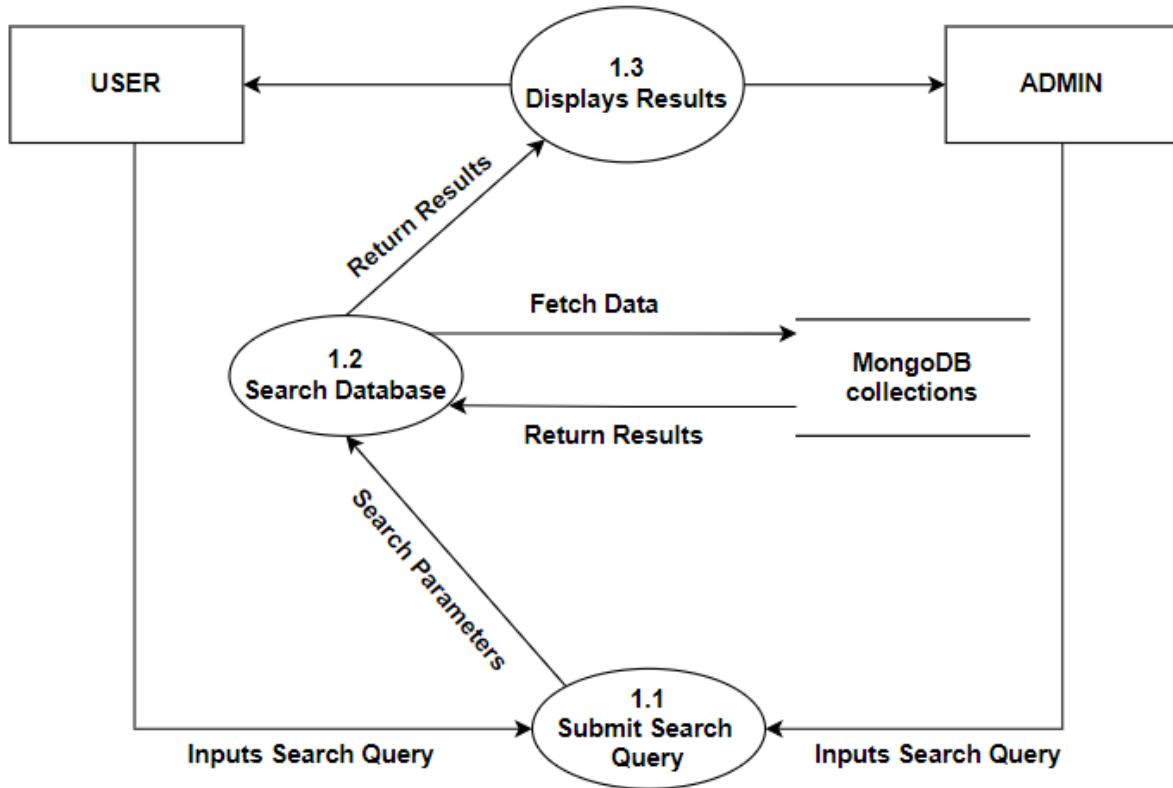


Figure 20 DFD Level 2 - (Searching Projects)

3.6.3.2.3 Activity Diagram

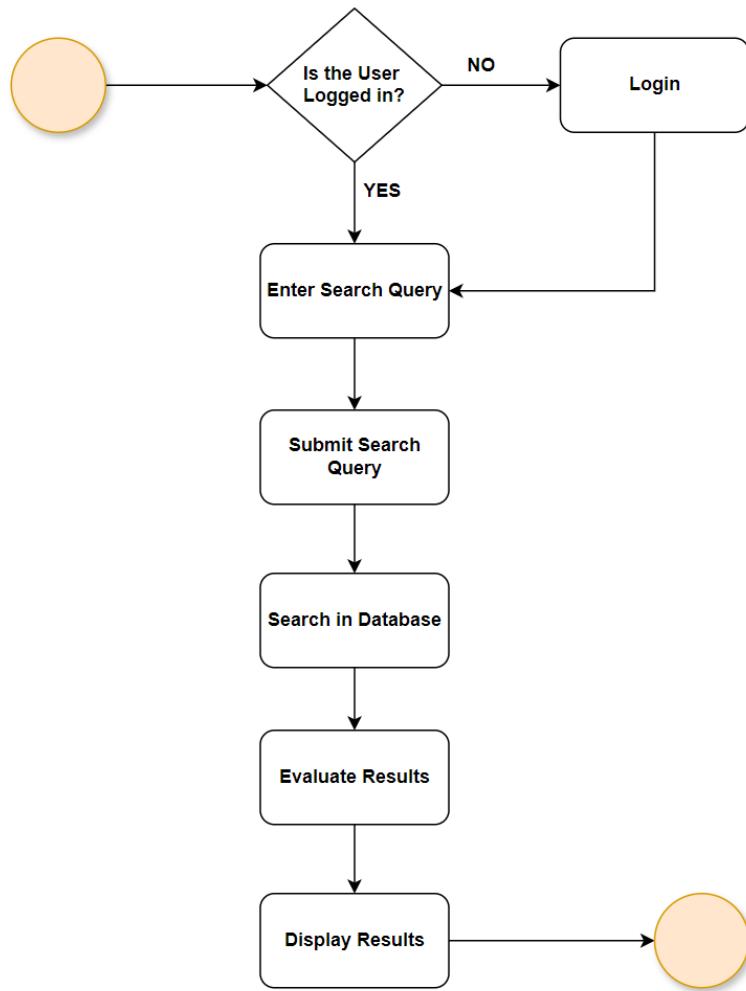


Figure 21 Activity Diagram for Searching Projects

3.6.3.3 Save Projects

3.6.3.3.1 Use Case Diagram

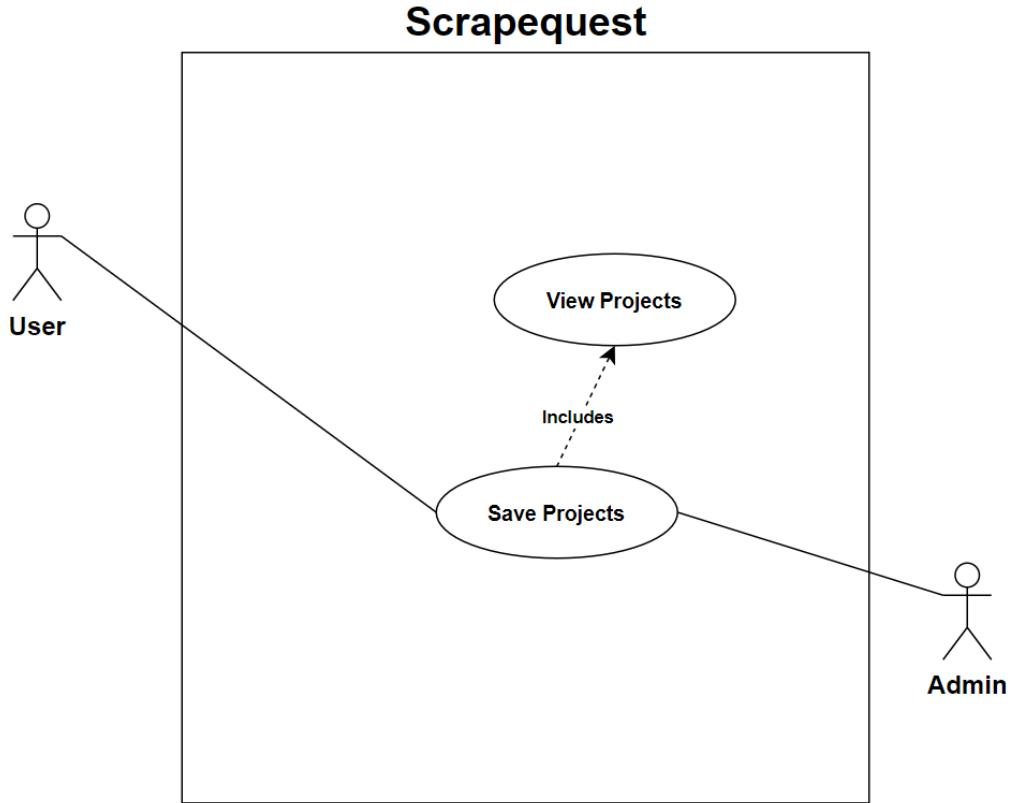


Figure 22 High Level Use Case Diagram of Saving Projects

3.6.3.3.2 Data Flow Diagram

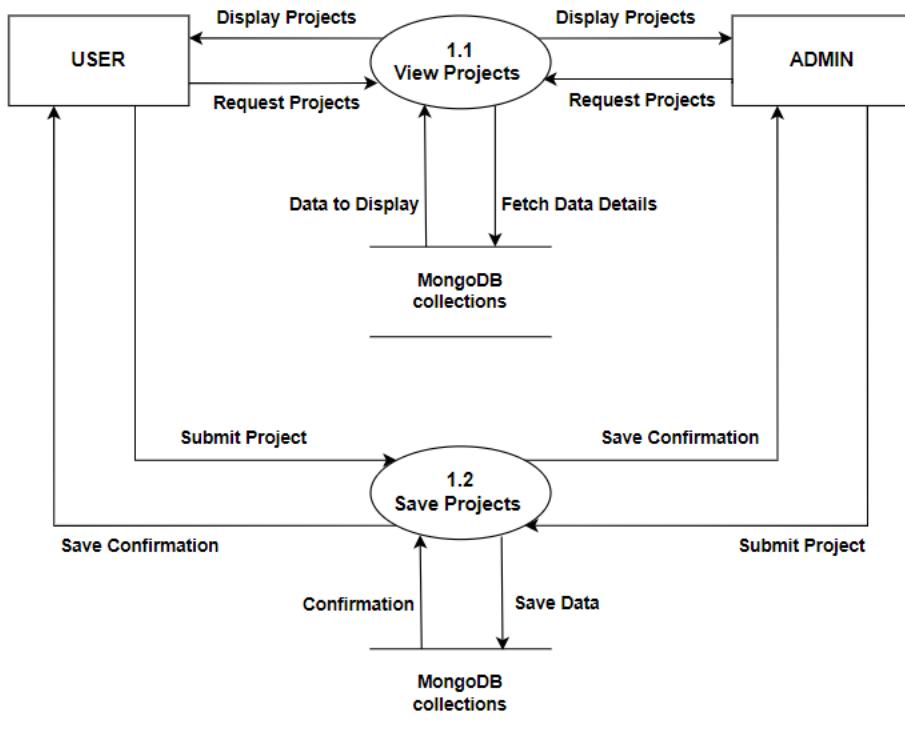


Figure 23 DFD Level 2 - (Saving Projects)

3.6.3.3.3 Activity Diagram

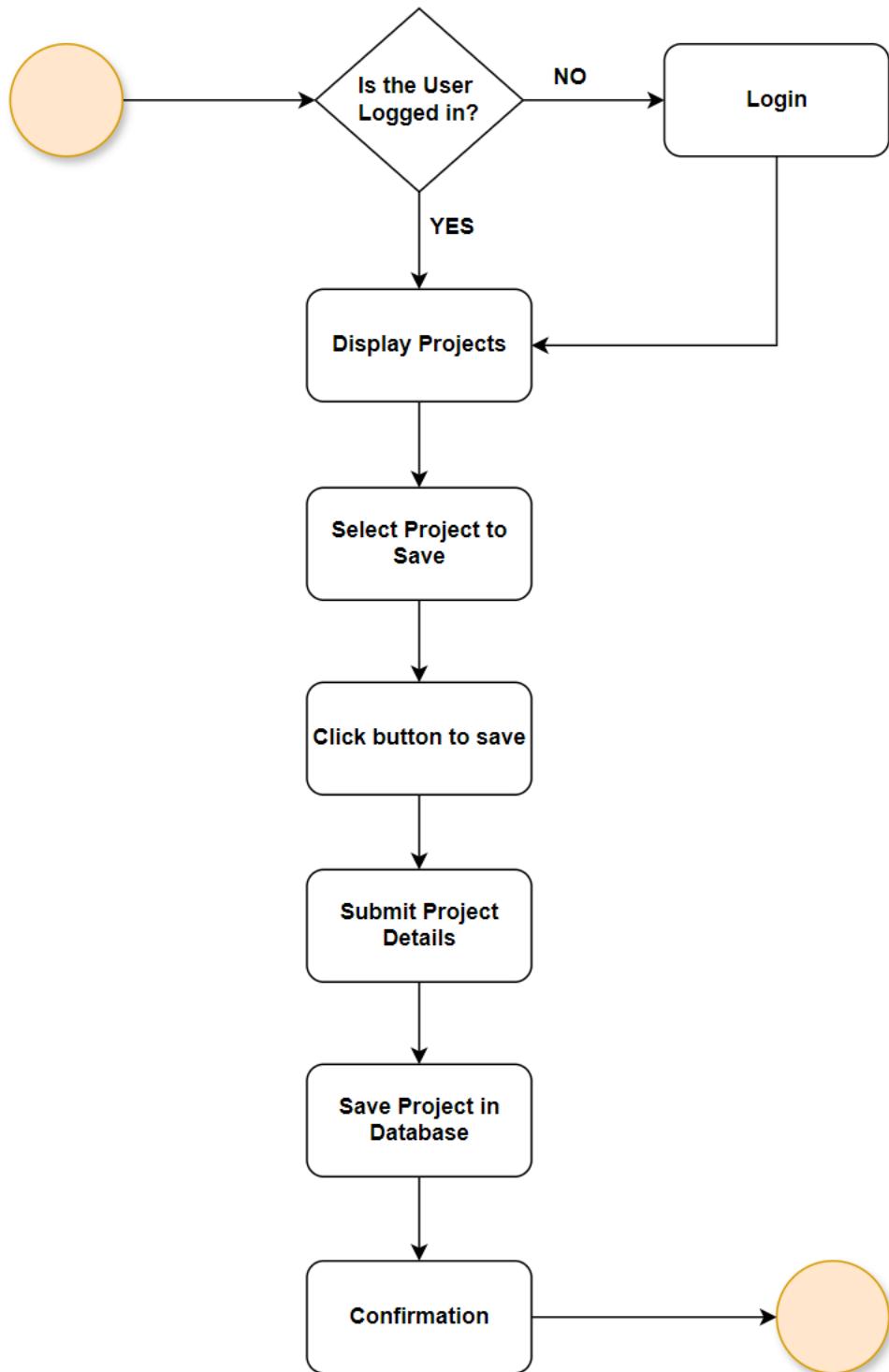


Figure 24 Activity Diagram for Saving Projects

3.6.3.4 View Saved Projects

3.6.3.4.1 Use Case Diagram

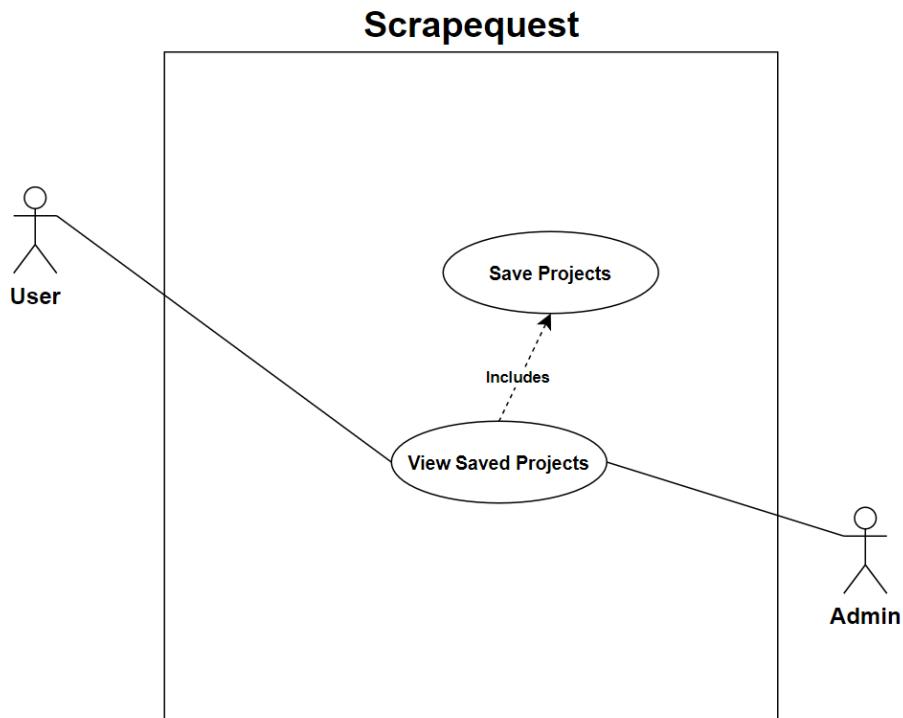


Figure 25 High Level Use Case Diagram of Viewing Saved Projects

3.6.3.4.2 Data Flow Diagram

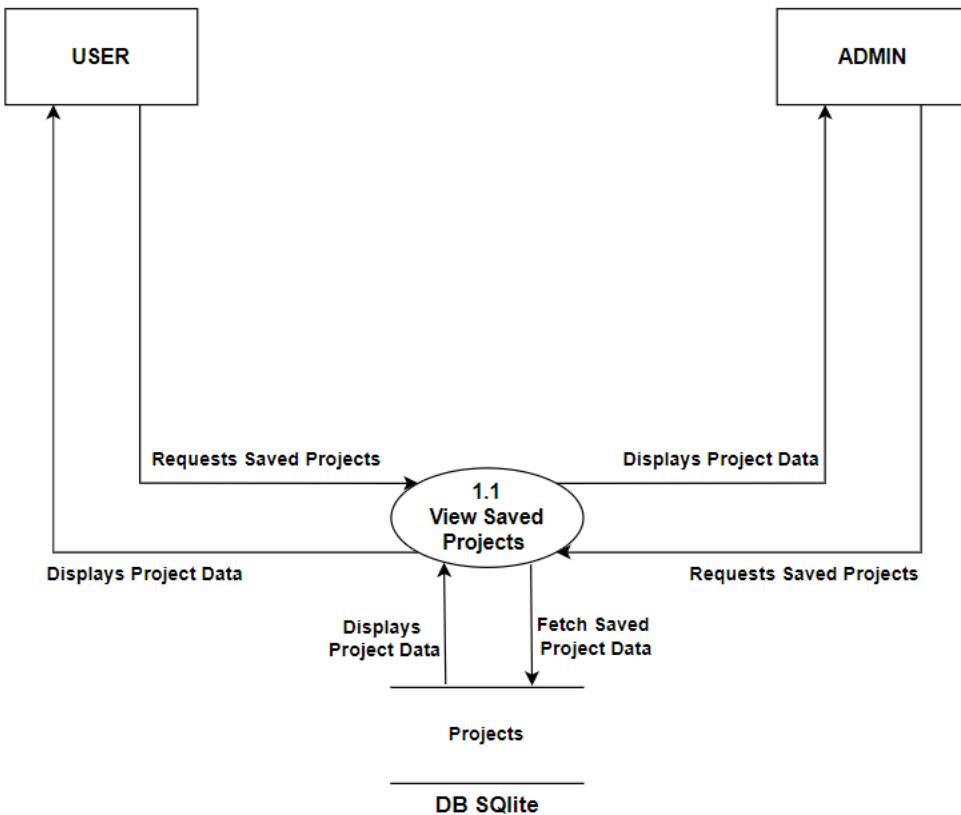


Figure 26 DFD Level 2 - (Viewing Saved Projects)

3.6.3.4.3 Activity Diagram

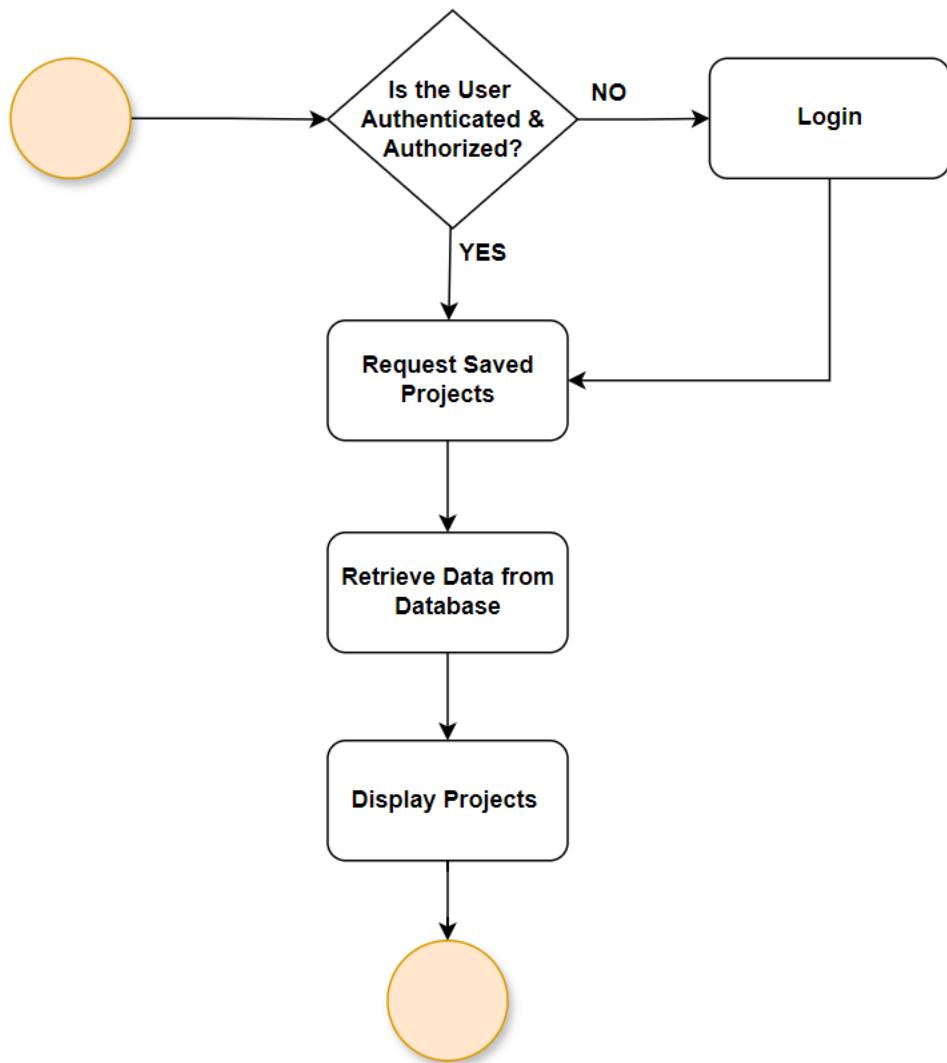


Figure 27 Activity Diagram for Viewing Saved Projects

3.6.3.5 View Project Details

3.6.3.5.1 Use Case Diagram

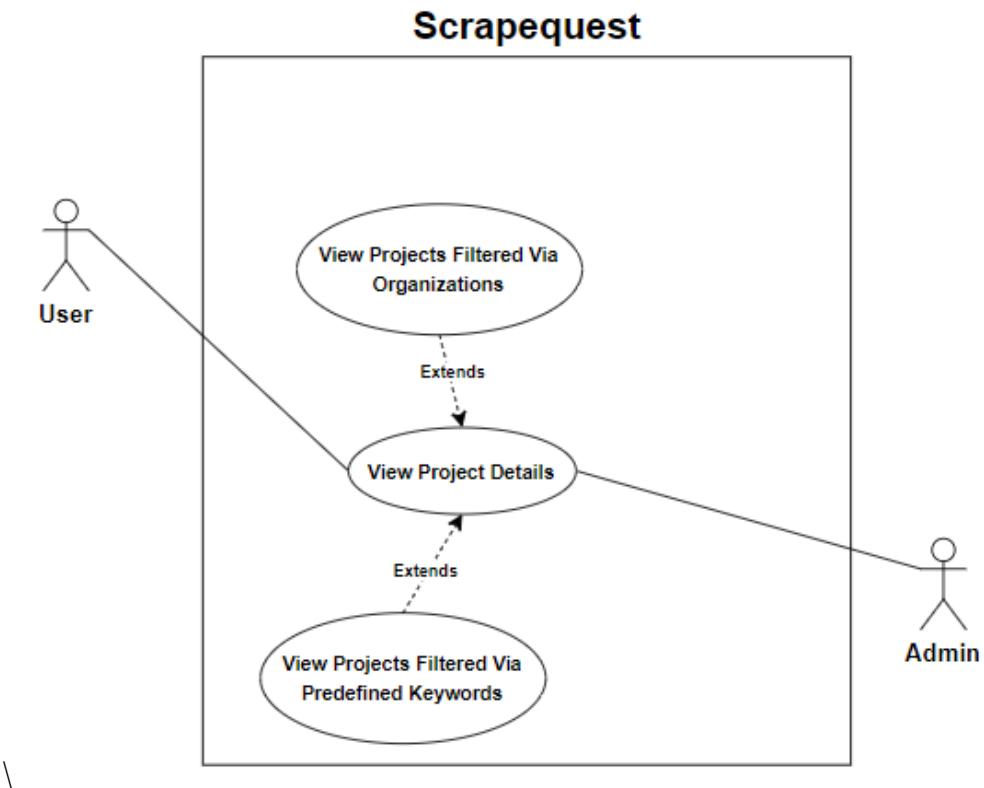


Figure 28 High Level Use Case Diagram of Viewing Project Detail

3.6.3.5.2 Data Flow Diagram

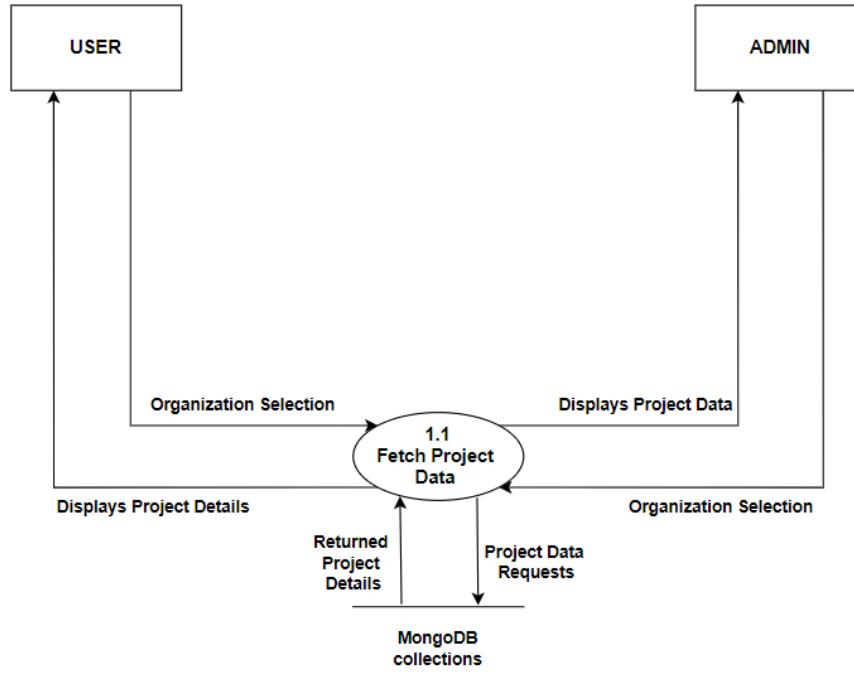


Figure 29 DFD Level 2 - (Viewing Project Details – Filtered via Organizations)

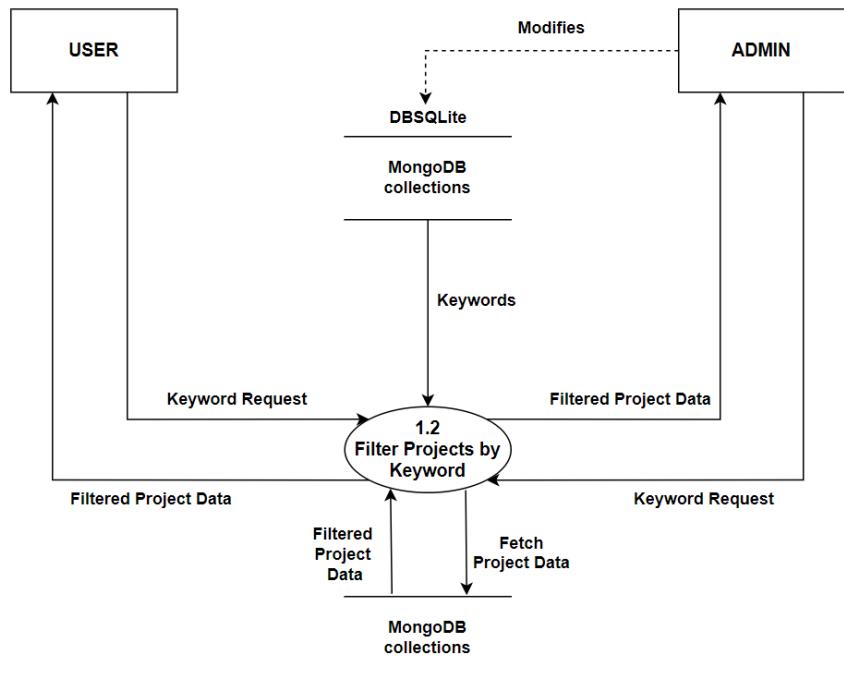


Figure 30 DFD Level 2 - (Viewing Project Details – Filtered via Predefined Keywords)

3.6.3.5.3 Activity Diagram

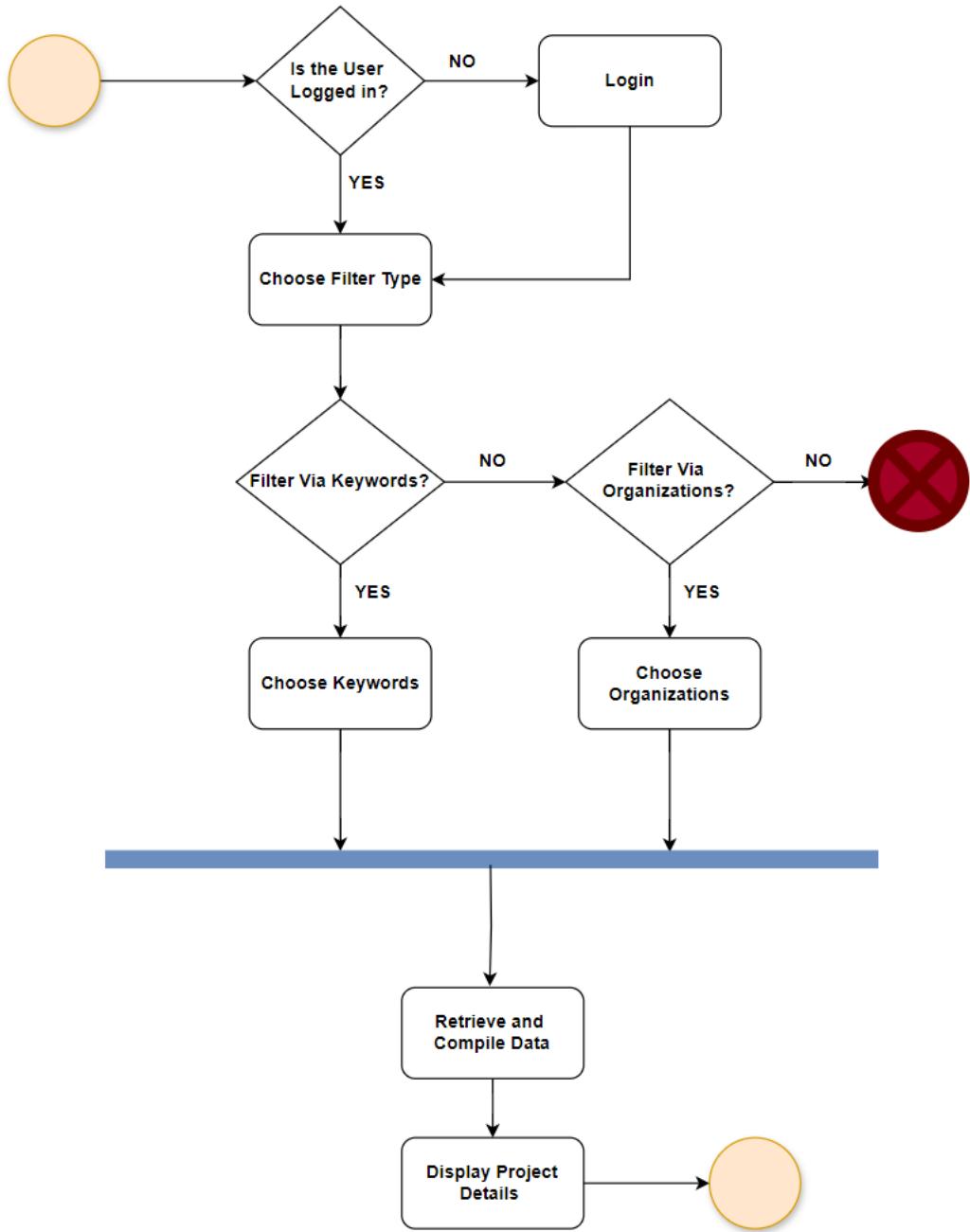


Figure 31 Activity Diagram for Viewing Project Details

3.6.3.6 View Project Details

3.6.3.6.1 Use Case Diagram

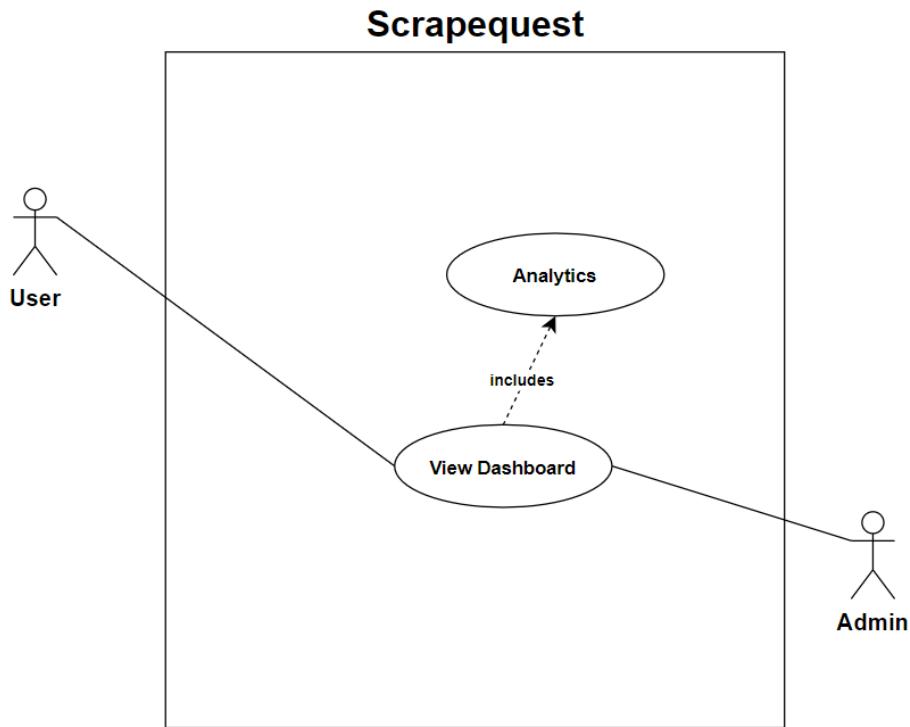


Figure 32 High Level Use Case Diagram of Viewing Analytics

3.6.3.6.2 Data Flow Diagram

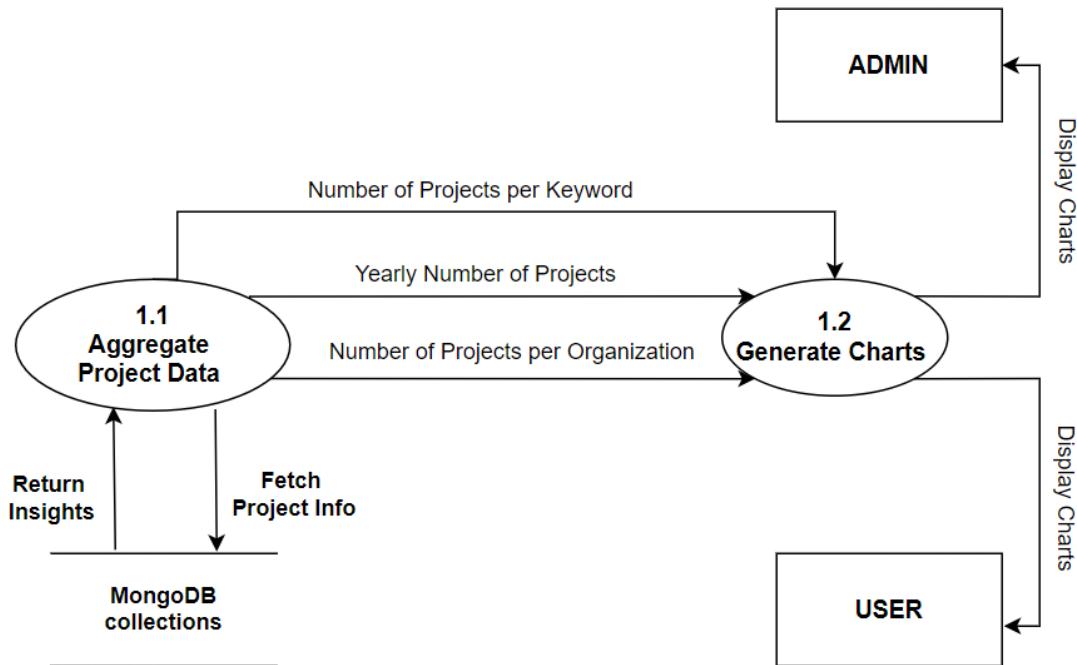


Figure 33 DFD Level 2 - (Viewing Analytics)

3.6.3.6.3 Activity Diagram

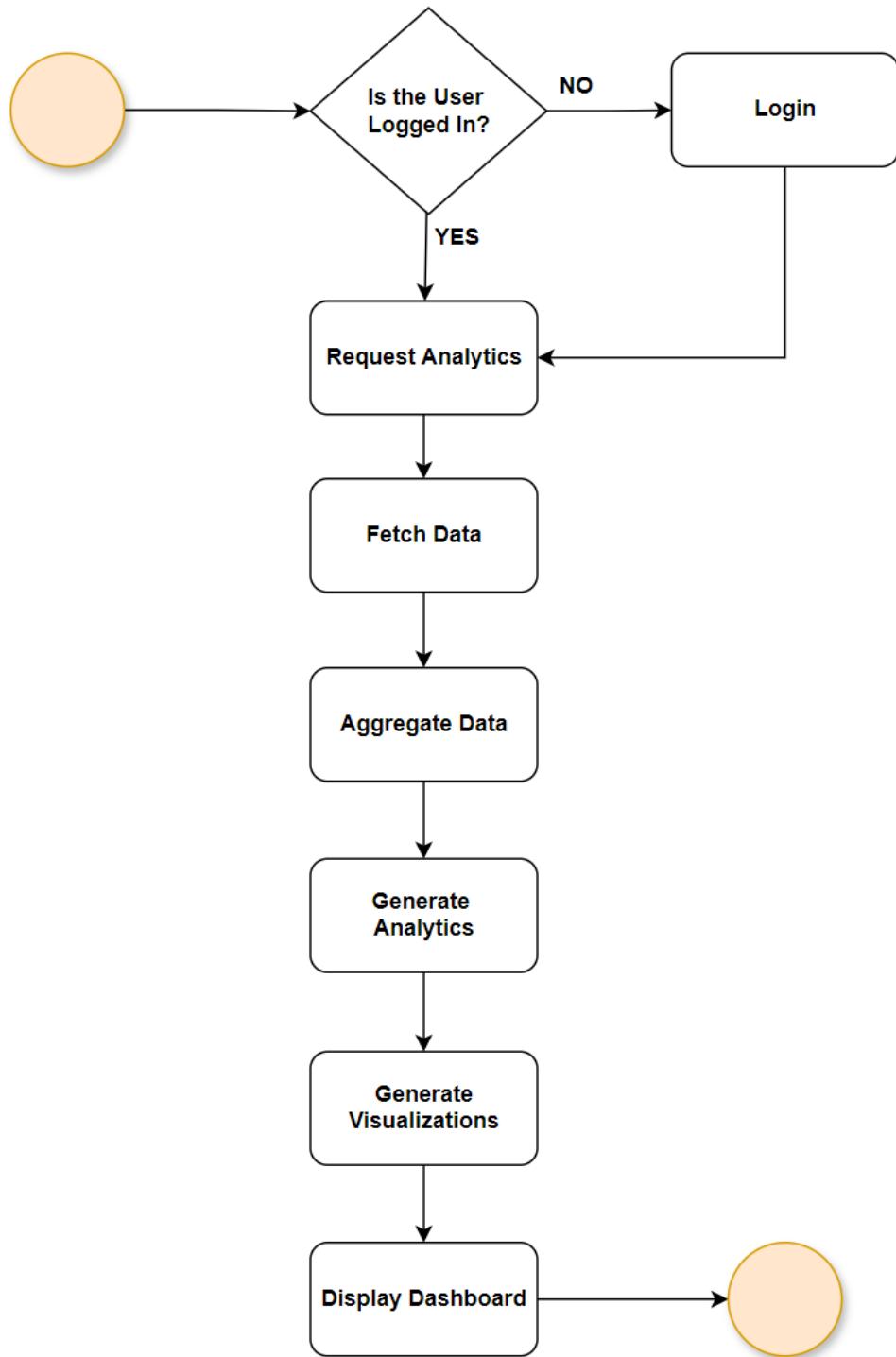


Figure 34 Activity Diagram for Viewing Analytics

3.6.3.7 Modify Predefined Keywords

3.6.3.7.1 Use Case Diagram

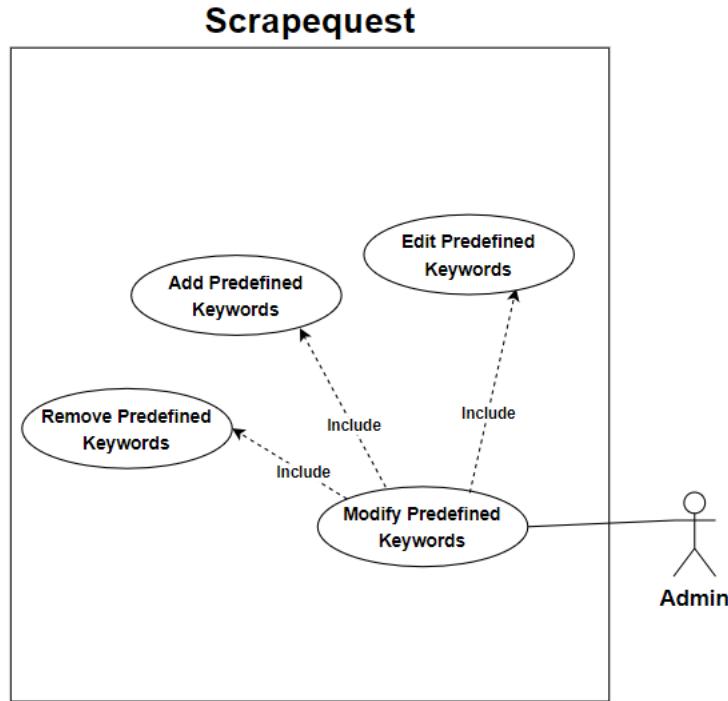


Figure 35 High Level Use Case of Modifying Predefined Keywords

3.6.3.7.2 Data Flow Diagram

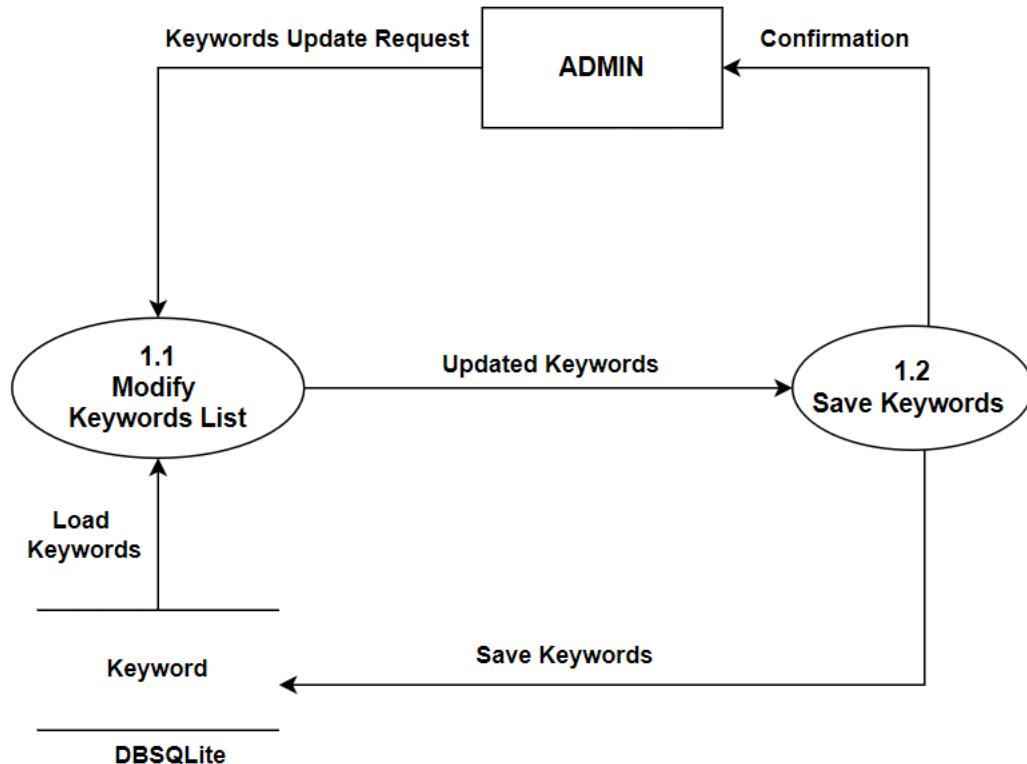


Figure 36 DFD Level 2 – (Modify Predefined Keywords)

3.6.3.7.3 Activity Diagram

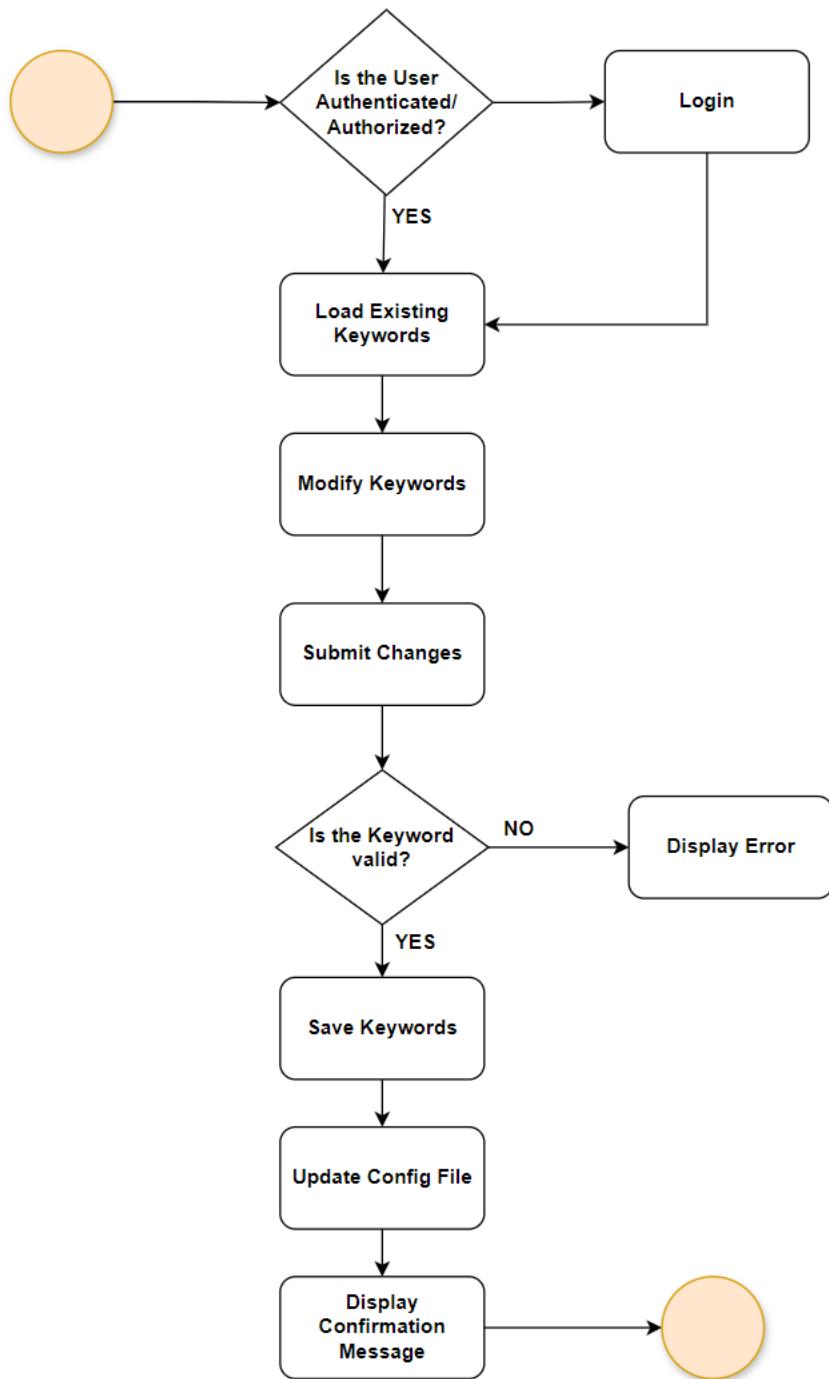


Figure 37 Activity Diagram for Modifying Predefined Keywords

3.6.3.8 Manage User

3.6.3.8.1 Use Case Diagram

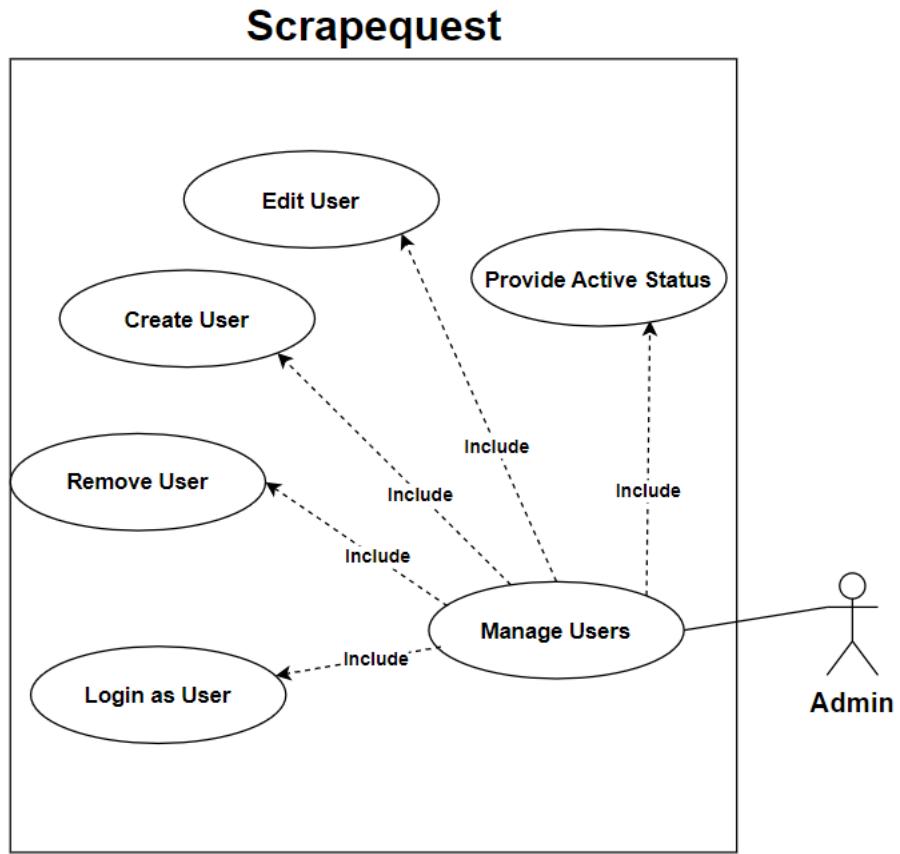


Figure 38 High Level Use Case Diagram of Managing Users

3.6.3.8.2 Data Flow Diagram

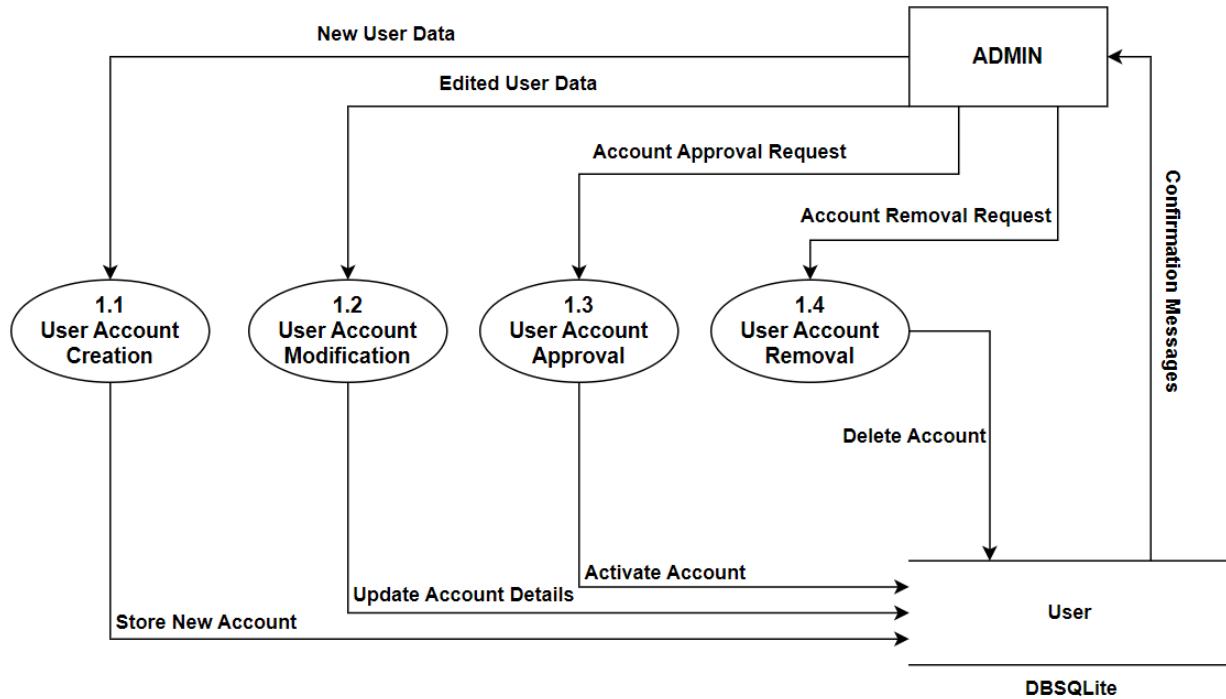


Figure 39 DFD Level 2 - (Manage User)

3.6.3.8.3 Activity Diagram

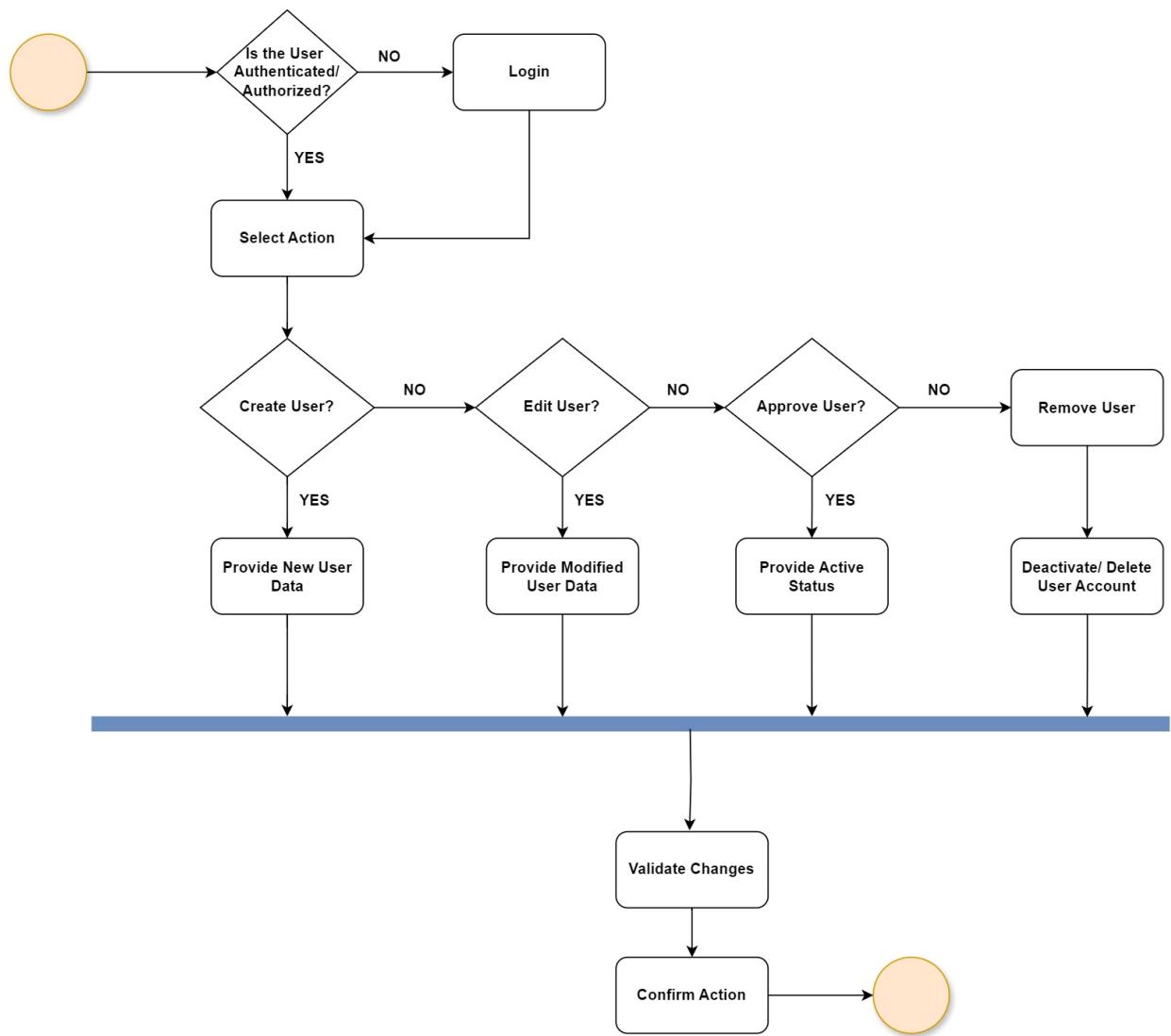


Figure 40 Activity Diagram for Managing Users

3.6.3.9 View and Export Saved Project Details

3.6.3.9.1 Use Case Diagram

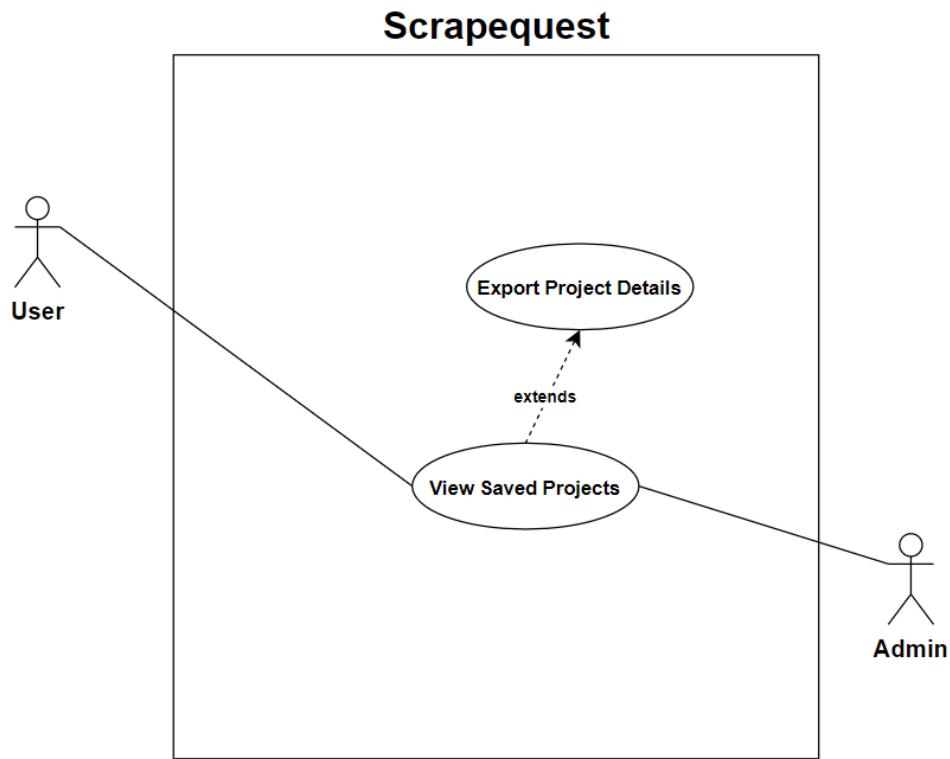


Figure 41 High Level Use Case Diagram of Viewing and Exporting Saved Project Details

3.6.3.9.2 Data Flow Diagram

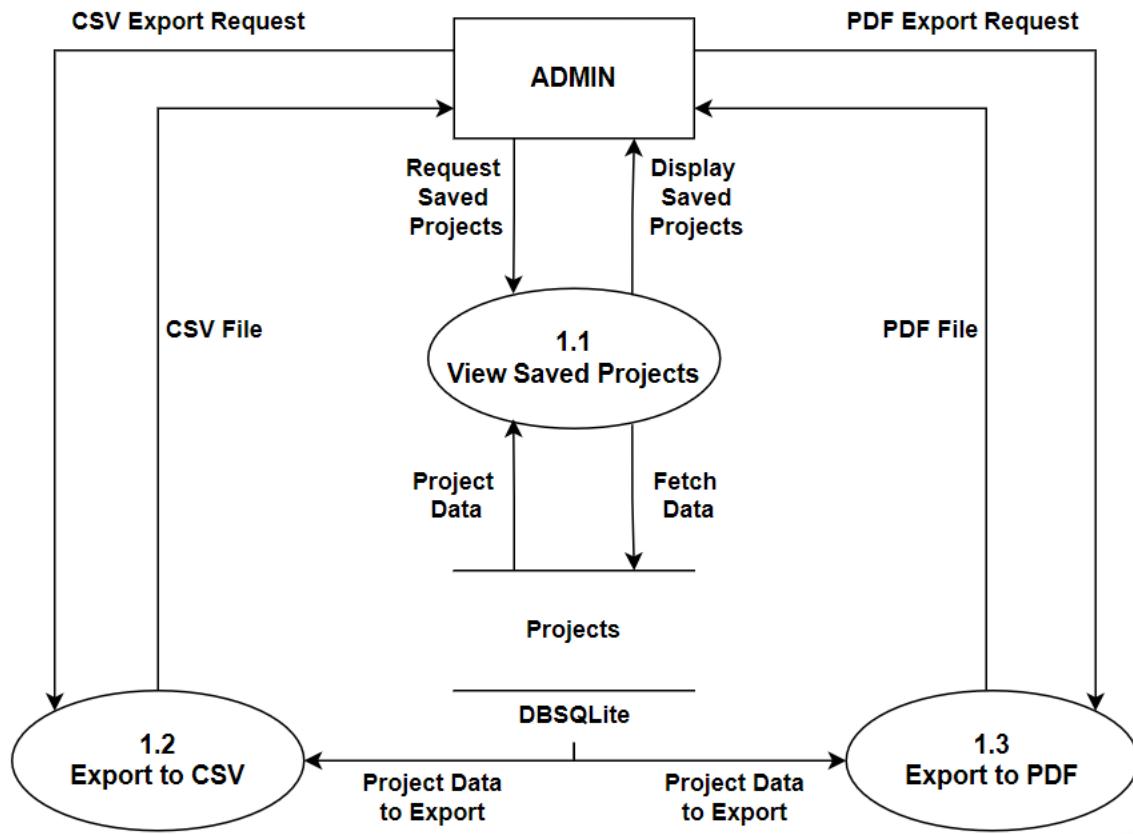


Figure 42 DFD Level 2 - (View and Export Saved Project Details)

3.6.3.9.3 Activity Diagram

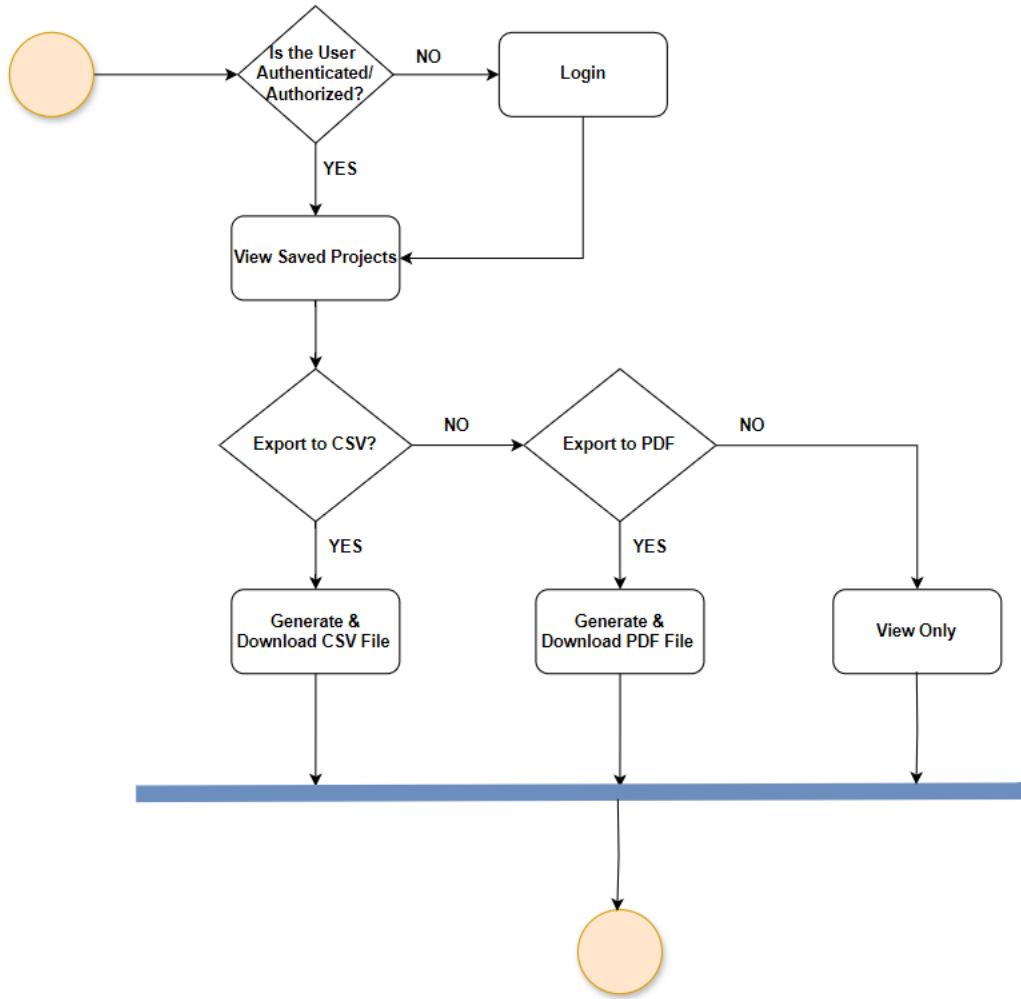


Figure 43 Activity Diagram for Viewing and Exporting Saved Project Details

3.7 Implementation

This section demonstrates the implementation of RUP methodology by showcasing tasks completed in each sprint.

[Note: Only screenshots of development core features are included in the section. For Detailed Codes of each UI components, refer to this [section](#).]

3.7.1 Inception

This phase has been completed in three sprints which is as follows:

<input type="checkbox"/>	▼ FYP Sprint 1 1 Oct – 8 Oct (4 issues)		
To finalize the project idea and understand client expectations.			
<input checked="" type="checkbox"/>	FYP-5 Brainstorming sessions to think about the project idea.	PROJECT INCEPTION	DONE ✓
<input checked="" type="checkbox"/>	FYP-6 Consult with supervisor and finalize the project idea	PROJECT INCEPTION	DONE ✓
<input checked="" type="checkbox"/>	FYP-7 Research and identify the client	PROJECT INCEPTION	DONE ✓
<input checked="" type="checkbox"/>	FYP-8 Preparing and conducting a meeting with the client to present the idea and gather initial feedback.	PROJECT INCEPTION	DONE ✓

Figure 44 Sprint 1 - Backlogs (Project Inception)

<input type="checkbox"/>	▼ FYP Sprint 2 8 Oct – 15 Oct (3 issues)		
To define detailed project requirements and constraints.			
<input checked="" type="checkbox"/>	FYP-9 Gathering all necessary information about the project scope, technical and non-technical requirements.	PROJECT INCEPTION	DONE ✓
<input checked="" type="checkbox"/>	FYP-10 Conducting analysis to understand the feasibility and implications of the requirements.	PROJECT INCEPTION	DONE ✓
<input checked="" type="checkbox"/>	FYP-20 Drafting the SRS document and develop system use cases	PROJECT INCEPTION	DONE ✓

Figure 45 Sprint 2 - Backlogs (Project Inception)

<input type="checkbox"/>	▼ FYP Sprint 3 15 Oct – 25 Oct (5 issues)		
To evaluate and determine the resources needed for the project and create an initial, flexible project plan that outlines the project timeline, milestones, and resource allocation.			
<input checked="" type="checkbox"/>	FYP-11 Identifying technological, and informational resources needed to carry out the project.	PROJECT INCEPTION	DONE ✓
<input checked="" type="checkbox"/>	FYP-12 Assessing resource availability and identifying gaps.	PROJECT INCEPTION	DONE ✓
<input checked="" type="checkbox"/>	FYP-13 Drafting a project timeline with major milestones.	PROJECT INCEPTION	DONE ✓
<input checked="" type="checkbox"/>	FYP-14 Allocating resources to different project components.	PROJECT INCEPTION	DONE ✓
<input checked="" type="checkbox"/>	FYP-15 Establishing communication and drafting risk management plans.	PROJECT INCEPTION	DONE ✓

Figure 46 Sprint 3 - Backlogs (Project Inception)

3.7.2 Project Elaboration

This phase has been completed in four sprints which is as follows:

<input type="checkbox"/> ▾ FYP Sprint 4 26 Oct – 2 Nov (3 issues)	To reassess and confirm the resources necessary for the project based on the refined project scope and objectives.	
<input checked="" type="checkbox"/> FYP-16 Reviewing the initial resource evaluation against the detailed project requirements.		PROJECT ELABORATION
<input checked="" type="checkbox"/> FYP-17 Identifying additional skills, technology, or information resources that are needed.		PROJECT ELABORATION
<input checked="" type="checkbox"/> FYP-18 Adjusting resource allocation plans to fill any gaps identified.		PROJECT ELABORATION

Figure 47 Sprint 4 - Backlogs (Project Elaboration)

<input type="checkbox"/> ▾ FYP Sprint 5 3 Nov – 10 Nov (3 issues)	To create a comprehensive project plan that includes detailed timelines, resource allocations, and milestones.	
<input checked="" type="checkbox"/> FYP-19 Breaking down the project into smaller tasks.		PROJECT ELABORATION
<input checked="" type="checkbox"/> FYP-20 Establishing a detailed timeline with specific milestones and deadlines.		PROJECT ELABORATION
<input checked="" type="checkbox"/> FYP-21 Planning sprints and allocating resources for each sprint based on task complexity and dependencies.		PROJECT ELABORATION

Figure 48 Sprint 5 - Backlogs (Project Elaboration)

<input type="checkbox"/> ▾ FYP Sprint 6 11 Nov – 18 Nov (3 issues)	To identify potential risks to the project and outline mitigation strategies.	
<input checked="" type="checkbox"/> FYP-22 Conducting a risk analysis to identify potential project risks, including technical, financial, and organizational risks.		PROJECT ELABORATION
<input checked="" type="checkbox"/> FYP-23 Developing mitigation strategies for each identified risk.		PROJECT ELABORATION
<input checked="" type="checkbox"/> FYP-24 Prioritizing risks based on their potential impact on the project and the likelihood of occurrence.		PROJECT ELABORATION

Figure 49 Sprint 6 - Backlogs (Project Elaboration)

<input type="checkbox"/> ▾ FYP Sprint 7 18 Nov – 29 Nov (3 issues)	To compile and document a project proposal that includes the project plan, resource allocation, risk management plan, and other essential details.	
<input checked="" type="checkbox"/> FYP-25 Drafting a comprehensive project proposal document.		PROJECT ELABORATION
<input checked="" type="checkbox"/> FYP-26 Incorporating feedback from previous sprints to refine the proposal.		PROJECT ELABORATION

Figure 50 Sprint 7 - Backlogs (Project Elaboration)

3.7.3 Project Construction

This phase has been completed in fourteen sprints which is as follows:

<input type="checkbox"/>	▼ FYP Sprint 8 29 Nov – 6 Dec (5 issues)	To gather all necessary project information from UNDP and UNFAO that will feed into the system.	
<input checked="" type="checkbox"/>	FYP-29 Analyzing the website of UNDP to initiate scraping		PROJECT CONSTRUCTI...
<input checked="" type="checkbox"/>	FYP-30 Implementing scraping scripts to collect data from UNDP		PROJECT CONSTRUCTI...
<input type="checkbox"/>	<input checked="" type="checkbox"/> FYP-31 <u>Analyzing the website of UNFAO to initiate scraping</u> 		PROJECT CONSTRUCTI...
<input checked="" type="checkbox"/>	FYP-35 Implementing scraping scripts to collect data from UNFAO		PROJECT CONSTRUCTI...
<input checked="" type="checkbox"/>	FYP-36 Implementing codes for cleaning collected data		PROJECT CONSTRUCTI...

Figure 51 Sprint 8 - Backlogs (Project Construction)

<input type="checkbox"/>	▼ FYP Sprint 9 6 Dec – 13 Dec (5 issues)	To gather all necessary information from websites of UKAID and SDC that will feed into the system.	
<input checked="" type="checkbox"/>	FYP-32 Analyzing the website of SDC to initiate scraping		PROJECT CONSTRUCTI...
<input checked="" type="checkbox"/>	FYP-33 Implementing scraping scripts to collect data from SDC		PROJECT CONSTRUCTI...
<input checked="" type="checkbox"/>	FYP-34 Analyzing the website of UKAID to initiate scraping		PROJECT CONSTRUCTI...
<input checked="" type="checkbox"/>	FYP-37 Implementing scraping scripts to collect data from UKAID		PROJECT CONSTRUCTI...
<input checked="" type="checkbox"/>	FYP-38 Implementing codes for cleaning collected data		PROJECT CONSTRUCTI...

Figure 52 Sprint 9 - Backlogs (Project Construction)

<input type="checkbox"/>	▼ FYP Sprint 10 13 Dec – 20 Dec (5 issues)	To gather all necessary project information from websites of ADB and WB that will feed into the system.	
<input checked="" type="checkbox"/>	FYP-39 Analyzing the website of ADB to initiate scraping		PROJECT CONSTRUCTI...
<input checked="" type="checkbox"/>	FYP-40 Implementing scraping scripts to collect data from ADB		PROJECT CONSTRUCTI...
<input checked="" type="checkbox"/>	FYP-41 Analyzing the website of WB to initiate scraping		PROJECT CONSTRUCTI...
<input checked="" type="checkbox"/>	FYP-42 Implementing scraping scripts to collect data from WB		PROJECT CONSTRUCTI...
<input type="checkbox"/>	<input checked="" type="checkbox"/> FYP-43 <u>Implementing codes for cleaning collected data</u> 		PROJECT CONSTRUCTI...

Figure 53 Sprint 10 - Backlogs (Project Construction)

<input type="checkbox"/>	▼ FYP Sprint 11 21 Dec – 29 Dec (6 issues)	To gather all necessary project information from JICA and USAID that will feed into the system, and organizing folder structure and re-testing created scripts
<input checked="" type="checkbox"/>	FYP-44 Analyzing the website of JICA to initiate scraping	PROJECT CONSTRUCT...
<input checked="" type="checkbox"/>	FYP-45 Implementing scraping scripts to collect data from JICA	PROJECT CONSTRUCT...
<input checked="" type="checkbox"/>	FYP-46 Analyzing the website of USAID to initiate scraping	PROJECT CONSTRUCT...
<input checked="" type="checkbox"/>	FYP-47 Implementing scraping scripts to collect data from USAID	PROJECT CONSTRUCT...
<input checked="" type="checkbox"/>	FYP-48 Implementing codes for cleaning collected data	PROJECT CONSTRUCT...
<input checked="" type="checkbox"/>	FYP-49 Organizing folder structure and testing all of the created scripts.	PROJECT CONSTRUCT...

Figure 54 Sprint 11 - Backlogs (Project Construction)

3.7.3.1 Important Screenshots (Sprint 8-11):

```

while True:
    try:
        ADBS = []
        page = driver.page_source
        page_soup = soup(page, 'html.parser')

        containers = page_soup.findAll('div', {'class': 'item linked'})

        for container in containers:
            ADB = {}

            title = container.find('div', {'class' : 'item-title'})
            project_titles = title.find('a')
            project_link = title.a['href']
            for project_title in project_titles:
                project_title = (project_titles.text.strip())

            ADB['Name of the Project'] = project_title
            ADB['Link to the project'] = project_link

            project_status = container.find('div', {'class': 'item-meta'})
            stats = project_status.find('div')
            stat = (stats.text.strip())

            ADB['Project Status'] = stat

            project_summary = container.find('div', {'class':'item-summary'})
            summary = (project_summary.text.strip())

            ADB['Project Description'] = summary

            file = open('ADB.csv', 'a', newline= '', encoding='utf-8')
            writer = csv.writer(file)
            headers = ([project_title, project_link ,stat, summary])
            writer.writerow(headers)
            writer.writerow(ADB)
            file.close()

            pass

            time.sleep(5)
            page_num += 1
            new_url = f'https://www.adb.org/projects/tenders/country/nepal?page={page_num}'
            driver.get(new_url)
            driver.set_page_load_timeout(60)

            driver.quit()
    except:
        break

```

Figure 55 Sample Scraping Logic for ADB

General Overview of Scraping Process

- Each script starts by setting up the required libraries and sometimes parameters like the page number to start from.
- Selenium is used to open and navigate web pages. It can interact with the webpage, click buttons, and send keys when necessary.
- The scripts use BeautifulSoup to parse the HTML content of the pages retrieved by Selenium.
- They look for specific HTML elements, such as div, a, h3, etc., which contain the data they are interested in.
- The scripts extract text from these elements, clean it up, and sometimes manipulate URLs to ensure they are absolute and not relative.
- The extracted data is structured into dictionaries, with keys representing the kind of data, which is then written to CSV files, with each dictionary representing a row in the file.
- To scrape multiple pages, a few scripts also increment the page number and modify the URL to navigate to the next page.
- They also include try-except blocks to handle errors that may occur during scraping, such as issues with loading a page or finding an expected HTML element.

[Note: Click [here](#) to view all the scraping scripts]

```

import pandas as pd

def load_data(file_path):
    """ Load data from a CSV file """
    return pd.read_csv(file_path)

def clean_data(df):
    """ Clean the dataframe """
    # Remove duplicate rows
    df = df.drop_duplicates()
    # Drop rows where all elements are missing
    df = df.dropna(how='all')
    # Replace missing values with a placeholder or an interpolation
    df = df.fillna(method='ffill').fillna('Unknown')
    return df

def test_cleaning(original_df, cleaned_df):
    """ Perform tests to ensure data cleaning was successful """
    assert len(cleaned_df) <= len(original_df), "Cleaned data should not have more rows than original"
    assert cleaned_df.isna().sum().max() == 0, "No NaN values should remain after cleaning"
    print( path + " Tests passed successfully!")

#usage
directory = '../ScrapeBots/ProjectData/'
file_names = ['ADB.csv', 'JICA.csv', 'SDC.csv', 'UKAID.csv', 'UNDP.csv']
file_paths = [directory + file_name for file_name in file_names]
all_dataframes = []

for path in file_paths:
    df = load_data(path)
    cleaned_df = clean_data(df)
    test_cleaning(df, cleaned_df)
    all_dataframes.append(cleaned_df)

```

Figure 56 Data Cleaning Script

The code imports pandas library for data manipulation and analysis. It defines functions like load_data, clean_data, and test_cleaning to read and analyze CSV files. It also initializes an empty list of all_dataframes to store cleaned DataFrames, iterates over file paths, loads each file, cleans it, tests the cleaning process, and adds the cleaned DataFrame to the list.

<input type="checkbox"/>	▼ FYP Sprint 12 30 Dec – 10 Jan (5 issues)	
To set up a database to store and manage the collected data, and migrate the data.		
<input checked="" type="checkbox"/>	FYP-50 Creating a Mongo Atlas Account and setting it up	PROJECT CONSTRUCTI...
<input checked="" type="checkbox"/>	FYP-51 Creating a 'free' cluster and the database	PROJECT CONSTRUCTI...
<input checked="" type="checkbox"/>	FYP-52 Testing connection to the cluster and database using pymongo	PROJECT CONSTRUCTI...
<input checked="" type="checkbox"/>	FYP-53 Implementing scripts to migrate information of projects from .csv files to the database in Mongo Atlas Cluster	PROJECT CONSTRUCTI...
<input checked="" type="checkbox"/>	FYP-54 Design and develop database structure, create initial ERD	PROJECT CONSTRUCTI...

Figure 57 Sprint 12 - Backlogs (Project Construction)

3.7.3.2 Important Screenshots (Sprint 12):

```

import csv
from pymongo import MongoClient
from pymongo.errors import BulkWriteError

client = MongoClient("mongodb+srv://sharams:Su9868797972@atlascluster.abktl4t.mongodb.net/")

db = client["SCRAPEQUEST"]

def process_collection(file_path, collection_name, headers):
    collection = db[collection_name]
    # Drop the collection if it already exists
    db.drop_collection(collection_name)

    documents = []
    count = 0
    try:
        with open(file_path, 'r', encoding='utf-8') as csvfile:
            reader = csv.DictReader(csvfile)
            for each in reader:
                row = {field: each.get(field, "").strip() for field in headers}
                documents.append(row)
                if len(documents) == 100:
                    collection.insert_many(documents)
                    count += len(documents)
                    documents = []
    except FileNotFoundError:
        print(f"Error: File not found {file_path}")
    except csv.Error as e:
        print(f"CSV read error: {e}")
    except BulkWriteError as bwe:
        print(f"Bulk write error: {bwe.details}")
    except Exception as e:
        print(f"An error occurred: {e}")

collections = [
    ('../ScrapeBots/ProjectData/ADB.csv', 'ADB', ['Name of the Project', 'Link to the project', 'Project Status', 'Project Description']),
    ('../ScrapeBots/ProjectData/SDC.csv', 'SDC', ['Name of the Project', 'Link to the project', 'Duration of the Project', 'Project Details']),
    ('../ScrapeBots/ProjectData/UKAID.csv', 'UKAID', ['Name of the Project', 'Link to the project', 'Project Details', 'Start Date']),
    ('../ScrapeBots/ProjectData/UNDP.csv', 'UNDP', ['Name of the Project', 'Link to the project']),
    ('../ScrapeBots/ProjectData/WB.csv', 'WB', ['Name of the Project', 'Link to the project', 'Procurement Details', 'Link to the Procurement', 'Date of Publication']),
    ('../ScrapeBots/ProjectData/USAID.csv', 'USAID', ['Name of the Project', 'Duration of the Project', 'Sector']),
    ('../ScrapeBots/ProjectData/UNFAO.csv', 'UNFAO', ['Name of the Project', 'Duration of the Project', 'Link to the project']),
    ('../ScrapeBots/ProjectData/JICA.csv', 'JICA', ['Name of the Project', 'Link to the project'])
]

for file_path, collection_name, headers in collections:
    process_collection[file_path, collection_name, headers]

```

Figure 58 Migration Script

This code imports necessary libraries for CSV file operations and MongoDB interactions and sets up a MongoDB client. It defines a function called process_collection, which accepts a file path, collection name, and headers, cleans the collection, reads the content, creates documents, batches them, and handles errors. The function tracks insertions, prints the total, and iterates over a list of collections, calling process_collection for each tuple. The output displays the number of documents inserted or the error message.

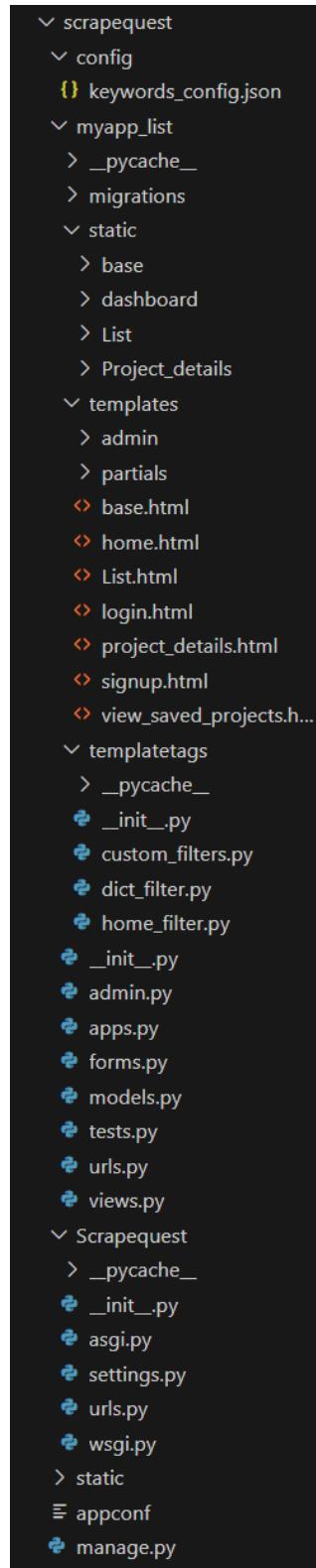
<input type="checkbox"/>	▼ FYP Sprint 13 11 Jan – 23 Jan (8 issues)	
To create the design artifacts that will guide the construction of the system by defining the system architecture to outline how the components will interact, creating mock-ups for the user interface to visualize the end		
<input checked="" type="checkbox"/>	FYP-55 Finalize use cases	PROJECT CONSTRUCTI...
<input checked="" type="checkbox"/>	FYP-56 Finalize the system flow and logic	PROJECT CONSTRUCTI...
<input checked="" type="checkbox"/>	FYP-57 Document the overview of system architecture, and flow	PROJECT CONSTRUCTI...
<input checked="" type="checkbox"/>	FYP-58 Take feedbacks from client and supervisors	PROJECT CONSTRUCTI...
<input checked="" type="checkbox"/>	FYP-59 Create Mock-up designs of the final system	PROJECT CONSTRUCTI...
<input checked="" type="checkbox"/>	FYP-60 Create wireframes for pages of Web-application	PROJECT CONSTRUCTI...
<input checked="" type="checkbox"/>	FYP-61 Finalize logo, color palate, and UI design of webpages.	PROJECT CONSTRUCTI...

Figure 59 Sprint 13 - Backlogs (Project Construction)

<input type="checkbox"/>	▼ FYP Sprint 14 23 Jan – 30 Jan (4 issues)	
To start Django project, integrating MongoDB and create a HTML page listing projects by organization		
<input checked="" type="checkbox"/>	FYP-63 Start Django Project	PROJECT CONSTRUCTI...
<input checked="" type="checkbox"/>	FYP-64 Research on integrating Django with MongoDB	PROJECT CONSTRUCTI...
<input checked="" type="checkbox"/>	FYP-65 Work on fetching list of projects by organizations	PROJECT CONSTRUCTI...
<input checked="" type="checkbox"/>	FYP-66 Create HTML page for listing projects by organizations	PROJECT CONSTRUCTI...

Figure 60 Sprint 14 - Backlogs (Project Construction)

3.7.3.3 Important Screenshots (Sprint 13-14)



```
scrapequest
├── config
│   └── keywords_config.json
├── myapp_list
│   ├── __pycache__
│   ├── migrations
│   └── static
│       ├── base
│       ├── dashboard
│       ├── List
│       └── Project_details
├── static
└── templates
    ├── admin
    ├── partials
    ├── base.html
    ├── home.html
    ├── List.html
    ├── login.html
    ├── project_details.html
    ├── signup.html
    └── view_saved_projects.h...
├── templatetags
│   ├── __pycache__
│   ├── __init__.py
│   ├── custom_filters.py
│   ├── dict_filter.py
│   ├── home_filter.py
│   ├── __init__.py
│   ├── admin.py
│   ├── apps.py
│   ├── forms.py
│   ├── models.py
│   ├── tests.py
│   ├── urls.py
│   └── views.py
└── Scrapequest
    ├── __pycache__
    ├── __init__.py
    ├── asgi.py
    ├── settings.py
    ├── urls.py
    └── wsgi.py
    └── static
        └── appconf
    └── manage.py
```

Figure 61 Django Project Directory

```

from django.shortcuts import render
from django.shortcuts import redirect
import datetime

##mongo_to_render
from pymongo import MongoClient
client = MongoClient("mongodb+srv://sharams:Su98
db = client['SCRAPEQUEST']

```

Figure 62 Integrating Django with MongoDB

The above snippet connects Django with MongoDB database using pymongo. Mongo Atlas' connection URI is used to establish connection.

```

#fetching data from MongoDB
#UNDP
UNDP = db['UNDP']
UNFAO = db['UNFAO']
USAID = db['USAID']
WB = db['WB']
ADB = db['ADB']
JICA = db['JICA']
UKAID = db['UKAID']
SDC = db['SDC']

from django.contrib.auth.decorators import login_required

@login_required
def display_data(request):

    # Fetch data from the collection
    UNDPprojects = list(UNDP.find({}, {'_id': 0}))

    for UNDPproject in UNDPprojects:
        UNDPproject['Name'] = UNDPproject.pop('Name of the Project')
        UNDPproject['Link'] = UNDPproject.pop('Link to the project')
    pass

    # Fetch data from the collection
    UNFAOprojects = list(UNFAO.find({}, {'_id': 0}))
    for UNFAOproject in UNFAOprojects:
        UNFAOproject['Name'] = UNFAOproject.pop('Name of the Project')
        UNFAOproject['Duration'] = UNFAOproject.pop('Duration of the Project')
        UNFAOproject['Link'] = UNFAOproject.pop('Link to the project')
    pass

    # Fetch data from the collection
    USAIDprojects = list(USAID.find({}, {'_id': 0}))
    for USAIDproject in USAIDprojects:
        USAIDproject['Name'] = USAIDproject.pop('Name of the Project')
        USAIDproject['Duration'] = USAIDproject.pop('Duration of the Project')
        USAIDproject['Sector'] = USAIDproject.pop('Sector')
    pass

    # Fetch data from the collection
    WBprojects = list(WB.find({}, {'_id': 0}))
    for WBproject in WBprojects:
        WBproject['Name'] = WBproject.pop('Name of the Project')
        WBproject['Project_Link'] = WBproject.pop('Link to the project')
        WBproject['Details'] = WBproject.pop('Procurement Details')
        WBproject['Procurement_Link'] = WBproject.pop('Link to the Procurement')
        WBproject['Published_Date'] = WBproject.pop('Date of Publication')
    pass

```

Figure 63 Fetching List of Project Filtered Via Organization

```

ADBprojects = list(ADB.find({}, {'_id': 0}))
for ADBproject in ADBprojects:
    ADBproject['Name'] = ADBproject.pop('Name of the Project')
    ADBproject['Link'] = ADBproject.pop('Link to the project')
    ADBproject['Status'] = ADBproject.pop('Project Status')
    ADBproject['Desc'] = ADBproject.pop('Project Description')
pass
# Fetch data from the collection
JICAprojects = list(JICA.find({}, {'_id': 0}))
for JICApject in JICAprojects:
    JICApject['Name'] = JICApject.pop('Name of the Project')
    JICApject['Link'] = JICApject.pop('Link to the project')
pass
# Fetch data from the collection
UKAIDprojects = list(UKAID.find({}, {'_id': 0}))
for UKAIDproject in UKAIDprojects:
    UKAIDproject['Name'] = UKAIDproject.pop('Name of the Project')
    UKAIDproject['Link'] = UKAIDproject.pop('Link to the project')
    UKAIDproject['Date'] = UKAIDproject.pop('Start Date')
    UKAIDproject['Desc'] = UKAIDproject.pop('Project Details')
pass
# Fetch data from the collection
SDCprojects = list(SDC.find({}, {'_id': 0}))
for SDCproject in SDCprojects:
    SDCproject['Name'] = SDCproject.pop('Name of the Project')
    SDCproject['Link'] = SDCproject.pop('Link to the project')
    SDCproject['Duration'] = SDCproject.pop('Duration of the Project')
    SDCproject['Desc'] = SDCproject.pop('Project Details')
pass
# Pass data to the template
context = {'UNFAOprojects': UNFAOprojects,
           'UNDPprojects': UNDPprojects,
           'USAIDprojects': USAIDprojects,
           'WBprojects': WBprojects,
           'ADBprojects': ADBprojects,
           'JICAprojects': JICAprojects,
           'UKAIDprojects': UKAIDprojects,
           'SDCprojects': SDCprojects,
           }
}

return render(request, 'List.html', context)

```

Figure 64 Fetching List of Project Filtered Via Organization

Above code snippets, fetches the lists of projects filtered via organization, from Mongo Atlas by establishing connection. All the collections within the database is searched by initializing fields within the document to a variable and then passed to HTML template to render the data.

<input type="checkbox"/>	▼ FYP Sprint 15 31 Jan – 7 Feb (4 issues)	To create HTML page which would filter lists of projects based upon pre-defined keywords, and implement fully functional search bar.
<input checked="" type="checkbox"/>	FYP-67 Write views function to query database and filter organizations based on predefined keywords	PROJECT CONSTRUCTI...
<input checked="" type="checkbox"/>	FYP-68 Create views function to seamlessly search projects from database using AJAX, based on user input	PROJECT CONSTRUCTI...
<input checked="" type="checkbox"/>	FYP-69 Create base page, and implement search input button in navbar and test the created functionality.	PROJECT CONSTRUCTI...
<input checked="" type="checkbox"/>	FYP-70 Create HTML page to display projects based on predefined keywords.	PROJECT CONSTRUCTI...

Figure 65 Sprint 15 - Backlogs (Project Construction)

3.7.3.4 Important Screenshots (Sprint 15)

```

def parse_date(project, organization):
    date_str = None
    formats = [] # Initialize formats to an empty list

    if organization == 'UNFAO':
        # Extract the start date from the duration using " - " as the separator
        duration = project.get('Duration of the Project', '')
        if " - " in duration:
            date_str = duration.split(" - ")[0]
            formats = ['%d-%b-%Y', '%d %Y'] # 20-Jun-2022, Jul 2019
    elif organization == 'USAID':
        # Check if the duration field is not empty and use " - " as the separator
        duration = project.get('Duration of the Project', '')
        if duration and " - " in duration:
            date_str = duration.split(" - ")[0]
            formats = ['%Y-%m-%d'] # 2006-09-29
    elif organization == 'WB':
        date_str = project.get('Date of Publication', '')
        formats = ['%B %d, %Y'] # March 12, 2023
    elif organization == 'SDC':
        date_str = project.get('Duration of the Project', '')
        formats = ['%d.%m.%Y'] # 31.12.2026
    elif organization == 'UKAID':
        date_str = project.get('Start Date', '')
        formats = ['%Y-%m-%d'] #2023-4-1
    elif organization == 'ADB':
        # Extract the date after "Posting date:"
        description = project.get('Project Description', '')
        marker = "Posting date."
        if marker in description:
            start_index = description.find(marker) + len(marker)
            end_index = description.find(".", start_index)
            if end_index > start_index:
                date_str = description[start_index:end_index].strip()
        formats = ['%d %b %Y'] # 10 Aug 2022

    for fmt in formats:
        if date_str: # Ensure date_str is not None or empty
            try:
                return datetime.datetime.strptime(date_str, fmt)
            except ValueError:
                continue

    return None

def sort_projects(projects):
    # This function will place None values at the end when sorting in descending order
    return sorted(projects, key=lambda x: (x['Date'] is not None, x['Date']), reverse=True)

```

Figure 66 Function to Sort Projects by Date

The function in above snippets sort projects by recent dates while filtering via keywords. Based on the complexity of the data collected, functions are customized accordingly to get as accurate results as possible.

```

from django.contrib.auth.decorators import login_required
from django.conf import settings
import json

@login_required
def list_projects_by_keyword(request):
    # Load keywords from the configuration file
    keywords = load_keywords()

    # handling requests
    keyword = request.GET.get('keyword', '') # Get the keyword from the request
    results = {}

    # logic for fetching and organizing projects
    results = {keyword: {'full_matches': [], 'partial_matches': []} for keyword in keywords}

    for organization, fields in organization_fields.items():
        collection = db[organization]
        for keyword in keywords:
            # Define additional search fields based on the organization
            search_fields = ['Name of the Project']
            if organization == 'WB':
                search_fields.append('Procurement Details')
            elif organization == 'ADB':
                search_fields.append('Project Description')
            elif organization == 'SDC':
                search_fields.append('Project Details')
            elif organization == 'UKAID':
                search_fields.append('Project Details')

            # Perform searches on all designated fields
            for field in search_fields:
                full_keyword_query = {field: {"$regex": f"\b{keyword}\b", "$options": "i"}}
                projects = collection.find(full_keyword_query)

                for project in projects:
                    project_info = {field: project.get(field, 'N/A') for field in fields}
                    project_info['Organization'] = organization
                    project_info['Date'] = parse_date(project, organization) # Parse the date

                    if project_info not in results[keyword]['full_matches']:
                        results[keyword]['full_matches'].append(project_info)

            # Partial keyword search
            for word in keyword.split():
                partial_keyword_query = {field: {"$regex": f"{word}", "$options": "i"}}
                projects = collection.find(partial_keyword_query)

                for project in projects:
                    project_info = {field: project.get(field, 'N/A') for field in fields}
                    project_info['Organization'] = organization
                    project_info['Date'] = parse_date(project, organization) # Parse the date

                    if project_info not in results[keyword]['full_matches'] and project_info not in results[keyword]:
                        results[keyword].append(project_info)

    for keyword in results:
        results[keyword]['full_matches'] = sort_projects(results[keyword]['full_matches'])
        results[keyword]['partial_matches'] = sort_projects(results[keyword]['partial_matches'])

    return render(request, 'project_details.html', {'results': results})

def load_keywords():
    with open(settings.BASE_DIR / 'config/keywords_config.json', 'r') as file:
        return json.load(file)

```

Figure 67 Function to Filter Projects Via Keywords

Keywords are stored initially in config file, which can be modified by admin exclusively. It is then loaded into the function. The function then queries the database with keywords selected by user in the UI, after which it creates a list of full matches and partial matches of the keyword and returns to the HTML template sorted by the recent dates.

```

##search functionality
search_collections = {
    'UNDP': ['Name of the Project'],
    'UNFAO': ['Name of the Project', 'Duration of the Project'],
    'USAID': ['Name of the Project', 'Duration of the Project', 'Sector'],
    'WB': ['Name of the Project', 'Procurement Details', 'Date of Publication'],
    'ADB': ['Name of the Project', 'Project Status', 'Project Description'],
    'JICA': ['Name of the Project'],
    'SDC': ['Name of the Project', 'Project Details', 'Duration of the Project'],
    'UKAID': ['Name of the Project', 'Project Details', 'Start-Date'],
}

for collection_name, fields in search_collections.items():
    # Create a list of tuples for the fields to be indexed
    index_fields = [(field, 'text') for field in fields]

    # Get the collection object
    collection = db[collection_name]

    # Get current text indexes on the collection
    current_indexes = collection.index_information()

    # Prepare a set of fields for the new index for comparison
    new_index_fields = set(index_fields)

    # Flag to determine if an equivalent index exists
    index_exists = False

    # Check each existing index
    for index in current_indexes.values():
        if 'key' in index and set(index['key']) == new_index_fields:
            # An equivalent index exists
            index_exists = True
            break

    # If an equivalent index does not exist, create it
    if not index_exists:
        collection.create_index(index_fields)

from django.http import HttpResponseRedirect, JsonResponse
from django.template.loader import render_to_string

def custom_search(request):
    search_query = request.GET.get('search', '').strip()
    results = {}

    if search_query:
        for collection_name, fields in search_collections.items():
            if collection_name in db.list_collection_names():
                collection = db[collection_name]
                # Perform a text search
                projects = collection.find(
                    {"$text": {"$search": search_query}},
                    {"_id": 0}
                ).sort([{"textScore": 1, "meta": "textScore"}])
                results[collection_name] = list(projects)

    if request.headers.get('X-Requested-With') == 'XMLHttpRequest':
        html = render_to_string('partials/search_results.html', {'results': results})
        return JsonResponse({'html': html})
    else:
        return render(request, 'base.html', {'results': results})

```

Figure 68 Function to handle seamless custom search.

The function above first creates text indexes of collection, if it already doesn't exist and then uses the created index to match custom user queries with projects in database. It then responds the matched result to partial HTML page which is set to forward AJAX request to enable seamless search.

<input type="checkbox"/>	▼ FYP Sprint 16 8 Feb – 15 Feb (3 issues)
To integrate previously created pages in the application and work on its homepage, creating a prototype of the final product.	
<input checked="" type="checkbox"/>	FYP-71 Integrate previously created pages in base page to create a prototype of final application
	PROJECT CONSTRUCTI...
<input checked="" type="checkbox"/>	FYP-72 Take feedback from supervisors and client regarding the created prototype.
	PROJECT CONSTRUCTI...
<input checked="" type="checkbox"/>	FYP-73 Work on home page of the application by integrating charts with insights of projects in database.
	PROJECT CONSTRUCTI...

Figure 69 Sprint 16 - Backlogs (Project Construction)

3.7.3.5 Important Screenshots (Sprint 16)

```
#visualization for homepage
from collections import Counter
from django.shortcuts import render
from django.contrib.auth.decorators import login_required

@login_required
def dashboard_view(request):
    collections = ['UNDP', 'UNFAO', 'USAID', 'WB', 'ADB', 'JICA', 'SDC', 'UKAID']
    project_counts = {collection: db[collection].count_documents({}) for collection in collections}

    keywords = load_keywords()

    # Aggregate data for keywords across all collections
    keyword_counts = Counter()
    for collection in collections:
        for keyword in keywords:
            keyword_counts[keyword] += db[collection].count_documents({'$text': {'$search': keyword}})

    # Count projects for 2023 and 2024
    years = [str(year) for year in range(2012, 2025)] # Years from 2012 to 2024
    yearly_counts = {year: 0 for year in years}

    for year in years:
        for collection_name in collections:
            collection = db[collection_name]
            # Use the text index to search for the year as a keyword
            count = collection.count_documents({'$text': {'$search': year}})
            yearly_counts[year] += count

    context = {
        'project_counts': project_counts,
        'keyword_counts': dict(keyword_counts),
        'yearly_counts': yearly_counts,
    }
    return render(request, 'home.html', context)
```

Figure 70 Function to handle visualization charts.

The above code snippet utilizes count_documents to get data of projects per organization, loads keywords from config file and queries each collection to get number of projects per keyword. Similarly, it queries each collection with a hardcoded list of years to get data of projects per year, and passes it on to the homepage, where chart.js is used to visualize the data.

<input type="checkbox"/>	▼ FYP Sprint 17 16 Feb – 23 Feb (7 issues)	To create a Log-in/ Sign-up feature , and admin page and implement functionality to save projects, and re-test the feature.	
<input checked="" type="checkbox"/>	FYP-74 Create Login and Signup form for authentication and authorization		PROJECT CONSTRUCTI...
<input checked="" type="checkbox"/>	FYP-75 Create superuser and initialize admin page.		PROJECT CONSTRUCTI...
<input checked="" type="checkbox"/>	FYP-76 Create models to handle users		PROJECT CONSTRUCTI...
<input checked="" type="checkbox"/>	FYP-77 Test authentication and authorization		PROJECT CONSTRUCTI...
<input checked="" type="checkbox"/>	FYP-78 Implement a save-button functionality to save projects in then created save-project model		PROJECT CONSTRUCTI...
<input checked="" type="checkbox"/>	FYP-79 Refine the functionality to save project according to user		PROJECT CONSTRUCTI...
<input checked="" type="checkbox"/>	FYP-80 Verify the saved projects are visible to admin in the admin page.		PROJECT CONSTRUCTI...

Figure 71 Sprint 17 - Backlogs (Project Construction)

3.7.3.6 Important Screenshots (Sprint 17):

```

def signup(request):
    if request.method == 'POST':
        form = SignUpForm(request.POST)
        if form.is_valid():
            user = form.save(commit=False)
            user.is_active = False # User will be inactive until approved
            user.save()
            return redirect('login') # Redirect to login page after signup
        else:
            form = SignUpForm()
    return render(request, 'signup.html', {'form': form})

def user_login(request):
    if request.method == 'POST':
        form = LoginForm(request.POST)
        if form.is_valid():
            username = form.cleaned_data.get('username')
            password = form.cleaned_data.get('password')
            user = authenticate(username=username, password=password)
            if user is not None:
                if user.is_active:
                    login(request, user)
                    return redirect('home') # Redirect to a home page or dashboard
                else:
                    messages.error(request, 'Your account has not been approved yet.')
            else:
                messages.error(request, 'Invalid username or password.')
        else:
            for field in form.errors:
                form[field].field.widget.attrs['class'] += ' border-red-500'
            messages.error(request, 'Please correct the errors below.')
    else:
        form = LoginForm()
    return render(request, 'login.html', {'form': form})

```

Figure 72 Function to handle Login and Signup

The above code snippet handles the login and signup functionality of the webpage. When, signup form is posted, it first validates the form and then saves the user without active status and redirects to the login page. For the user to login, admin has to manually provide the active status. Once the active status is achieved, the user can login to system with valid credentials only, else error is shown.

```

class Project(models.Model):
    content = models.TextField() # This will store all relevant project information
    user = models.ForeignKey(User, on_delete=models.CASCADE)

    def __str__(self):
        return self.content

```

Figure 73 Model to handle User-Saved Projects

```

# save_projects
from django.http import JsonResponse
from django.views.decorators.http import require_POST
from .models import Project
from django.views.decorators.csrf import csrf_exempt

@csrf_exempt
@require_POST
def save_project(request):
    content = request.POST.get('content')
    user = request.user

    # Create a new Project instance and save the form data
    try:
        project = Project(content=content, user=user)
        project.save()
        return JsonResponse({"success": True, "message": "Project saved successfully."})
    except Exception as e:
        # Log the error or handle it as per requirements
        return JsonResponse({"success": False, "message": "Failed to save the project."}, status=400)

from django.shortcuts import render
from .models import Project
from django.contrib.auth.decorators import login_required

@login_required
def view_saved_projects(request):
    projects = Project.objects.filter(user=request.user) # Fetch all projects from the database
    return render(request, 'view_saved_projects.html', {'projects': projects})

```

Figure 74 Function to save projects and view saved projects.

Once the user clicks the save project button, a form containing information of project is posted. Then, the first function above handles the posted content by saving it to the database, returning appropriate response based on the success and failure. Similarly, the saved content is rendered to respective HTML page, handled by the second function.

```

##saved projects
try:
    admin.site.unregister(Project)
except admin.sites.NotRegistered:
    pass

class ProjectAdmin(admin.ModelAdmin):
    list_display = ('get_project_details', 'user',)
    readonly_fields = ('get_project_details',)

    def get_project_details(self, obj):
        # Using BeautifulSoup to clean the content
        soup = BeautifulSoup(obj.content, features="html.parser")

        # Remove the 'Save Project' button and any other unwanted elements
        for button in soup.find_all('button'):
            button.decompose()

        # Optionally remove other unwanted elements here

        # Construct the clean HTML
        clean_html = ''.join(str(tag) for tag in soup)

        # Return safe HTML for the admin interface
        return mark_safe(clean_html)

    get_project_details.short_description = 'Project Details'

    # This makes the custom 'get_project_details' method be used on the detail view as well
    def get_readonly_fields(self, request, obj=None):
        if obj: # This is the detail view
            return ('get_project_details',)
        else: # This is the list view
            return self.readonly_fields

```

Figure 75 View User Saved Projects (Admin)

In the above code snippet, user and saved project details are passed on to the admin page. Since the posted content contains HTML tags, BS4 is used to remove the unwanted tags, which proves to be fruitful while exporting project details.

<input type="checkbox"/>	▼ FYP Sprint 18 24 Feb – 2 Mar (7 issues)	
To create features in admin panel according to requirements from client.		
<input checked="" type="checkbox"/> FYP-01 Create logic in admin page to handle user sign-up, efficiently		PROJECT CONSTRUCT...
<input checked="" type="checkbox"/> FYP-02 Test the implemented authentication and authorization		PROJECT CONSTRUCT...
<input checked="" type="checkbox"/> FYP-03 Create a json file to save lists of keyword and implemented logic to modify it via admin page.		PROJECT CONSTRUCT...
<input checked="" type="checkbox"/> FYP-04 Loaded the json file in views function to allow admin to modify the predefined keywords.		PROJECT CONSTRUCT...
<input checked="" type="checkbox"/> FYP-05 Add function to export saved project details		PROJECT CONSTRUCT...
<input checked="" type="checkbox"/> FYP-06 Add a logout feature, to allow user to successfully logout from the system.		PROJECT CONSTRUCT...
<input checked="" type="checkbox"/> FYP-07 Create a HTML page to view projects saved by users, for future reference.		PROJECT CONSTRUCT...

Figure 76 Sprint 18 - Backlogs (Project Construction)

3.7.3.7 Important Screenshots (Sprint 18)

```
scrapequest > config > {} keywords_config.json > ...
1  [ "Food",
2   "Food Aid",
3   "Food system",
4   "Food security",
5   "Agriculture",
6   "Digital",
7   "Food Science",
8   "Food Loss",
9   "Education",
10  "Migration",
11  "Nutrition",
12  "Health",
13  "Climate Change",
14  "Livelihood",
15  "Disaster Risk Reduction",
16  "Web Application",
17  "Mobile Application",
18  "Home"]
```

Figure 77 Created config file.

```

##keywords
from django.shortcuts import render
from django.http import HttpResponseRedirect
from .forms import KeywordsForm
from .models import KeywordUploadProxy
from django.urls import path
from django.conf import settings
import json
import os

@admin.register(KeywordUploadProxy)
class KeywordsConfigAdmin(admin.ModelAdmin):
    change_list_template = "admin/keywords_config_change_list.html"

    def get_urls(self):
        urls = super().get_urls()
        my_urls = [
            path('edit-keywords/', self.admin_site.admin_view(self.edit_keywords), name='edit_keywords'),
        ]
        return my_urls + urls

    def edit_keywords(self, request):
        # Load keywords from JSON file
        file_path = os.path.join(settings.BASE_DIR, 'config', 'keywords_config.json')
        with open(file_path, 'r') as file:
            keywords = json.load(file)

        if request.method == 'POST':
            form = KeywordsForm(request.POST, keywords=keywords)
            if form.is_valid():
                # Save updated keywords to JSON file
                updated_keywords = [kw.strip() for kw in form.cleaned_data['keywords'].split(',')]
                with open(file_path, 'w') as file:
                    json.dump(updated_keywords, file)
                self.message_user(request, 'Keywords updated successfully.')
                return HttpResponseRedirect(request.path_info)
            else:
                form = KeywordsForm(keywords=keywords)

        context = self.admin_site.each_context(request)
        context['opts'] = self.model._meta
        context['form'] = form
        return render(request, "admin/keywords_edit.html", context)

```

Figure 78 Function to modify Keywords List

In the above snippet, ‘KeywordUploadProxy’ model is registered at first to allow admin to interact with it using a custom template and a new view. Next, a custom view is created which loads keywords from a JSON file and updates it, upon getting a ‘POST’ request. If the request is valid, the data in form updates the JSON file, followed by a confirmation message.

```

@admin.action(description='Export selected to CSV')
def export_csv(self, request):
    queryset = self.model.objects.all()

    response = HttpResponse(content_type='text/csv')
    response['Content-Disposition'] = 'attachment; filename=projects.csv'
    writer = csv.writer(response)

    # Write the header.
    writer.writerow(['ID', 'Project Details', 'User'])

    for obj in queryset:
        # Use BeautifulSoup to parse the HTML content
        soup = BeautifulSoup(obj.content, features="html.parser")

        # Initialize an empty list to collect text lines
        details_lines = []

        # Iterate over paragraph tags or any other logical division of content
        for content in soup.find_all(['p', 'h1', 'h2', 'h3', 'div', 'li']): # Add or remove tags as needed
            # Extract the text from each content section
            content_text = content.get_text(separator=" ", strip=True)
            # Unescape HTML entities like '&amp;' to '&
            content_text = html.unescape(content_text)
            details_lines.append(content_text)

            # If there's an <a> tag, add its text and href as a new line
            for link in content.find_all('a', href=True):
                link_text = f'{link.text} ({link["href"]})'
                details_lines.append(link_text)

        # Join all the details with a newline character
        project_details = "\n".join(details_lines)

        # Get the user associated with the project
        user = obj.user.get_full_name() if obj.user else 'Anonymous'

        # Write row to CSV
        writer.writerow([obj.pk, project_details, user])

    return response

```

Figure 79 Function to handle Exporting to CSV

```

@admin.action(description='Export selected to PDF')
def export_pdf(self, request):
    buffer = BytesIO()
    doc = SimpleDocTemplate(buffer, pagesize=letter)
    styles = getSampleStyleSheet()

    # Define a Paragraph style for hyperlinks
    styles.add(ParagraphStyle(name='Link', textColor=colors.blue, underline=True))

    elements = []

    queryset = self.model.objects.all()
    for obj in queryset:
        soup = BeautifulSoup(str(self.get_project_details(obj)), features="html.parser")

        # Extract text and links from the soup object
        content_pieces = []
        for elem in soup.recursiveChildGenerator():
            if elem.name == 'a':
                # Make sure the link is complete (e.g., starts with http)
                link = elem['href'] if elem.has_attr('href') else ''
                # Only add text and a hyperlink if there's an actual link
                if link:
                    content_pieces.append(Paragraph(f'<a href="{link}" color="blue">{elem.text}</a>', styles['Link']))
            elif elem.name is None: # NavigableString objects have None as their name
                text = str(elem).strip()
                if text:
                    content_pieces.append(Paragraph(text, styles['Normal']))

        # Add each piece of content to the elements list
        for content_piece in content_pieces:
            elements.append(content_piece)
            elements.append(Spacer(1, 0.05 * inch))

        # Add extra space after each project
        elements.append(Spacer(1, 0.2 * inch))

    # Build the PDF
    doc.build(elements)
    pdf = buffer.getvalue()
    buffer.close()

    response = HttpResponse(content_type='application/pdf')
    response['Content-Disposition'] = 'attachment; filename=projects.pdf'
    response.write(pdf)

    return response

```

Figure 80 Function to handle exporting to PDF.

The code in the above snippets, sets up a CSV response object using the csv module, extracts text from HTML, and constructs CSV lines. Then, it handles links within the HTML, joins project details into a single string, and retrieves the associated user. The script outputs a row with the object's ID, project details, and username, and returns the response object for download.

Similarly, the PDF Export Action uses SimpleDocTemplate class to generate a PDF document. A StyleSheet is used to define styles, including a custom 'Link' style for hyperlinks. It creates an HttpResponse object with appropriate headers and writes the PDF content to it, which is then returned for download.

```
from django.contrib.auth import logout
from django.shortcuts import redirect

def logout_view(request):
    if request.method == 'POST':
        logout(request)
        return redirect('login')
    else:
        # Optionally, handle incorrect usage:
        return HttpResponseRedirect(['POST'])
```

Figure 81 Function to handle Logout.

In the code above, the system checks if the request method is POST, a security best practice to prevent CSRF attacks. If it is, it terminates the current user session and redirects them to the 'login' page after logging out.

<input type="checkbox"/>	▼ FYP Sprint 19 2 Mar – 8 Mar (4 issues)	
To finalize the Web Application, retesting its features, working on beautification and containerize the application.		
<input checked="" type="checkbox"/>	FYP-08 Test the overall functionality of the web application and ensure everything is upto client's requirements	PROJECT CONSTRUCTI...
<input checked="" type="checkbox"/>	FYP-09 Work on beautifying the webpages according to previously created mockups	PROJECT CONSTRUCTI...
<input checked="" type="checkbox"/>	FYP-00 Use WhiteNoise to serve static files in production mode of django	PROJECT CONSTRUCTI...
<input checked="" type="checkbox"/>	FYP-01 Create Dockerfile to containerize the application	PROJECT CONSTRUCTI...

Figure 82 Sprint 19 - Backlogs (Project Construction)

3.7.3.8 Important Screenshots (Sprint 19)

```
'django.middleware.security.SecurityMiddleware',
'whitenoise.middleware.WhiteNoiseMiddleware',
'django.contrib.sessions.middleware.SessionMiddleware',
```

Figure 83 WhiteNoise Configuration

Above, configuration was done to serve static files in **production** mode.

```
1 # Use an official Python runtime as a parent image
2 FROM python:3.10-slim
3
4 # Set environment variables
5 ENV PYTHONDONTWRITEBYTECODE 1
6 ENV PYTHONUNBUFFERED 1
7 # Set the MongoDB URI
8 ENV MONGO_URI mongodb+srv://
      sharams:Su9860797972@atlascluster.abktl4t.mongodb.net/
9
10 # Set the working directory in the container
11 WORKDIR /code
12
13 # Install dependencies
14 COPY requirements.txt /code/
15 RUN pip install --no-cache-dir -r requirements.txt
16
17 # Copy the Django project into the working directory in the container
18 COPY scrapequest /code/
19
20 # Make port 8000 available to the world outside this container
21 EXPOSE 8000
22
23 # Run the app. CMD is required to run on Heroku
24 CMD ["python", "manage.py", "runserver", "0.0.0.0:8000"]
```

Figure 84 Dockerfile Configuration

Above, configuration creates a distributable image of the web application, making it platform independent.

<input type="checkbox"/>	▼ FYP Sprint 20 3 Mar – 10 Mar (7 issues)	
	Initiate black-box testing of the application	
<input checked="" type="checkbox"/>	FYP-92 Finalize the test cases	PROJECT CONSTRUCTI...
<input checked="" type="checkbox"/>	FYP-93 Test the overall usability of the application, including serving of static files and if the application is containerized properly.	PROJECT CONSTRUCTI...
<input checked="" type="checkbox"/>	FYP-94 Test the working of implemented features	PROJECT CONSTRUCTI...
<input checked="" type="checkbox"/>	FYP-95 Test the form validation of created forms.	PROJECT CONSTRUCTI...
<input checked="" type="checkbox"/>	FYP-96 Test the authentication/authorization features.	PROJECT CONSTRUCTI...
<input checked="" type="checkbox"/>	FYP-97 Test functionality of admin page.	PROJECT CONSTRUCTI...
<input checked="" type="checkbox"/>	FYP-98 Test functionality of buttons, and search input	PROJECT CONSTRUCTI...

Figure 85 Sprint 20 - Backlogs (Project Construction)

<input type="checkbox"/>	▼ FYP Sprint 21 11 Mar – 20 Mar (6 issues)	
	To completely test the web application, scripts for scraping and migration, containerization and static files, and finalize the system	
<input checked="" type="checkbox"/>	FYP-99 Implement Unit testing to test functions with the code.	PROJECT CONSTRUCTI...
<input checked="" type="checkbox"/>	FYP-100 Test working of all the scraping scripts	PROJECT CONSTRUCTI...
<input checked="" type="checkbox"/>	FYP-101 Test migration script, if data is being migrated to the database properly.	PROJECT CONSTRUCT...
<input checked="" type="checkbox"/>	FYP-102 Test the views functions within Django Project	PROJECT CONSTRUCTI...
<input checked="" type="checkbox"/>	FYP-103 Check the routes within web app	PROJECT CONSTRUCTI...
<input checked="" type="checkbox"/>	FYP-104 Final test to check usability of the implemented features	PROJECT CONSTRUCTI...

Figure 86 Sprint 21 - Backlogs (Project Construction)

3.7.4 Project Transition

The tasks in this phase have been carried out in two sprints which is as follows:

<input type="checkbox"/> FYP Sprint 22 21 Mar – 2 Apr (4 issues)	
To deliver the final system to the client, and work on received feedbacks.	
<input checked="" type="checkbox"/> FYP-105 Deliver the project to client	PROJECT TRANSITION
<input checked="" type="checkbox"/> FYP-106 Collect feedback and analyze the feedback	PROJECT TRANSITION
<input checked="" type="checkbox"/> FYP-107 Implement feedback after analysis or list them down to work on it in the future.	PROJECT TRANSITION
<input checked="" type="checkbox"/> FYP-108 Tiny bit of upgrades and adjustments according to the received feedback	PROJECT TRANSIT

Figure 87 Sprint 22 - Backlogs (Project Transition)

<input type="checkbox"/> FYP Sprint 23 3 Apr – 24 Apr (2 issues)	
To complete documentation of the final report.	
<input checked="" type="checkbox"/> FYP-109 Start the documentation for the final report.	PROJECT TRANSITION
<input checked="" type="checkbox"/> FYP-110 Create comprehensive document for client to brief on the system.	PROJECT TRANSI

Figure 88 Sprint 23 - Backlogs (Project Transition)

4 Chapter 4: Testing and Analysis

4.1 Test Plan

4.1.1 Unit Testing, Test Plan

Test Cases	Objectives
Scraping Bots	
1	To test whether the scraping bot is scraping accurate and reliable project information from organization website. (ADB, JICA, SDC, UKAID, UNDP, UNFAO, USAID, WB)
2	To test whether the scraped data is cleaned accordingly.
Database Migration	
3	To test the connection to MongoDB Atlas using connection string.
4	To test whether the cleaned scraped data is successfully migrated to MongoDB Atlas.
Web Application	
5	To test whether the system validates sign-up forms.
6	To test whether the system validates login forms.
7	To test whether the user can log into the system with invalid credentials.
8	To test whether the unauthorized user can log into the system.
9	To test whether the user can navigate to webpages directly without logging in.
10	To test whether the logged-in user is recognized by the system.
11	To test whether the results container appears upon clicking search button and disappears upon clicking elsewhere.
12	To test the accuracy of data displayed in dashboard.
13	To test whether the projects displayed in filter via predefined keywords are sorted according to dates.
14	To test whether the buttons correctly toggle in the pages where project data are

	displayed.
15	To test accuracy of data filtered based on predefined keywords.
16	To test accuracy of data filtered based on organizations.
17	To test whether confirmation message is displayed upon clicking “save project” button.
18	To test whether the user is redirected to appropriate link upon viewing project.
19	To test the functionality of Logout Button.
20	To test whether unauthorized user can login to admin panel of the system.
21	To test whether user can login to admin panel with incorrect credentials.
22	To test whether admin can delete users.
23	To test whether admin can manually add users.
24	To test whether admin can modify user account permissions.
25	To test whether admin gets confirmation upon modifying predefined keywords.
26	To test whether visualization and keywords button change upon modifying keywords.
27	To test whether admin can export selected projects in respective formats.
28	To test whether all of the admin actions are logged.
29	To test whether admin can login to the system as user and access all of the user functionality.
Containerization	
30	To test whether the image is successfully created from Dockerfile without any issue.
31	To test whether the created image is successfully run.
32	To test whether the image is successfully exported to .tar file.
33	To test whether the image can be loaded into another system.

Table 6 Unit Testing Plan

4.1.2 System Testing, Test Plan

Test Cases	Objectives
Platform Independent	
1	To run Scrapequest system in Windows Platform.
2	To run Scrapequest system in Linux Platform.
Web Application	
3	To test ability to view the analytics dashboard easily.
4	To test ability to search projects and view details accurately & easily.
4	To test ability to view the page displaying projects filtered via organization, easily.
5	To test ability to view the page displaying projects filtered via keywords, easily.
6	To test ability to view user saved projects, easily.
8	To test admin's ability to view admin dashboard, and access exclusive features.
9	To test whether the app supports debug mode.
10	To test whether the app is responsive and cross-platform compatible.

Table 7 System Testing Plan

4.2 Unit Testing

Test Case 1	
Objective	To test whether the scraping bot is scraping accurate and reliable project information from organization website. (ADB, JICA, SDC, UKAID, UNDP, UNFAO, USAID, WB)
Action	Run the scraping script and review the scraped data.
Expected Test Result	The bot should scrape accurate data from webpages as instructed.
Actual Test Result	Accurate project details are scraped from webpages.
Conclusion	Test Successful.

Table 8 Unit Testing Test Case 1

The screenshot shows the ADB website interface. At the top, there's a header with the ADB logo and a search bar. Below the header, a navigation menu has 'Tenders' selected. The main content area shows a search result for tenders in Nepal. One specific tender is highlighted:

Institutional Development Expert/TA Coordinator
 57130-001; Nepal; Public sector management; Posting date: 12 Apr 2024
 Notice Type: Individual - Consulting
 Status: Closed Deadline: 18 Apr 2024

Below this, another tender notice is listed:

Contract agreement with M/S Tundi-Bidari JV for upgradation, construction and Performance-Based Maintenance works of Adalat Chautari-Tityang-Saalbot-Daha-Bhakunde-Rayadanda-Damek-Bareng Road (ch 0+000 to 15+000) in Baglung District of Gandaki Province.
 48218-011; Nepal; Agriculture, natural resources and rural development; Contract date: 31 Mar 2024
 Notice Type: Contracts Awarded
 Approval Number: Loan 4320
 Executing Agency: Ministry of Urban Development
 Contractor Name: M/S Tundi-Bidari JV
 Address: Kathmandu
 Total Contract Amount (US\$): 2,394,630.47
 Contract Amount Financed by ADB (US\$): 2,095,804.10

Figure 89 Sample of Project Data in ADB Website

Name of the Project,Link to the project,Project Status,Project Description
Institutional Development Expert/TA Coordinator, https://adb.org//node/960071 ,Status
Posting date:
12 Apr 2024"
"Contract agreement with M/S Tundi-Bidari JV for upgradation, construction and Per
31 Mar 2024"
"Contract agreement with M/S VBEPL-ACPL JV for upgradation, construction and Perfo
31 Mar 2024"
Transmission and Distribution Specialist (Smart Grid and ICT Infrastructure Expert
Posting date:
25 Mar 2024"
Horticulturist/Post harvest management expert/ PoP develeopment and training of tr
Posting date:
14 Mar 2024"
Plant Pathologist/PoP development and training of trainers, https://adb.org//node/9
Posting date:
14 Mar 2024"
Agriculture Economist/PoP Development and Training of Trainers, https://adb.org//no
Posting date:
14 Mar 2024"
Soil Scientist/PoP Development and Training of Trainers, https://adb.org//node/9534
Posting date:
14 Mar 2024"
Horticulturist/Nursery management Expert/PoP Development and Training of Trainers,
Posting date:
14 Mar 2024"
Team Leader/Horticulturist/PoP Development and Training of Trainers, https://adb.or
Posting date:
14 Mar 2024"

Figure 90 Scraped Data (ADB)

Major Projects
 Japan's Official Development Assistance (ODA): Rolling Plan for Nepal (PDF)
 (TCP: Technical Cooperation Project; DS: Development Study (TCP); GA: Grant Aid; LA: Loan Aid (Yen Loan); JPP: JICA Partnership Program)

Education
 (JPP) Follow-up project on JICA Partnership Program on Promoting Quality Education through Community based School Management (Save the Children Follow-up Project/Japan)

Health
 (JPP) Project for Promotion of Maternal and Child Health through Community Empowerment [Association of Medical Doctors of Asia (AMDA-Minds)]
 (JPP) Life Improvement in Dhal Village Development Committee(VDC) of Kaski District - Supply Promotion of Safe Water- (Kobe Tokiwa University)

Water Resources and Disaster Management
 (LA) Melamchi Water Supply Project (Water Treatment Plant)
 (TCP) The Project for Capacity Development on Water Supply in Semi-Urban Areas in Nepal
 (JPP) Community Initiatives for Disaster Risk Reduction (Shapla Neer)

Governance
 (TCP) Participatory Watershed Management and Local Governance Project
 (TCP) Project for the Strengthening Monitoring and Evaluation System in Nepal Phase 2 (SMES 2)

Peace-Building
 State Building Support: "Study Tour and Orientation on Constitution Making and Management of Parliament"
 (TCP) Strengthening Community Mediation Capacity for Peaceful and Harmonious Society
 Support Election Commission of Nepal

Figure 91 Sample of Project Data in JICA Website

Name of the Project, Link to the project
Japan's Official Development Assistance (ODA): Rolling Plan for Nepal (PDF),[]
(JPP) Follow-up project on JICA Partnership Program on Promoting Quality Education,[]
(JPP) Project for Promotion of Maternal and Child Health through Community Empowerment [Association of Medical Doctors of Asia (AMDA-Minds)],[]
(LA) Melamchi Water Supply Project (Water Treatment Plant),["https://www.jica.go.jp/..."]
(TCP) Participatory Watershed Management and Local Governance Project,["https://www.jica.go.jp/..."]
"State Building Support: ""Study Tour and Orientation on Constitution Making and Management of Parliament",[]
(TCP) The Project for the Operation and Maintenance of Sindhuli Road,"["https://www.jica.go.jp/..."]
(TCP) Nationwide Master Plan Study on the storage type hydroelectric power development,[]
(TCP) The Project for the Master Plan Study on High Value Agriculture Extension,[]
(TCP) Gender Mainstreaming and Social Inclusion Project (GeMSIP),["https://www.jica.go.jp/..."]
(GA) Project for the Introduction of Clean Energy by Solar Electricity Generation,[]
Japan/JICA NPPR Background Paper (FY2010) (PDF/531KB),['https://www.jica.go.jp/...']
Map of JICA Major Projects in Nepal (PDF/51KB),['https://libportal.jica.go.jp/...']

Figure 92 Scraped Data (JICA)

The screenshot shows the official website of the Swiss Federal Department of Economic Affairs, Education and Research (EAE) for Switzerland and Nepal. The top navigation bar includes links to 'The Federal Council', 'FDFA', and 'Switzerland around the world'. The main header features the Swiss flag and the text 'Schweizerische Eidgenossenschaft', 'Confédération suisse', 'Confederazione Svizzera', and 'Confederaziun svizra'. The page title is 'Switzerland and Nepal'. A search bar and language selection ('English') are also present.

The main content area is titled 'Projects' and includes a sidebar with links to 'International cooperation', 'Swiss Cooperation Programme 2023-26', 'Themes', 'Projects' (which is selected), 'Partnerships and mandates', and 'Switzerland's engagement worldwide'. The main content area features a filter section with fields for 'Search term', 'Topic' (set to 'Selection'), 'Subtopic' (set to 'Selection'), and 'Status' (checkboxes for 'Planned', 'Active', and 'Completed' are checked). Below the filter is a link to 'Advanced search' and a 'Search' button. The results page shows 'Object 1 – 12 of 70' with a navigation bar for pages 1 through 6. Two project entries are listed:

- MiRiDew - Migrant Rights and Decent Work**
01.07.2023 - 31.12.2026
Remittances sent home by over 4 million migrant workers have significantly contributed to the economic development of Nepal. The proposed project will enhance the capacities of the Government of Nepal and strengthen mechanisms to better protect the rights of workers abroad. There will be a special focus on women's need and the adverse effects of climate change exacerbating the vulnerability of migrants. The project capitalises on Switzerland's longstanding engagement on labour migration in Nepal.
- DIGI Nepal - Digital Innovation for Growth and Inclusion**
01.07.2023 - 30.06.2025
This project Digital Innovation for Growth and Inclusion in Nepal, promotes digital innovation that is essential for an increased access of women and men, including from discriminated groups and beyond the capital, to affordable basic services. Therefore, it provides technical

Figure 93 Sample of Project Data in SDC Website

Name of the Project	Link to the project	Duration of the Project	Project Details
MiRiDew - Migrant Rights and Decent Work,	https://www.eda.admin.ch/countries/nepal	- 31.12.2026",	Remittances sent home by over 4 million people in Nepal.
DIGI Nepal - Digital Innovation for Growth and Inclusion,	https://www.eda.admin.ch/countries/nepal	- 30.06.2025",	This project Digital Innovation for Growth and Inclusion.
Enhanced Skills for Sustainable and Rewarding Employment (ENSSURE) Phase II,	https://www.eda.admin.ch/countries/nepal	- 15.07.2025",	For Nepalese women and men to gain better skills.
2021 Population and Housing Census of Nepal,	https://www.eda.admin.ch/countries/nepal	- 31.12.2024",	The goal is to support the Government of Nepal's efforts to improve the quality of life.
Quality TVET for Youth (Quality),	https://www.eda.admin.ch/countries/nepal/en/home	- 15.07.2026",	The overall goal of the project is to improve the quality of TVET.
Reintegration of Returnee Migrant Workers Project,	https://www.eda.admin.ch/countries/nepal	- 31.05.2027",	The project will contribute to ensuring the safe return of migrant workers.
MLRBP - Motorable Local Roads Bridge Programme,	https://www.eda.admin.ch/countries/nepal	- 31.12.2024",	The project contributes to the connectivity of local roads.
Private Sector-led Economic Recovery from COVID-19,	https://www.eda.admin.ch/countries/nepal	- 31.12.2024",	The project aims to support the private sector.
Nepal Vocational Qualifications System (NVQS) Phase II,	https://www.eda.admin.ch/countries/nepal	- 15.07.2024",	Based on Swiss expertise in Technical and Vocational Education and Training (TVET).
			transparency on the labour market for employers and workers, improve official recognition of workers' competencies in hospitality, construction and automobile sectors and allow workers to move between sectors.
			TVET and general education to access higher level qualifications for better employment opportunities in the domestic and international labour market."
NAMDP - Nepal Agricultural Markets Development Program,	https://www.eda.admin.ch/countries/nepal	- 30.11.2024",	This market systems development program aims to support the agricultural sector.

Figure 94 Scraped Data (SDC)

Results 1 - 20 of 105[Advanced filters](#)**Illegal Wildlife Trade Challenge Fund Round 6**

Department for Environment, Food, and Rural Affairs

Illegal wildlife trade (IWT) is the fifth most lucrative transnational crime, worth up to £17bn a year globally. As well as threatening species with...

[Read more ...](#)

Programme Id	Start date	Status	Total budget
GB-GOV-7-IWTCF-R6	2020-4-1	Implementation	£3,417,064

A contribution to Financial Sector Deepening Africa (FSDA) the United Nations Development Programme Biodiversity Finance Initiative (Biofin) to support delivery of the Kunming-Montreal Global Biodiversity Framework.

Department for Environment, Food, and Rural Affairs

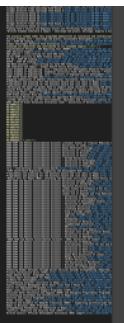
The programme will support low and lower-middle income countries to grow their economies in ways that help to protect and restore their natural capital...

[Read more ...](#)

Programme Id	Start date	Status	Total budget
GB-GOV-7-NPE	2023-2-1	Implementation	£7,200,000

Figure 95 Sample of Project Data in UKAID Website

Name of the Project, [Link to the project](#), [Project Details](#), [Start Date](#)
 Illegal Wildlife Trade Challenge Fund Round 6, <https://devtracker.fcdo.gov.uk/projects/GB-GOV-7-IWTCF-R6>
 A contribution to Financial Sector Deepening Africa (FSDA) the United Nations Development Programme Biodiversity Finance Initiative (Biofin) to support delivery of the Kunming-Montreal Global Biodiversity Framework, <https://devtracker.fcdo.gov.uk/projects/GB-GOV-7-NPE>
 Illegal Wildlife Trade Challenge Fund Round 7, <https://devtracker.fcdo.gov.uk/projects/GB-GOV-7-IWTCF-R7>
 Illegal Wildlife Trade Challenge Fund Round 9, <https://devtracker.fcdo.gov.uk/projects/GB-GOV-7-IWTCF-R9>
 Darwin Initiative Round 27, <https://devtracker.fcdo.gov.uk/projects/GB-GOV-7-DAR27>
 Darwin Initiative Round 28, <https://devtracker.fcdo.gov.uk/projects/GB-GOV-7-DAR28>
 Darwin Initiative Round 29, <https://devtracker.fcdo.gov.uk/projects/GB-GOV-7-DAR29>
 Darwin Initiative Round 24, <https://devtracker.fcdo.gov.uk/projects/GB-GOV-7-DAR24>
 Illegal Wildlife Trade Challenge Fund Round 3, <https://devtracker.fcdo.gov.uk/projects/GB-GOV-7-IWTCF-R3>
 Darwin Initiative Round 23, <https://devtracker.fcdo.gov.uk/projects/GB-GOV-7-DAR23>
 Fleming Fund □ Country and Regional Grants and Fellowships Programme, <https://devtracker.fcdo.gov.uk/projects/GB-GOV-7-FLEMING>
 "British Academy Coherence & Impact - Challenge-led grants: Heritage, Dignity & Violence, <https://devtracker.fcdo.gov.uk/projects/GB-GOV-7-BACI>

*Figure 96 Scrapped Data (UKAID)*



#NextGenUNDP is designed to achieve integrated results at speed and at scale during both quiet and turbulent times.

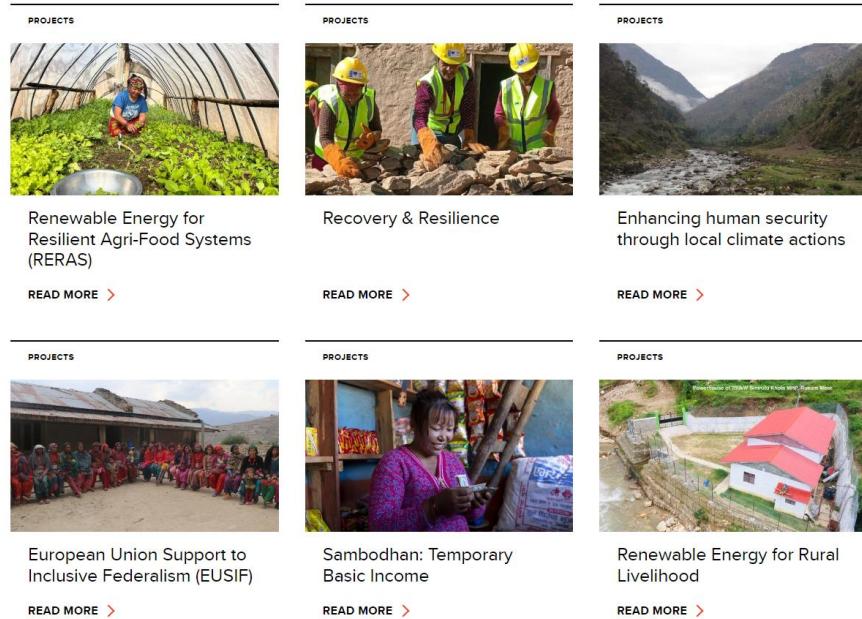


Figure 97 Sample of Project Data in UNDP Website

Name of the Project, [Link to the project](#)

Renewable Energy for Resilient Agro-Food Systems (RERAS) ,<https://www.undp.org/nepal/projects/renewable-energy-for-resilient-agro-food-systems-reras>
 Recovery & Resilience ,<https://www.undp.org/nepal/projects/recovery-and-resilience>
 Enhancing human security through local climate actions ,<https://www.undp.org/nepal/projects/enhancing-human-security-through-local-climate-actions>
 European Union Support to Inclusive Federalism (EUSIF) ,<https://www.undp.org/nepal/projects/european-union-support-to-inclusive-federalism-eusif>
 Sambodhan: Temporary Basic Income ,<https://www.undp.org/nepal/projects/tbi/sambodhan-temporary-basic-income>
 Renewable Energy for Rural Livelihood ,<https://www.undp.org/nepal/projects/renewable-energy-for-rural-livelihood>
 Value Chain Development of Fruit and Vegetables Project (VCDP) ,<https://www.undp.org/nepal/projects/vcdp-value-chain-development-of-fruit-and-vegetables-project>
 " Leaving No One Behind in Nepal's Green, Resilient, and Inclusive COVID-19 Recovery ,<https://www.undp.org/nepal/projects/leaving-no-one-behind-nepals-green-resilient-and-inclusive-covid-19-recovery>
 " Strengthening Urban Preparedness, Earthquake Preparedness and Response in Western Nepal ,<https://www.undp.org/nepal/projects/strengthening-urban-preparedness-earthquake-preparedness-and-response-in-western-nepal>
 Support to Knowledge and Lifelong Learning Skills (SKILLS) Programme ,<https://www.undp.org/nepal/projects/support-to-knowledge-and-lifelong-learning-skills-skills-programme>
 Sustainable Tourism for Livelihood Recovery ,<https://www.undp.org/nepal/projects/sustainable-tourism-for-livelihood-recovery>
 Developing Climate Resilient Livelihoods in the Vulnerable Watershed in Nepal ,<https://www.undp.org/nepal/projects/developing-climate-resilient-livelihoods-in-the-vulnerable-watershed-in-nepal>
 Cooperative Market Development Programme ,<https://www.undp.org/nepal/projects/cooperative-market-development-programme>
 [Closed]Urban Disaster Prepared Project (ECHO III) ,<https://www.undp.org/nepal/projects/closed-urban-disaster-prepared-project-echo-iii>

Figure 98 Scrapped Data (UNDP)

FAO in Nepal

Our office [Programmes and Projects](#) News Resources Our Partners Programmes

Success Stories

Projects list

Project List

UTF/GCP/MTF Projects

- [\[+\] UNJP/NEP/078/UNJ: Accelerating Progress towards Rural Women's Economic Empowerment II](#)
- [\[+\] GCP /NEP/076/GCF – Building a Resilient Churia Region in Nepal \(BRCRN\)](#)
- [\[+\] UNJP/NEP/084/UNJ Assessing the impact of the global crisis on the agriculture and food security in Nepal](#)

GCP /NEP/11126P/GFF: Restoration of Forests and Mountain Ecosystems (ReFaME) in Far-West Nepal (PPG)

Objective: Development of Full Project Proposal on "Restoration of Forests and Mountain Ecosystems in Far-West Nepal."

Donor: Global Environment Facility

Period: July 2023- December 2024

Budget: 100000 USD

Geographic Coverage: Sudurpachim Province

District: Baitadi, Doti and Darchula

Figure 99 Sample of Project Data in UNFAO Website

Name of the Project	Duration of the Project	Link to the project
UNJP/NEP/078/UNJ: Accelerating Progress towards Rural Women's Economic Empowerment		
GCP /NEP/076/GCF	Building a Resilient Churia Region in Nepal (BRCRN), May 2020 to	
UNJP/NEP/084/UNJ	Assessing the impact of the global crisis on the agriculture and	
UTF /NEP/077:	Technical Assistance to Food and Nutrition Security Enhancement Proj	
"GCP /NEP/080/GFF:	Enhancing Capacity for Sustainable Management of Forests, Land	
UTF/NEP/073 -	Technical Assistance to the Agriculture and Food Security Project (A	
GCP/NEP/072:	Reducing Vulnerability and Increasing Adaptive Capacity to Respond th	
TCP/NEP/3801/C1	TCPF: Support to the development of strategic roadmap of intervention	
TCP/NEP/3802/C2:	TCPF: Support to the formulation of a Full Funding Proposal on CF	
TCP/NEP/3803/C3:	Technical Assistance to the Nepal Integrated Agricultural Statist	
TCP/NEP/3804:	Strengthening capacity of public & private sectors on organic agricult	
TCP/NEP/3805/C4	TCPF: Support to the establishment of GIS based data and Informa	
TCP/NEP/3806/C5	TCPF: Support to operationalize the "One Health Strategy 2019" o	
TCP/NEP/3807/C6-	TCPF: Support to strengthen preparedness and response capacity to	
TCP/NEP/3808/C7	TCPF: Support good dairy husbandry practices for dairy sector de	
"TCP/NEP/3809 -	Emergency and early recovery support to floods-affected farming ho	
TCP/NEP/3810/C8	TCPF: Support to formulation of sustainable mountain development a	
OSRO/NEP/401/USA:	Immediate technical assistance to strengthen emergency preparedne	
OSRO/NEP/001:	Immediate TA for animal health system to address emerging / priority	
OSRO/NEP/602/USA-	Building Resilience to Landslides and the Establishment of Early	
OSRO/NEP/401/USA:	Immediate technical assistance to strengthen emergency preparedne	
TCP/NEP/3504 -	Emergency response to restore the rural livelihoods of earthquake-a	
OSRO/NEP/501/BEL -	Nepal Earthquake Flash Appeal 2015, 10/05/2015 - 09/11/2015 (Pro	
OSRO/NEP/502/ITA -	Emergency Assistance for Agricultural Livelihoods in Earthquake	
OSRO/NEP/503/NOR -	Emergency assistance to restore agricultural-based livelihoods	

Figure 100 Scrapped Data (UNFAO)

Activity	Country	Managing Agency	International Sector
MCC Program Management & Support	Nepal	Millennium Challenge Corporation	Energy
MCC Transmission Lines Activity	Nepal	Millennium Challenge Corporation	Energy
MCC Program Administration	Nepal	Millennium Challenge Corporation	Operating Expenses
Momentum 2B: Private Healthcare Delivery	Nepal	U.S. Agency for International Development	Maternal and Child Health, Family Planning
URJA Nepal	Nepal	U.S. Agency for International Development	Energy
Agriculture	Nepal	U.S. Agency for International Development	Agriculture
Basic Education	Nepal	U.S. Agency for International Development	Basic Education

Figure 101 Sample of Project Data in USAID Website

```
"Name of the Project", "Duration of the Project", "Sector"
"MCC Program Management & Support", "", "Energy"
"MCC Transmission Lines Activity", "", "Energy"
"MCC Program Administration", "", "Operating Expenses"
"\"Momentum 2B: Private Healthcare Delivery", "", "Maternal and Child Health, Family Planning"
"URJA Nepal", "", "Energy"
"Agriculture", "", "Agriculture"
"Basic Education", "", "Basic Education"
"Red Book Support Program", "", "Maternal and Child Health, Family Planning"
"Red Book Support Program", "", "Basic Health"
"USAID Pay and Benefits", "", "Operating Expenses"
"Private Sector Productivity", "", "Business and Other Services"
"Water Supply and Sanitation", "", "Water Supply and Sanitation"
"WASH Assistance", "", "Water Supply and Sanitation"
"USAID redacted this field in accordance with the exceptions outlined in the Foreign Aid Act"
"Polio Immunization Funding", "", "Maternal and Child Health, Family Planning"
"Red Book Support Program", "", "Health, General"
"Breakthrough-ACTION", "2017-07-21 - 2025-07-31", "Basic Education"
"WASH Assistance", "", "General Environmental Protection"
"USAID Travel and Transportation", "", "Operating Expenses"
"MCC Not Applicable", "", "Unallocated/ Unspecified"
"USAID redacted this field in accordance with the exceptions outlined in the Foreign Aid Act"
" Col 1: Name of the Project | 2 - 2024-03-11", "Government and Civil Society"
```

Figure 102 Scraped Data (USAID)

Desktop computer-4(2 provincial and 2 PMU office) and Printer

Country:Nepal |

Project Title:Dedicated Grant Mechanism for Indigenous Peoples and Local Communities in Nepal - P171720 |

Notice Type:Contract Award | Language:English | Published Date:April 17, 2024

Construction of; Slice 1: Store Room, Record Room and Compound Wall at Bull Mother shed and Slice 2; Construction of Quarantine Shed at NLBO, Pokhara

Country:Nepal | Project Title:Nepal Livestock Sector Innovation Project - P156797 | Notice Type:Contract Award |

Language:English | Published Date:April 10, 2024

Construction of; Slice 1: Store Room, Record Room and Compound Wall at Bull Mother shed and Slice 2; Construction of Quarantine Shed at NLBO, Pokhara

Country:Nepal | Project Title:Nepal Livestock Sector Innovation Project - P156797 | Notice Type:Contract Award |

Language:English | Published Date:April 10, 2024

Procurement of UV-VIS Spectrophotometer for National Animal Feed and Livestock Quality Management Laboratory (NAFLQML) PMU

Country:Nepal | Project Title:Nepal Livestock Sector Innovation Project - P156797 | Notice Type:Contract Award |

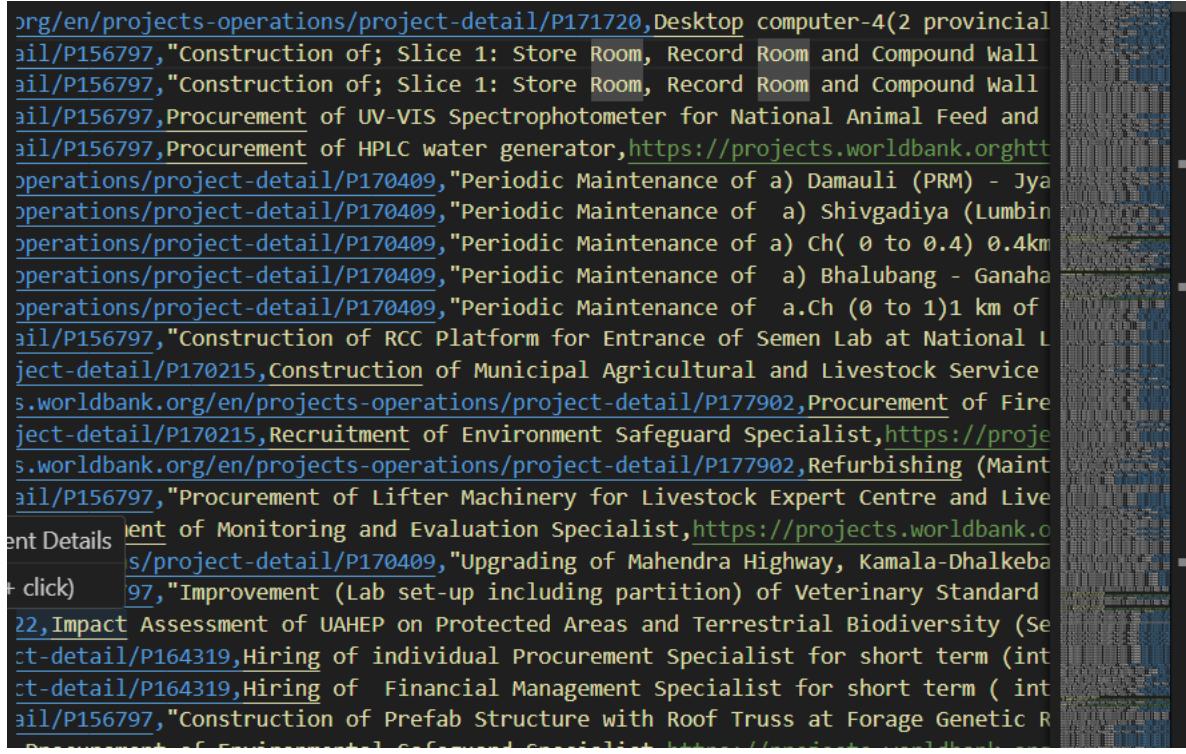
Language:English | Published Date:April 10, 2024

Procurement of HPLC water generator

Country:Nepal | Project Title:Nepal Livestock Sector Innovation Project - P156797 | Notice Type:Contract Award |

Language:English | Published Date:April 10, 2024

Figure 103 Sample of Project Data in WB Website



org/en/projects-operations/project-detail/P171720, Desktop computer-4(2 provincial
ail/P156797, "Construction of; Slice 1: Store Room, Record Room and Compound Wall
ail/P156797, "Construction of; Slice 1: Store Room, Record Room and Compound Wall
ail/P156797, Procurement of UV-VIS Spectrophotometer for National Animal Feed and
ail/P156797, Procurement of HPLC water generator, <https://projects.worldbank.org/>
operations/project-detail/P170409, "Periodic Maintenance of a) Damauli (PRM) - Jya
operations/project-detail/P170409, "Periodic Maintenance of a) Shrivadiya (Lumbin
operations/project-detail/P170409, "Periodic Maintenance of a) Ch(0 to 0.4) 0.4km
operations/project-detail/P170409, "Periodic Maintenance of a) Bhalubang - Ganaha
operations/project-detail/P170409, "Periodic Maintenance of a.Ch (0 to 1)1 km of
ail/P156797, "Construction of RCC Platform for Entrance of Semen Lab at National L
ject-detail/P170215, Construction of Municipal Agricultural and Livestock Service
s.worldbank.org/en/projects-operations/project-detail/P177902, Procurement of Fire
ject-detail/P170215, Recruitment of Environment Safeguard Specialist, <https://proje>
s.worldbank.org/en/projects-operations/project-detail/P177902, Refurbishing (Maint
ail/P156797, "Procurement of Lifter Machinery for Livestock Expert Centre and Live
ent Details ent of Monitoring and Evaluation Specialist, <https://projects.worldbank.o>
s/project-detail/P170409, "Upgrading of Mahendra Highway, Kamala-Dhalkeba
+ click) 97, "Improvement (Lab set-up including partition) of Veterinary Standard
22, Impact Assessment of UAHEP on Protected Areas and Terrestrial Biodiversity (Se
ct-detail/P164319, Hiring of individual Procurement Specialist for short term (int
ct-detail/P164319, Hiring of Financial Management Specialist for short term (int
ail/P156797, "Construction of Prefab Structure with Roof Truss at Forage Genetic R
Procurement of Environmental Safeguard Specialist <https://projects.worldbank.org/>

Figure 104 Scrapped Data (WB)

Test Case 2	
Objective	To test whether the scraped data is cleaned accordingly.
Action	Run the data cleaning & testing python script and review the output in terminal.
Expected Test Result	The script should clean the scraped data and “tests passed successfully” output shall be displayed in terminal.
Actual Test Result	The “tests passed successfully” output was displayed in terminal.
Conclusion	Test Successful.

Table 9 Unit Testing Test Case 2

```

def clean_data(df):
    """ Clean the dataframe """
    # Remove duplicate rows
    df = df.drop_duplicates()
    # Drop rows where all elements are missing
    df = df.dropna(how='all')
    # Replace missing values with a placeholder or an interpolation
    df = df.fillna(method='ffill').fillna('Unknown')
    return df

def test_cleaning(original_df, cleaned_df):
    """ Perform tests to ensure data cleaning was successful """
    assert len(cleaned_df) <= len(original_df), "Cleaned data should not have more rows than original."
    assert cleaned_df.isna().sum().max() == 0, "No NaN values should exist in cleaned data."
    print("Tests passed successfully!")

#usage
directory = '../ScrapeBots/ProjectData/'
file_names = ['ADB.csv', 'JICA.csv', 'SDC.csv', 'UKAID.csv', 'UNDP.csv', 'UNFAO.csv', 'USAID.csv', 'WB.csv']
file_paths = [directory + file_name for file_name in file_names]
all_dataframes = []

for path in file_paths:
    df = load_data(path)
    cleaned_df = clean_data(df)
    test_cleaning(df, cleaned_df)
    all_dataframes.append(cleaned_df)

```

Figure 105 Script to clean the data.

```
c:\Users\User\Documents\GitHub\Scraping\FYP\DataMigration\cleaned_data.py:14:  
    df = df.fillna(method='ffill').fillna('Unknown')  
..../ScrapeBots/ProjectData/ADB.csv Tests passed successfully!  
c:\Users\User\Documents\GitHub\Scraping\FYP\DataMigration\cleaned_data.py:14:  
    df = df.fillna(method='ffill').fillna('Unknown')  
..../ScrapeBots/ProjectData/JICA.csv Tests passed successfully!  
c:\Users\User\Documents\GitHub\Scraping\FYP\DataMigration\cleaned_data.py:14:  
    df = df.fillna(method='ffill').fillna('Unknown')  
..../ScrapeBots/ProjectData/SDC.csv Tests passed successfully!  
c:\Users\User\Documents\GitHub\Scraping\FYP\DataMigration\cleaned_data.py:14:  
    df = df.fillna(method='ffill').fillna('Unknown')  
..../ScrapeBots/ProjectData/UKAID.csv Tests passed successfully!  
c:\Users\User\Documents\GitHub\Scraping\FYP\DataMigration\cleaned_data.py:14:  
    df = df.fillna(method='ffill').fillna('Unknown')  
..../ScrapeBots/ProjectData/UNDP.csv Tests passed successfully!  
c:\Users\User\Documents\GitHub\Scraping\FYP\DataMigration\cleaned_data.py:14:  
    df = df.fillna(method='ffill').fillna('Unknown')  
..../ScrapeBots/ProjectData/UNFAO.csv Tests passed successfully!  
c:\Users\User\Documents\GitHub\Scraping\FYP\DataMigration\cleaned_data.py:14:  
    df = df.fillna(method='ffill').fillna('Unknown')  
..../ScrapeBots/ProjectData/USAID.csv Tests passed successfully!  
c:\Users\User\Documents\GitHub\Scraping\FYP\DataMigration\cleaned_data.py:14:  
    df = df.fillna(method='ffill').fillna('Unknown')  
..../ScrapeBots/ProjectData/WB.csv Tests passed successfully!  
PS C:\Users\User\Documents\GitHub\Scraping\FYP\scrapequest> █
```

Figure 106 Displayed Output in Terminal

Test Case 3	
Objective	To test the connection to MongoDB Atlas using connection string.
Action	Run the python script which attempts to ping the database using MongoDB uri and review the output with status of connection in terminal.
Expected Test Result	The script should print “MongoDB connection successful” in terminal.
Actual Test Result	The “MongoDB connection successful” output was printed in terminal.
Conclusion	Test Successful.

Table 10 Unit Testing Test Case 3

```

from pymongo import MongoClient
from pymongo.errors import ConnectionFailure

def test_mongodb_connection(uri):
    """Attempt to connect to MongoDB Atlas using the provided URI."""
    try:
        # Connect to the MongoDB client
        client = MongoClient(uri, serverSelectionTimeoutMS=5000) # 5 second
        # Attempt to fetch a small amount of data
        client.admin.command('ping')
        print("MongoDB connection successful.")
    except ConnectionFailure as e:
        print(f"MongoDB connection failed: {e}")

# 'uri' = MongoDB Atlas connection string
uri = "mongodb+srv://sharams:Su9860797972@atlascluster.abktl4t.mongodb.net/"
test_mongodb_connection(uri)

```

Figure 107 Script to check connection.

```

on/Python311/python.exe c:/Users/User/Documents/GitHub/Scraping/FYP/DataMigration/MongoAtlasconnection.
py
MongoDB connection successful.

```

Figure 108 Displayed Output

Test Case 4	
Objective	To test whether the cleaned scraped data is successfully migrated to MongoDB Atlas.
Action	Run the python script which attempts to migrate cleaned data in csv to MongoDB Atlas.
Expected Test Result	The script should create collections in Mongo Atlas and populate them with project information.
Actual Test Result	The script created collections in Mongo Atlas and populated them with project information.
Conclusion	Test Successful.

Table II Unit Testing Test Case 4

```

def process_collection(file_path, collection_name, headers):
    collection = db[collection_name]
    # Drop the collection if it already exists
    db.drop_collection(collection_name)

    documents = []
    count = 0
    try:
        with open(file_path, 'r', encoding='utf-8') as csvfile:
            reader = csv.DictReader(csvfile)
            for each in reader:
                row = {field: each.get(field, "").strip() for field in headers}
                documents.append(row)
                # Insert in batches of 100
                if len(documents) == 100:
                    collection.insert_many(documents)
                    count += len(documents)
                    documents = []

        # Insert any remaining documents
        if documents:
            collection.insert_many(documents)
            count += len(documents)

        print(f"Total {count} documents inserted into {collection_name}")
    except FileNotFoundError:
        print(f"Error: File not found {file_path}")
    except csv.Error as e:
        print(f"CSV read error: {e}")
    except BulkWriteError as bwe:
        print(f"Bulk write error: {bwe.details}")
    except Exception as e:
        print(f"An error occurred: {e}")

collections = [
    ('../ScrapeBots/ProjectData/ADB.csv', 'ADB', ['Name of the Project', 'Link to the project', 'Project Status', 'Project Description']),
    ('../ScrapeBots/ProjectData/SDC.csv', 'SDC', ['Name of the Project', 'Link to the project', 'Duration of the Project', 'Project Details']),
    ('../ScrapeBots/ProjectData/UKAID.csv', 'UKAID', ['Name of the Project', 'Link to the project', 'Project Details', 'Start Date']),
    ('../ScrapeBots/ProjectData/UNDP.csv', 'UNDP', ['Name of the Project', 'Link to the project']),
    ('../ScrapeBots/ProjectData/WB.csv', 'WB', ['Name of the Project', 'Link to the project', 'Procurement Details', 'Link to the Procurement', 'Date of Publication']),
    ('../ScrapeBots/ProjectData/USAID.csv', 'USAID', ['Name of the Project', 'Duration of the Project', 'Sector']),
    ('../ScrapeBots/ProjectData/UNFAO.csv', 'UNFAO', ['Name of the Project', 'Duration of the Project', 'Link to the project']),
    ('../ScrapeBots/ProjectData/JICA.csv', 'JICA', ['Name of the Project', 'Link to the project'])
]

for file_path, collection_name, headers in collections:
    process_collection(file_path, collection_name, headers)

```

Figure 109 Script to Migrate Data to DB

```
Total 340 documents inserted into ADB
Total 82 documents inserted into SDC
Total 105 documents inserted into UKAID
Total 37 documents inserted into UNDP
Total 1820 documents inserted into WB
Total 1256 documents inserted into USAID
Total 51 documents inserted into UNFAO
Total 13 documents inserted into JICA
```

Figure 110 Output in Terminal

DATABASES: 1 COLLECTIONS: 8

The screenshot shows the Mongo Atlas Cluster interface. On the left, there's a sidebar with a '+ Create Database' button, a search bar for namespaces, and a tree view under 'SCRAPEQUEST' containing databases: ADB, JICA, SDC, UKAID, UNDP, UNFAO, USAID, and WB. The 'ADB' database is currently selected. The main panel is titled 'SCRAPEQUEST.ADB' and displays storage details: STORAGE SIZE: 44KB, LOGICAL DATA SIZE: 110.22KB, TOTAL DOCUMENTS: 340, and INDEXES TOTAL SIZE: 28KB. It has tabs for 'Find', 'Indexes', 'Schema Anti-Patterns (0)', 'Aggregation', and 'Search Indexes'. Below these tabs, there's a link 'Generate queries from natural language in Compass' and a 'Filter' button with a placeholder 'Type a query: { field: 'value' }'. At the bottom, it says 'QUERY RESULTS: 1-20 OF MANY' and shows a single document snippet:

```
_id: ObjectId('66236e1672cd1e38f5fff528')
Name of the Project: "Institutional Development Expert/TA Coordinator"
Link to the project: "https://adb.org//node/960071"
Project Status: "Status: Closed"
Project Description: "57130-001; Nepal; Public sector management;
Posting date:
12 Apr 2024"
```

Figure 111 Screenshot of Mongo Atlas Cluster

Test Case 5	
Objective	To test whether the system validates sign-up forms.
Action	Input invalid characters in forms.
Expected Test Result	The system should throw errors and the form shall not be submitted.
Actual Test Result	The system threw errors, and the form wasn't submitted.
Conclusion	Test Successful.

Table 12 Unit Testing Test Case 5

Sign Up

Username:

Required. 150 characters or fewer.
@/./+/-/_ only.

! Please fill out this field.

Figure 112 Error upon submitting empty fields.

Email:

Only letters, numbers, and @/./+/-/_

! Please include an '@' in the email address. 'sssss' is missing an '@'.

Your password can't be too similar to your ...

Figure 113 Email Validation

Password confirmation:

...

...

This password is too short. It must contain at least 8 characters.
 This password is too common.
 This password is entirely numeric.

Figure 114 Password Validation

Test Case 6	
Objective	To test whether the system validates login forms.
Action	Input invalid characters in forms.
Expected Test Result	The system should throw errors and the form shall not be submitted.
Actual Test Result	The system threw errors, and the form wasn't submitted.
Conclusion	Test Successful.

Table 13 Unit Testing Test Case 6

Login

Please enter a correct username and password. Note that both fields may be case-sensitive.

Username:

Password:

Login

Figure 115 Form Validation

Username:

Password

 Please fill out this field.

Figure 116 Error on Empty Form Submission

Test Case 7	
Objective	To test whether the user can log into the system with invalid credentials.
Action	Input invalid passwords for authorized user in forms.
Expected Test Result	The system should throw errors and the form shall not be submitted.
Actual Test Result	The system threw errors, and the form wasn't submitted.
Conclusion	Test Successful.

Table 14 Unit Testing Test Case 7

The screenshot shows a user profile page with the following sections:

- sharams**: Username input field containing "sharams".
- Personal info**: Section containing "First name:" (empty), "Last name:" (empty), and "Email address:" (sharamskunwar.sk@gmail.com).
- Permissions**: Section with a checked checkbox labeled "Active". A note below it says: "Designates whether this user should be treated as active. Unselect this instead of deleting accounts."

Figure 117 User with Active Status

The screenshot shows a login interface with the following fields:

- Username:** Input field containing "sharams".
- Password:** Input field containing three dots (...).
- Login**: A large green button labeled "Login".

Figure 118 Credentials Validation

Test Case 8	
Objective	To test whether the user can log into the system with invalid credentials.
Action	Input valid credentials in forms for user without active status.
Expected Test Result	The system should throw errors and the form shall not be submitted.
Actual Test Result	The system threw errors, and the form wasn't submitted.
Conclusion	Test Successful.

Table 15 Unit Testing Test Case 8

sharams1

Username: sharams1

Required: 150 characters or fewer. Letters, digits and @/_/+/- only.

Password: algorithm: pbkdf2_sha256 iterations: 720000 salt: 8L2yIH***** hash: r7Mp+B***** Raw passwords are not stored, so there is no way to see this user's password, but you can change the password using this form.

Personal info

First name:

Last name:

Email address:

Permissions

Active

Designates whether this user should be treated as active. Unselect this instead of deleting accounts.

Figure 119 User with no Active Status

Login

Please enter a correct username and password. Note that both fields may be case-sensitive.

Username:

Password:

Figure 120 Authorization Validation

Test Case 9	
Objective	To test whether the user can navigate to webpages directly without logging in.
Action	Input path to next page directly on the search bar without logging in.
Expected Test Result	The system should redirect user to the login page.
Actual Test Result	The system redirected user to the login page.
Conclusion	Test Successful.

Table 16 Unit Testing Test Case 9

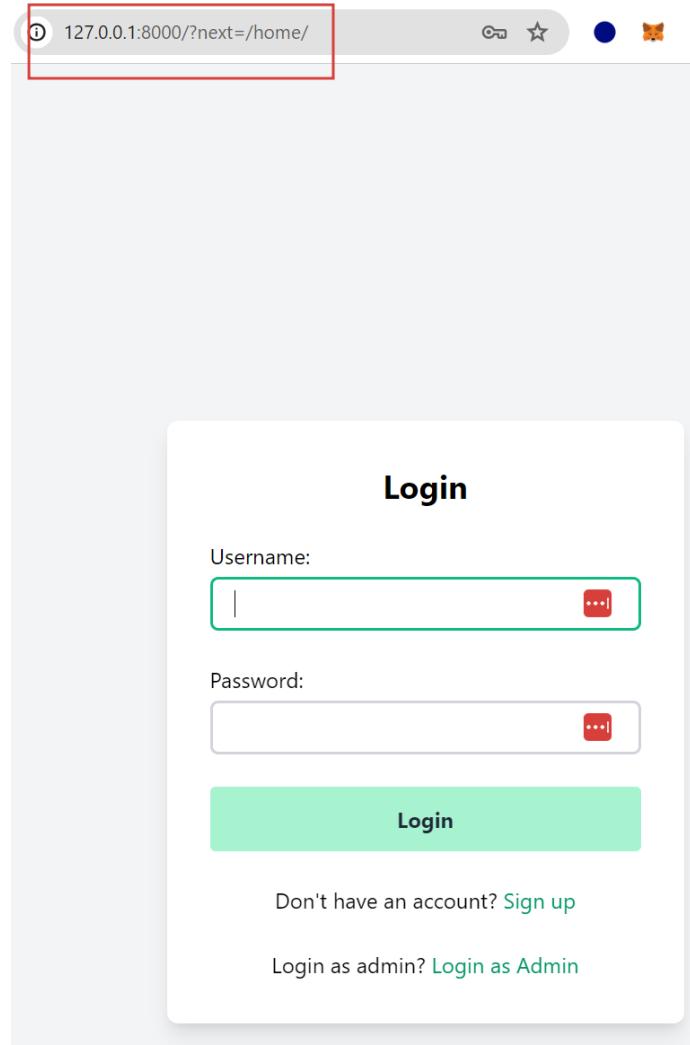


Figure 121 Navigating to home page without logging in.

Test Case 10	
Objective	To test whether the logged-in user is recognized by the system.
Action	Input valid credentials and login to the system.
Expected Test Result	The system should display which user is logged in.
Actual Test Result	The system displayed the username of logged in user.
Conclusion	Test Successful.

Table 17 Unit Testing Test Case 10

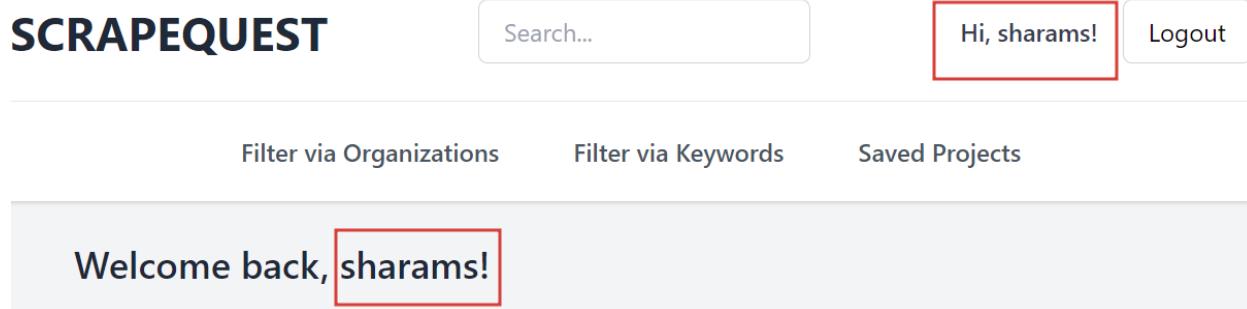


Figure 122 User Recognized by the System

Test Case 11	
Objective	To test the functionality of search button.
Action	Click on the search button in the homepage.
Expected Test Result	A div showing results appears upon clicking search button and disappears upon clicking elsewhere.
Actual Test Result	A div showing results appeared upon clicking search button and disappeared upon clicking elsewhere.
Conclusion	Test Successful.

Table 18 Unit Testing Test Case 11

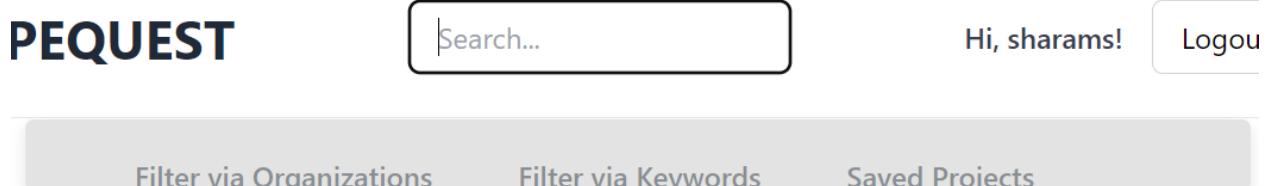


Figure 123 Search Results Container Appearing

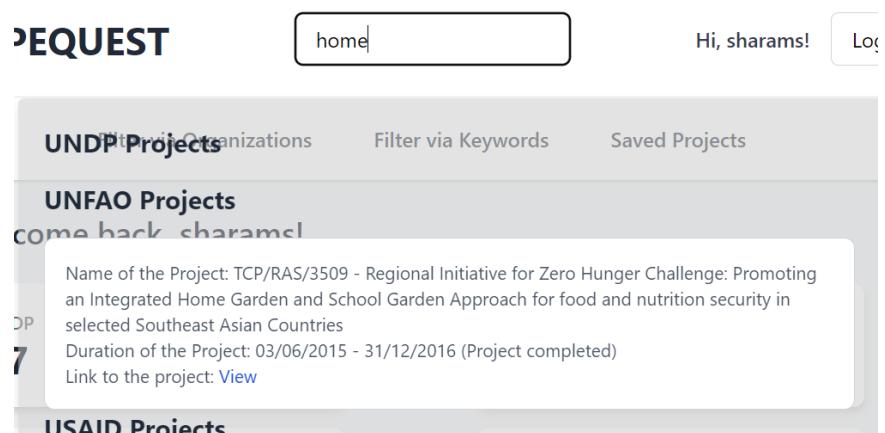


Figure 124 Results Populated in Container

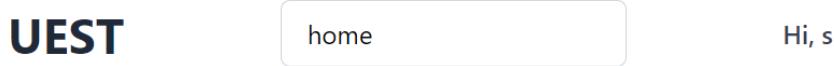


Figure 125 Results Container Disappearing

Test Case 12	
Objective	To test the accuracy of data displayed in dashboard.
Action	Navigate to homepage and view analytics displayed in dashboard.
Expected Test Result	The system should display accurate results as per the projects in database.
Actual Test Result	The system displayed accurate results as per the projects in database.
Conclusion	Test Successful.

Table 19 Unit Testing Test Case 12

```
Total 340 documents inserted into ADB
Total 82 documents inserted into SDC
Total 105 documents inserted into UKAID
Total 37 documents inserted into UNDP
Total 1820 documents inserted into WB
Total 1256 documents inserted into USAID
Total 51 documents inserted into UNFAO
Total 13 documents inserted into JICA
```

Figure 126 Documents Inserted in Database

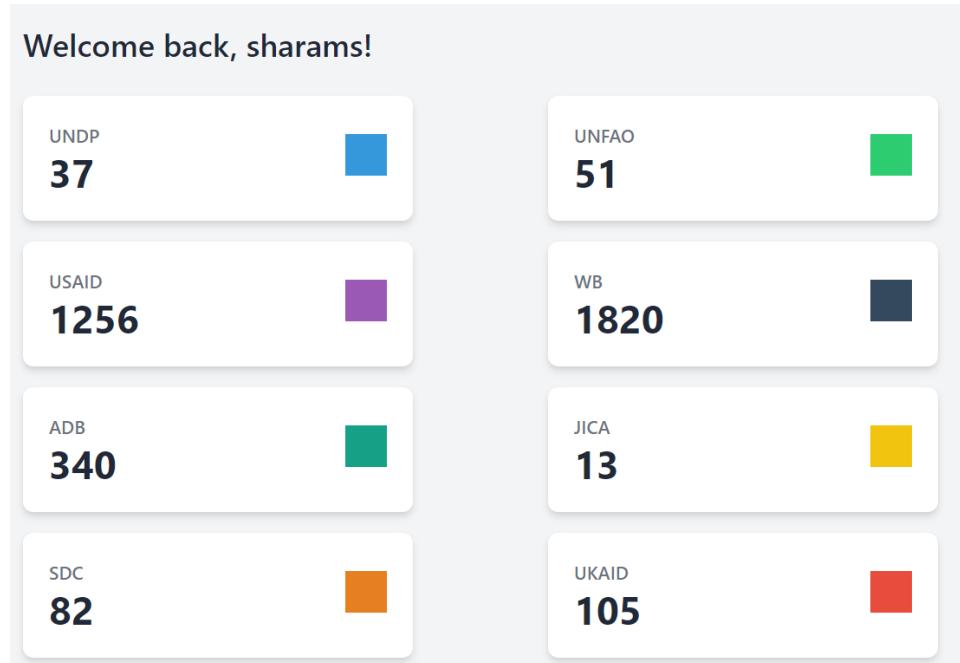


Figure 127 Results Displayed in Dashboard

Test Case 13	
Objective	To test whether the projects displayed in filter via predefined keywords are sorted according to dates.
Action	Navigate to respective page and review displayed results.
Expected Test Result	Recent projects shall be displayed at first, sorted according to dates.
Actual Test Result	Recent projects were displayed at first, sorted according to dates.
Conclusion	Test Successful.

Table 20 Unit Testing Test Case 13

The figure displays four project cards arranged in a 2x2 grid, illustrating the results of the unit testing for Test Case 13. Each card contains the following information:

- Name of the Project:** TCP/RAS/3618: strengthening the capacity for monitoring Food Security and Sustainable Agriculture in the context of Sustainable Development Goals (SDG) 2 and 12
- Duration of the Project:** 01 Jan 2018 - 30 Sep 2019 (highlighted with a red box)
- Link to the project:** [View](#)
- Organization:** UNFAO
- Save Project** button

A red arrow points from the first card to the second, indicating the sorting order. The second card shows:

- Name of the Project:** GCP/RAS/294/ASB - Regional Capacity Development through Regional Cooperation on Food Security through Control of Transboundary Animal Diseases in South Asia
- Duration of the Project:** 2015 - 2017 (Project completed) (highlighted with a red box)
- Link to the project:** [View](#)
- Organization:** UNFAO
- Save Project** button

A red arrow points from the second card to the third, indicating the sorting order. The third card shows:

- Name of the Project:** TCP/RAS/3409 - Building statistical capacity for quality food security and nutrition information in support of better informed policies
- Duration of the Project:** 2013 - 2015 (Project completed) (highlighted with a red box)
- Link to the project:** [View](#)
- Organization:** UNFAO
- Save Project** button

A red arrow points from the third card to the fourth, indicating the sorting order. The fourth card shows:

- Name of the Project:** GCP/GLO/416/UK - Integrated Food Security Phase Classification (IPC) Chronic Roll-out Project in Nepal
- Duration of the Project:** 2014 - 2015 (Project completed) (highlighted with a red box)
- Link to the project:** [View](#)
- Organization:** UNFAO
- Save Project** button

Figure 128 Projects Sorted via Date

Test Case 14	
Objective	To test whether the buttons correctly toggle in the pages where project data are displayed.
Action	Click on toggle buttons in respective pages.
Expected Test Result	Project information shall be displayed based on buttons clicked.
Actual Test Result	Project information was displayed based on buttons clicked.
Conclusion	Test Successful.

Table 21 Unit Testing Test Case 14

UNDP UNFAO USAID JICA ADB WB UKAID SDC

JICA Projects

Japan's Official Development Assistance (ODA): Rolling Plan for Nepal (PDF)

[View Project](#) [Save Project](#)

(JPP) Follow-up project on JICA Partnership Program on Promoting Quality Education through Community based School Management (Save the Children Follow-up Project/Japan)

[View Project](#) [Save Project](#)

Figure 129 JICA project information being displayed upon clicking JICA button.

The screenshot shows a grid of buttons at the top representing various categories: Food, Food Aid, Food system, Food security, Agriculture, Digital, Food Science, Food Loss, Education, Migration, Nutrition, Health, Climate Change, Livelihood, Disaster Risk Reduction, Web Application, and Mobile Application. The 'Migration' button is highlighted. Below this, a section titled 'Showing Results For Migration' displays two projects. The first project's details are: Name of the Project: International Organization for Migration (IOM) - Support: Voluntary Contribution to Fund Certain Activities Related to Processing Individuals Requesting Refugee Status and Resettlement in the United States. Its duration is listed as 'Duration of the Project:', sector as 'Emergency Response', and organization as 'USAID'. A green 'Save Project' button is present. The second project's details are: Name of the Project: International Organization for Migration to Fund A Tuberculosis (Tb) Program Benefitting Bhutanese Refugees in Nepal. Its duration is listed as 'Duration of the Project:', sector as 'Emergency Response', and organization as 'USAID'. A green 'Save Project' button is also present.

Food Food Aid Food system Food security Agriculture Digital

Food Science Food Loss Education Migration Nutrition Health

Climate Change Livelihood Disaster Risk Reduction Web Application

Mobile Application

Showing Results For Migration

Name of the Project: International Organization for Migration (IOM) - Support: Voluntary Contribution to Fund Certain Activities Related to Processing Individuals Requesting Refugee Status and Resettlement in the United States

Duration of the Project:

Sector: Emergency Response

Organization: USAID

Name of the Project: International Organization for Migration to Fund A Tuberculosis (Tb) Program Benefitting Bhutanese Refugees in Nepal

Duration of the Project:

Sector: Emergency Response

Organization: USAID

Save Project

Figure 130 Project information related to Migration being displayed upon clicking Migration button.

Test Case 15	
Objective	To test accuracy of data filtered based on predefined keywords.
Action	Click on buttons in respective page and review the displayed results.
Expected Test Result	Project information directly mentioning respective keyword shall be displayed under main results and related results shall be displayed under another section.
Actual Test Result	Project information directly mentioning respective keyword was displayed under main results and related results was displayed under another section.
Conclusion	Test Successful.

Table 22 Unit Testing Test Case 15

Showing Results For Food

The image displays four cards, each representing a project result for the keyword 'Food'. Each card contains the following information:

- Name of the Project:** Food and Nutrition Security Enhancement Project - P164319
- Link to the project:** [View](#)
- Procurement Details:** Hiring of individual Procurement Specialist for short term (intermittent basis)
- Link to the Procurement:** [View](#)
- Date of Publication:** March 17, 2024
- Organization:** WB
- Save Project** button

The four cards show slightly different procurement details, indicating they are filtered results for the keyword 'Food'.

Figure 131 Accurately showing Filtered Projects based on Keyword.

Related Results For Food

Name of the Project: GCP /GLO/505/ROK:
Implementation of Codex standards to support containment and reduction of foodborne AMR

Duration of the Project: 20-Jul-2021 - 19-Jun-2026

Link to the project: [View](#)

Organization: UNFAO

Save Project

Figure 132 Accurately Displaying Related Results for selected keyword.

Test Case 16	
Objective	To test accuracy of data filtered based on organizations.
Action	Click on buttons in respective page and review the displayed results.
Expected Test Result	Project information related to respective organizations shall be displayed upon clicking respective buttons.
Actual Test Result	Project information related to respective organizations was displayed upon clicking respective buttons.
Conclusion	Test Successful.

Table 23 Unit Testing Test Case 16

Name of the Project, Duration of the Project, Link to the project
UNJP/NEP/078/UNJ: Accelerating Progress towards Rural Women's Economic Empowerment (JP RWEE) Phase II, Duration: May 2020 to May 2027, [View Project](#)
GCP /NEP/076/GCF – Building a Resilient Churia Region in Nepal (BRCRN), May 2020 to May 2027, [View Project](#)

Figure 133 UNFAO projects.

The screenshot shows a user interface for managing UNFAO projects. At the top, there is a horizontal navigation bar with buttons for UNDP, UNFAO, USAID, JICA, ADB, WB, UKAID, and SDC. Below this, the title "UNFAO Projects" is centered. Two project cards are displayed side-by-side:

- UNJP/NEP/078/UNJ: Accelerating Progress towards Rural Women's Economic Empowerment (JP RWEE) Phase II**
Duration: 21-Jun-2022 - 24-May-2027
[View Project](#)
[Save Project](#)
- GCP /NEP/076/GCF – Building a Resilient Churia Region in Nepal (BRCRN)**
Duration: May 2020 to May 2027
[View Project](#)
[Save Project](#)

Figure 134 Projects Displayed Upon Clicking UNFAO button.

Test Case 17	
Objective	To test whether confirmation message is displayed upon clicking “save project” button.
Action	Click on save project button and review the results.
Expected Test Result	A confirmation message shall be displayed upon clicking “save project” button.
Actual Test Result	A confirmation message was displayed upon clicking “save project” button.
Conclusion	Test Successful.

Table 24 Unit Testing Test Case 17

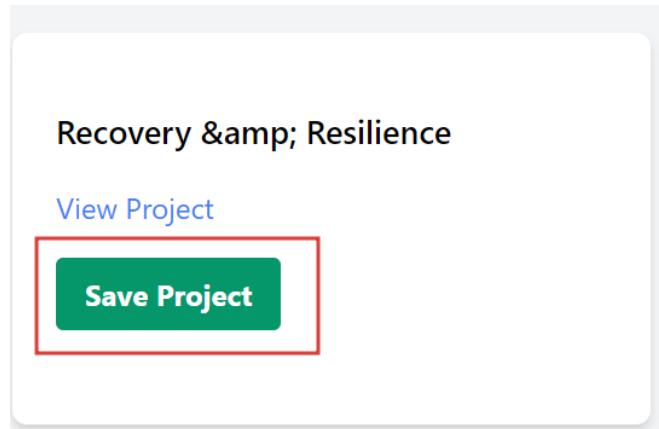


Figure 135 Clicking Save Project Button

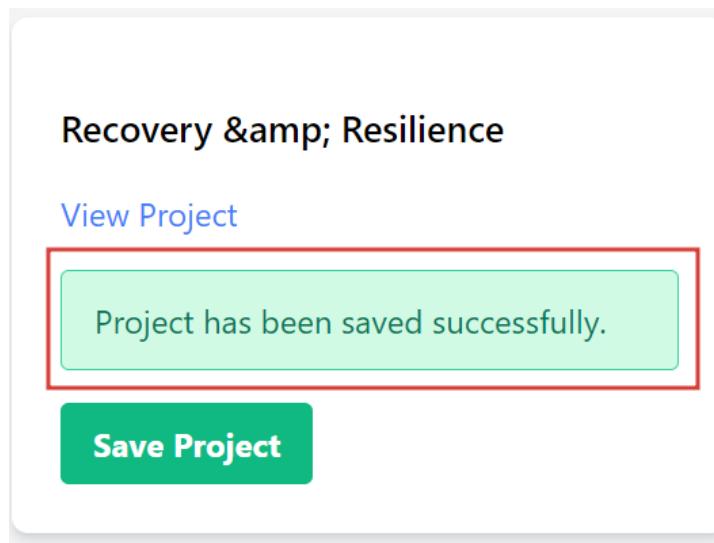


Figure 136 Confirmation Message

Test Case 18	
Objective	To test whether the user is redirected to appropriate link upon viewing project.
Action	Click on “View Project” button and review the results.
Expected Test Result	User shall be navigated to external page, displaying detailed information about projects.
Actual Test Result	User was navigated to external page, displaying detailed information about projects.
Conclusion	Test Successful.

Table 25 Unit Testing Test Case 18

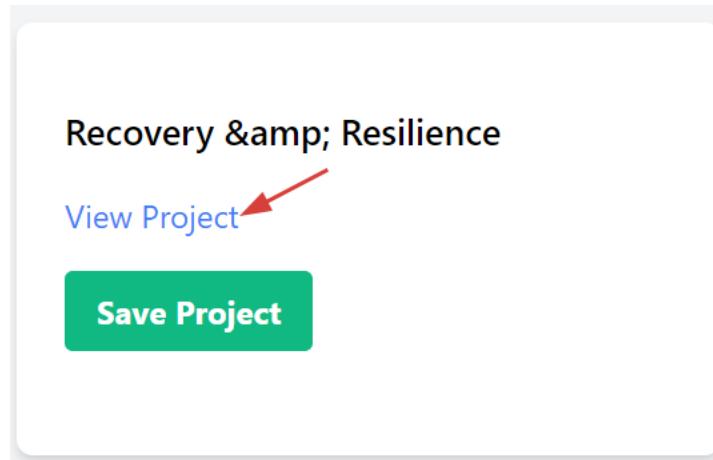
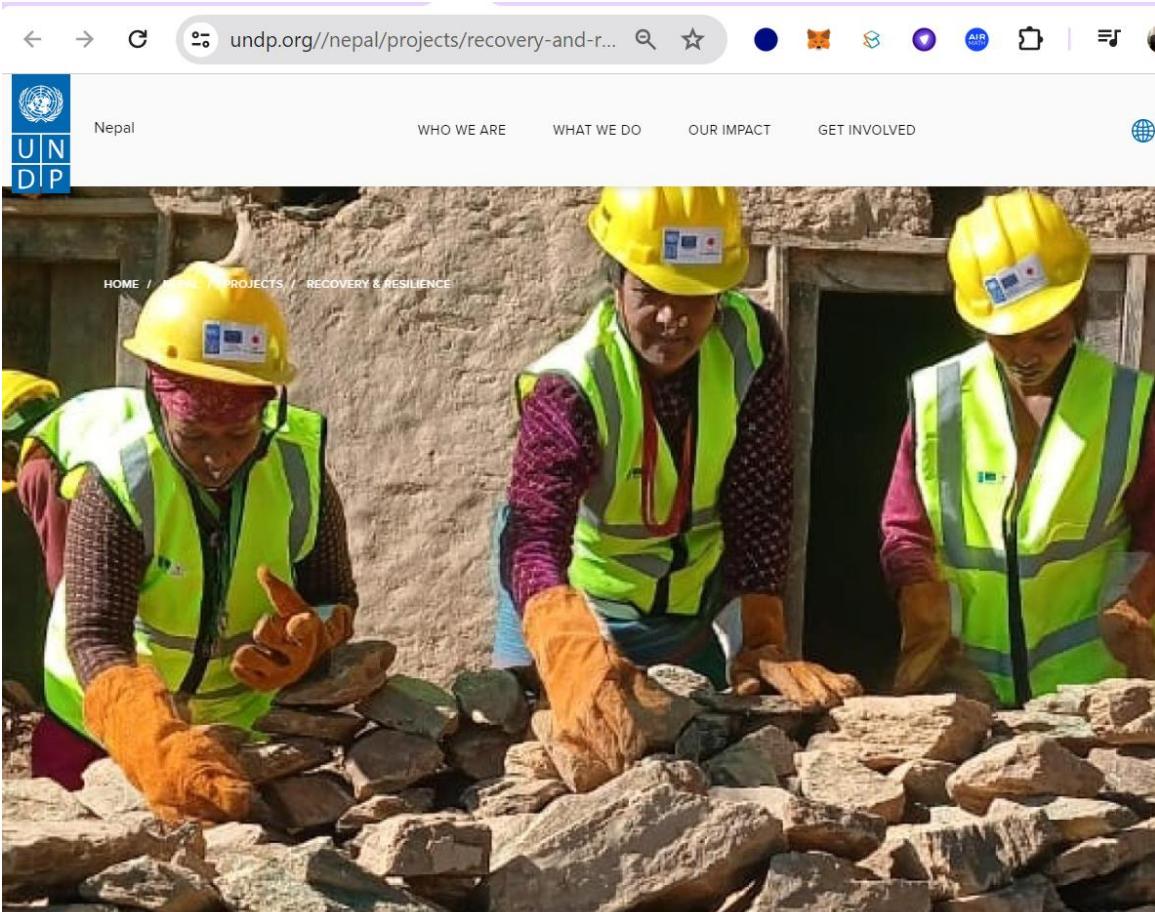


Figure 137 Clicking on "View Project" Button



Recovery and Resilience

Karnali Earthquake 2023

'On November 3, 2023, a powerful earthquake measuring 6.4 magnitude struck the western region of Nepal, causing severe damage to districts of Jajarkot, Rukum West, and Salyan in the Karnali province. The quake, followed by numerous aftershocks in Kathmandu, resulted in the tragic loss of 158 lives and injuries to over 364 people as of 27 December 2023. A significant impact was seen on 26,557 houses, rendering them completely damaged, while 35,455 houses suffered partial damage. The earthquake also disrupted 106 schools and 77 hospitals in the affected region. This has caused many people to be displaced from their homes, especially affecting women and children.'

Figure 138 Page Redirection

Test Case 19	
Objective	To test the functionality of Logout Button.
Action	Click on “Logout” button and review the results.
Expected Test Result	User shall be navigated to Login Page and shall login again to navigate webpages.
Actual Test Result	User was navigated to Login Page and needed to login again to navigate webpages.
Conclusion	Test Successful.

Table 26 Unit Testing Test Case 19

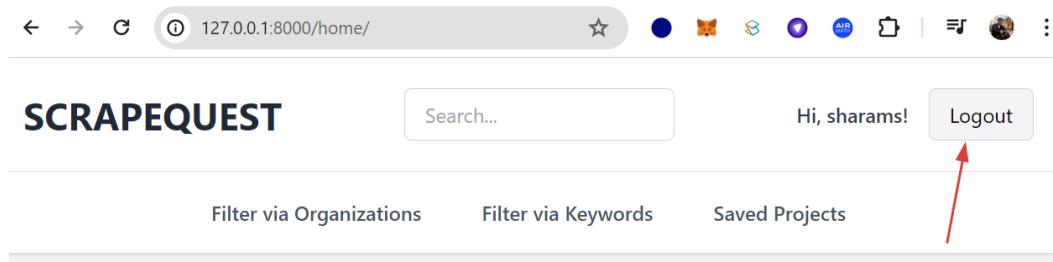


Figure 139 Clicking on Logout Button

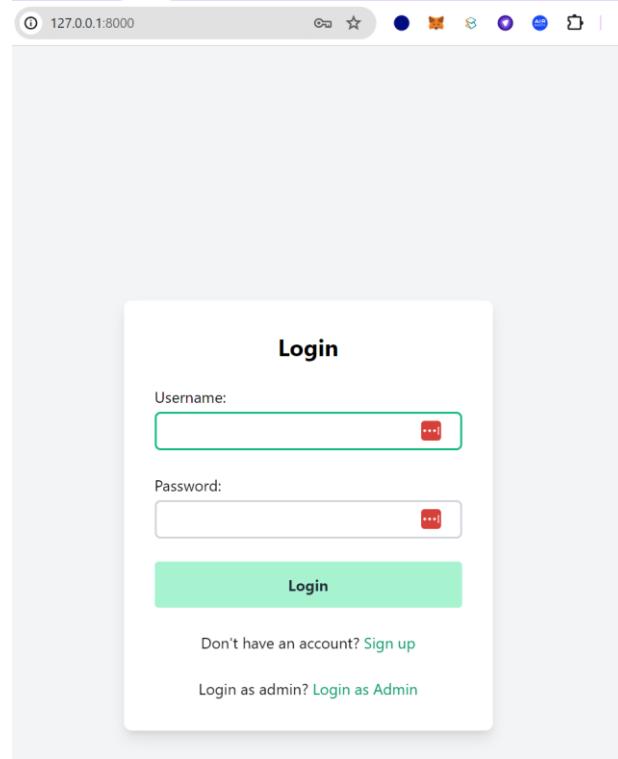


Figure 140 Navigated to Login Page

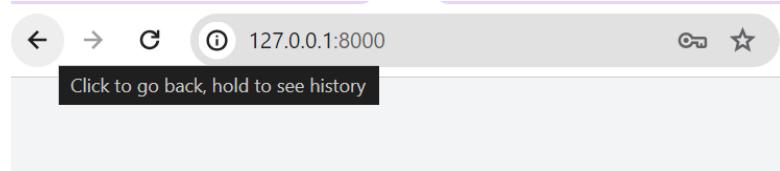


Figure 141 Navigating to Previous Page

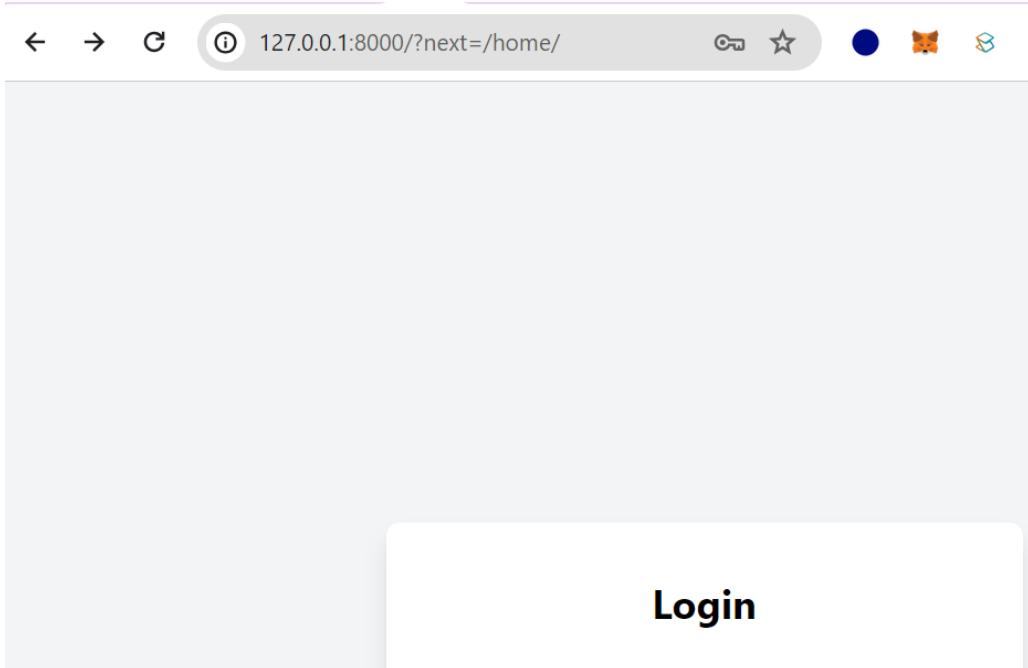


Figure 142 Prompted to Login Again

Test Case 20	
Objective	To test whether unauthorized user can login to admin panel of the system.
Action	Input valid credentials of user account with “active status” only.
Expected Test Result	User shall be displayed an error message.
Actual Test Result	User was displayed an error message.
Conclusion	Test Successful.

Table 27 Unit Testing Test Case 20

The screenshot shows the Django administration login interface. At the top, it says "Django administration". Below that, there is a red-bordered box containing the error message: "Please enter the correct username and password for a staff account. Note that both fields may be case-sensitive." Underneath the message, there is a "Username:" label followed by an input field containing "sharams". To the right of the input field is a red button with three dots (...). Below the username field is a "Password:" label followed by a red input field with a red button with three dots (...). At the bottom center is a blue "Log in" button.

Figure 143 Displayed Error Message

Test Case 21	
Objective	To test whether user can login to admin panel with incorrect credentials.
Action	Input incorrect credentials of admin account.
Expected Test Result	Admin shall be displayed an error message.
Actual Test Result	Admin was displayed an error message.
Conclusion	Test Successful.

Table 28 Unit Testing Test Case 21

The screenshot shows a Django administration login screen. At the top, a teal header bar displays the text "Django administration" next to a sun icon. Below this, a red-bordered error message box contains the text: "Please enter the correct username and password for a staff account. Note that both fields may be case-sensitive." Below the error message, there are two input fields: one for "Username" containing "admin" and another for "Password" with a red redacted placeholder. At the bottom right is a blue "Log in" button.

Figure 144 Admin was displayed an error message.

Test Case 22	
Objective	To test whether admin can delete users.
Action	Navigate to admin panel and delete users.
Expected Test Result	User shall be deleted followed up by a confirmation message.
Actual Test Result	User was deleted followed up by a confirmation message.
Conclusion	Test Successful.

Table 29 Unit Testing Test Case 22

Action: Go
1 of 3 selected Run the

<input type="checkbox"/>	USERNAME	EMAIL ADDRESS
<input type="checkbox"/>	admin	
<input type="checkbox"/>	sharams	sharamskunwar.sk@
<input checked="" type="checkbox"/>	sharams1	

3 users

Figure 145 Deleting the user.

Are you sure?

Are you sure you want to delete the selected user? All of the following objects and their related items will be deleted:

Summary

- Users: 1

Objects

- User: sharams1

Figure 146 Confirmation dialogue.



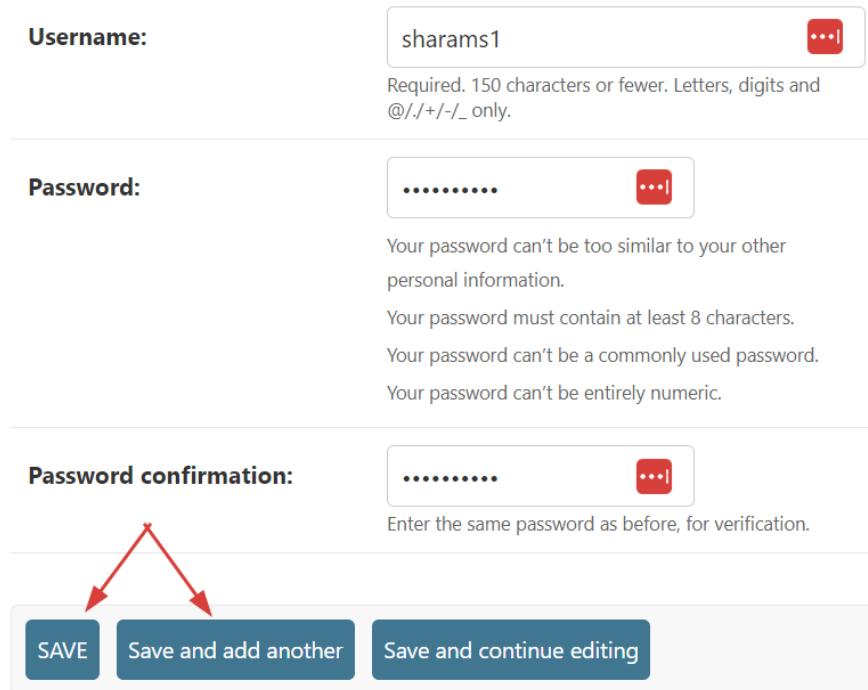
Figure 147 Confirmation message.

Test Case 23	
Objective	To test whether admin can manually add users.
Action	Navigate to admin panel and manually add user.
Expected Test Result	User shall be created followed up by a confirmation message.
Actual Test Result	User was created followed up by a confirmation message.
Conclusion	Test Successful.

Table 30 Unit Testing Test Case 23

Add user

First, enter a username and password. Then, you'll be able to edit more user options.



The screenshot shows a user addition form with the following fields:

- Username:** sharams1 (with a red info icon)
- Password:** (with a red info icon)
 - Your password can't be too similar to your other personal information.
 - Your password must contain at least 8 characters.
 - Your password can't be a commonly used password.
 - Your password can't be entirely numeric.
- Password confirmation:** (with a red info icon)
 - Enter the same password as before, for verification.
- Buttons:** SAVE, Save and add another, Save and continue editing

Figure 148 Manually Adding User

- ✓ The user "sharams1" was added successfully. You may edit it again below.

Figure 149 Confirmation Message.

Test Case 24	
Objective	To test whether admin can modify user account permissions.
Action	Navigate to admin panel and modify user account permissions. Login to admin panel again with the user account.
Expected Test Result	User account permission shall be modified followed up by a confirmation message.
Actual Test Result	User account permission was modified followed up by a confirmation message.
Conclusion	Test Successful.

Table 31 Unit Testing Test Case 24

Permissions

Active
Designates whether this user should be treated as active. Unselect this instead of deleting accounts.

Staff status
Designates whether the user can log into this admin site.

Figure 150 Modifying User Account Permissions

✓ The user "sharams1" was changed successfully.

Figure 151 Confirmation Message

Django administration

WELCOME, SHARAMS1. [VIEW SITE](#) / [CHANGE PASSWORD](#) / [LOG OUT](#) ·

Site administration

You don't have permission to view or edit anything.

Recent actions

My actions

None available

Figure 152 Login Successful

Test Case 25	
Objective	To test whether admin gets confirmation upon modifying predefined keywords.
Action	Navigate to admin panel and modify predefined keywords and review the results.
Expected Test Result	Admin shall get a confirmation message upon successful modification.
Actual Test Result	Admin gets a confirmation message upon successful modification.
Conclusion	Test Successful.

Table 32 Unit Testing Test Case 25

Edit Keywords

Keywords:

Enter keywords separated by commas

Save Changes

Figure 153 Modifying Keywords



Figure 154 Confirmation Message

Test Case 26	
Objective	To test whether visualization and keywords button change upon modifying keywords.
Action	Navigate to respective pages and review the results.
Expected Test Result	Modifications implemented shall be reflected on respective pages.
Actual Test Result	Modifications implemented were reflected on respective pages.
Conclusion	Test Successful.

Table 33 Unit Testing Test Case 26

Mobile Application Home

Showing Results For Home

Name of the Project: TCP/RAS/3509 - Regional Initiative for Zero Hunger Challenge: Promoting an Integrated Home Garden and School Garden Approach for food and nutrition security in selected Southeast Asian Countries

Duration of the Project: 03/06/2015 - 31/12/2016 (Project completed)

Link to the project: [View](#)

Organization: UNFAO

Save Project

Name of the Project: Food and Nutrition Security Enhancement Project - P164319

Link to the project: [View](#)

Procurement Details: Procurement of Poultry for Home Nutrition Garden Groups

Link to the Procurement: [View](#)

Date of Publication: November 30, 2023

Organization: WB

Save Project

Figure 155 Changes Reflected in Filter via Keywords Page

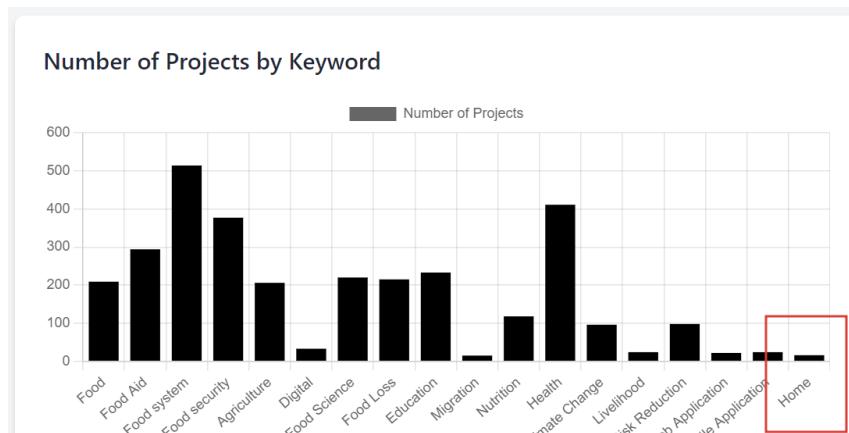


Figure 156 Changes Reflected in Dashboard Page

Test Case 27	
Objective	To test whether admin can export selected projects in respective formats.
Action	Navigate to view saved projects page and select options to export.
Expected Test Result	Project Details shall be exported to chosen format and downloaded on the local device.
Actual Test Result	Project Details were exported to chosen format and downloaded on the local device.
Conclusion	Test Successful.

Table 34 Unit Testing Test Case 27

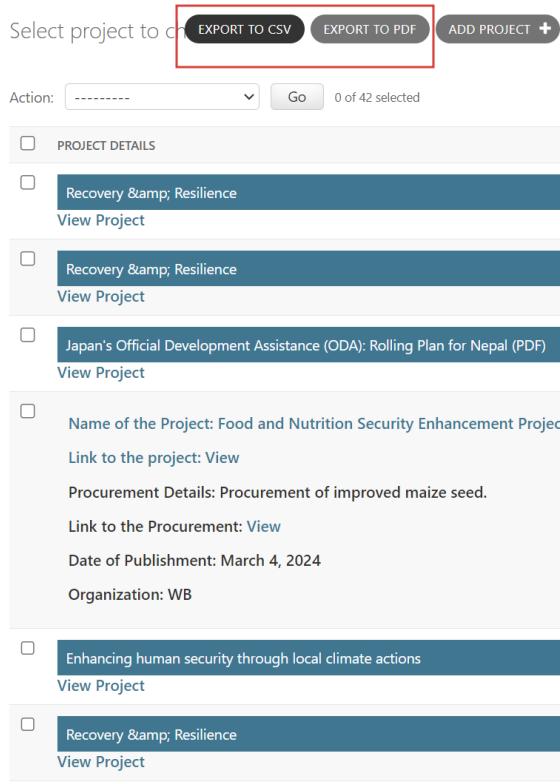


Figure 157 Buttons to Export

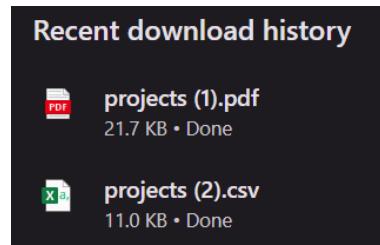


Figure 158 Downloaded Files

A	B
1 ID	Project Details
2	1 Recovery & Resilience
3	2 MCC Transmission Lines Activity
4	3
5	4 UNJP/NEP/078/UNJ: Accelerating Progress towards Rural Women`s
6	5 4109-NEP: Electricity Grid Modernization Project-Additional Financing
7	6 MiRiDew - Migrant Rights and Decent Work
8	7 2021 Population and Housing Census of Nepal
9	8 Fleming Fund â€“ Country and Regional Grants and Fellowships Programme
10	9 Illegal Wildlife Trade Challenge Fund Round 3
11	10
12	11
13	12 Darwin Initiative Round 23
14	13 MCC Road Maintenance Activity
15	14
16	15 UNJP/NEP/078/UNJ: Accelerating Progress towards Rural Women`s
17	16
18	17 Recovery & Resilience
19	18 Recovery & Resilience
20	19 Enhancing human security through local climate actions
21	20 Leaving No One Behind in Nepalâ€œ Green, Resilient, and Inclusive
22	21 Recovery & Resilience
23	22 GCP /NEP/076/GCF â€“ Building a Resilient Churia Region in Nepal (BRCRN)

Figure 159 Exported CSV

Recovery & Resilience
[View Project](#)
[View Project](#)

MCC Transmission Lines Activity
Duration:
Sector: Energy

Name of the Project: UNJP/NEP/084/UNJ Assessing the impact of the global crisis on the agriculture and food security situation in Nepal
Duration of the Project: 20-Jun-2022 - 31-Mar-2023
Link to the project:
[View](#)
[View](#)
Organization: UNFAO

UNJP/NEP/078/UNJ: Accelerating Progress towards Rural Women's Economic Empowerment (JP
RWEE) Phase II
Duration: 21-Jun-2022 - 24-May-2027
[View Project](#)
[View Project](#)

4109-NEP: Electricity Grid Modernization Project-Additional Financing
[PMD/EGMPAF/CPCUGTLP-079/80-02]
[View Project](#)
View Project
Status: Status: Closed
Description: 54107-002; Nepal; Energy; Posting date: 08 May 2023

MiRiDew - Migrant Rights and Decent Work
[View Project](#)
[View Project](#)
Project Details: Remittances sent home by over 4 million migrant workers have significantly contributed to the economic development of Nepal. The proposed project will enhance the capacities of the Government of Nepal and strengthen mechanisms to better protect the rights of workers abroad. There will be a special focus on women's need and the adverse effects of climate change exacerbating the vulnerability of migrants. The project capitalises on Switzerland's longstanding engagement on labour migration in Nepal.
Duration: 01.07.2023 - 31.12.2026

Figure 160 Exported PDF

Test Case 28	
Objective	To test whether all of the admin actions are logged.
Action	Navigate to admin panel and recent actions tab.
Expected Test Result	All of the admin action shall be logged recent actions tab.
Actual Test Result	All of the admin logs were logged under history tab.
Conclusion	Test Successful.

Table 35 Unit Testing Test Case 28

The screenshot shows a 'Recent actions' section with a header 'Recent actions'. Below it, a list titled 'My actions' displays ten entries. Each entry consists of an icon (pencil for edits, plus for additions, minus for deletions), the user name ('sharams1' or 'user'), and the action type ('User'). The entries are:

- pencil sharams1 User
- pencil sharams1 User
- + sharams1 User
- sharams1 User
- pencil sharams1 User
- + sharams1 User
- sharams1 User
- + user User
- sharams1 User
- pencil user User

Figure 161 Admin Logs

Test Case 29	
Objective	To test whether admin can login to the system as user and access all of the user functionality.
Action	Navigate to login panel and input admin credentials.
Expected Test Result	Admin shall be logged in to the system as user.
Actual Test Result	Admin was logged in to the system as user.
Conclusion	Test Successful.

Table 36 Unit Testing Test Case 29

SCRAPEQUEST

Search...

Hi, admin!

Logout

Filter via Organizations

Filter via Keywords

Saved Projects

Welcome back, admin!

Figure 162 Admin Logged in as User.

Test Case 30	
Objective	To test whether the image is successfully created from Dockerfile without any issue.
Action	Draft Dockerfile and create an image of the web app.
Expected Test Result	Image shall be created without any error.
Actual Test Result	Image was created without any error.
Conclusion	Test Successful.

Table 37 Unit Testing Test Case 30

```
sharumss@sharumss-virtual-machine:~/Desktop/Scrapequest$ sudo docker build -t scrapequest .
[sudo] password for sharumss:
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
Install the buildx component to build images with BuildKit:
https://docs.docker.com/go/buildx/

Sending build context to Docker daemon 6.344MB
Step 1/10 : FROM python:3.10-slim
--> 797a4d7093b1
Step 2/10 : ENV PYTHONDONTWRITEBYTECODE 1
--> Using cache
--> 91263c97ba41
Step 3/10 : ENV PYTHONUNBUFFERED 1
--> Using cache
--> 6bee20e070e1
Step 4/10 : ENV MONGO_URI mongodb+srv://sharams:Su9860797972@atlascluster.abktl4t.mongodb.net/
--> Using cache
--> 736e3d894334
Step 5/10 : WORKDIR /code
--> Using cache
--> 119fae8c88b1
Step 6/10 : COPY requirements.txt /code/
--> Using cache
--> 8872c1c55c4c
Step 7/10 : RUN pip install --no-cache-dir -r requirements.txt
--> Using cache
--> 221bd370c876
Step 8/10 : COPY scrapequest /code/
--> Using cache
--> 4db2e4489d92
Step 9/10 : EXPOSE 8000
--> Using cache
--> dd30d3e4ef9a
Step 10/10 : CMD ["python", "manage.py", "runserver", "0.0.0.0:8000"]
--> Using cache
--> 06023190be01
Successfully built 06023190be01
Successfully tagged scrapequest:latest
```

Figure 163 Creating Docker Image

Test Case 31	
Objective	To test whether the created image is successfully run.
Action	Run the created image within the system.
Expected Test Result	Image shall be run without any error.
Actual Test Result	Image ran without any error.
Conclusion	Test Successful.

Table 38 Unit Testing Test Case 31

```
sharumss@sharumss-virtual-machine:~/Desktop/Scrapequest$ sudo docker run scrapequest:latest
Performing system checks...
System check identified some issues:
WARNINGS:
?: (urls.W005) URL namespace 'admin' isn't unique. You may not be able to reverse all URLs in this namespace
System check identified 1 issue (0 silenced).
April 20, 2024 - 11:23:48
Django version 5.0.4, using settings 'Scrapequest.settings'
Starting development server at http://0.0.0.0:8000/
Quit the server with CONTROL-C.
```

Figure 164 Image ran without any error.

Test Case 32	
Objective	To test whether the image is successfully exported to .tar file.
Action	Export the created image in .tar file.
Expected Test Result	Image shall be exported without any error.
Actual Test Result	Image was exported without any error.
Conclusion	Test Successful.

Table 39 Unit Testing Test Case 32

```
sharumss@sharumss-virtual-machine:~/Desktop/Scrapequest$ sudo docker save -o /home/sharumss/Desktop/Scrapequest/scrapequest.tar scrapequest:latest
sharumss@sharumss-virtual-machine:~/Desktop/Scrapequest$
```

Figure 165 Export Successful



Figure 166 Exported file.

Test Case 33	
Objective	To test whether the image can be loaded into another system.
Action	Load the exported .tar file into another system.
Expected Test Result	Image shall be loaded without any error.
Actual Test Result	Image was loaded without any error.
Conclusion	Test Successful.

Table 40 Unit Testing Test Case 33

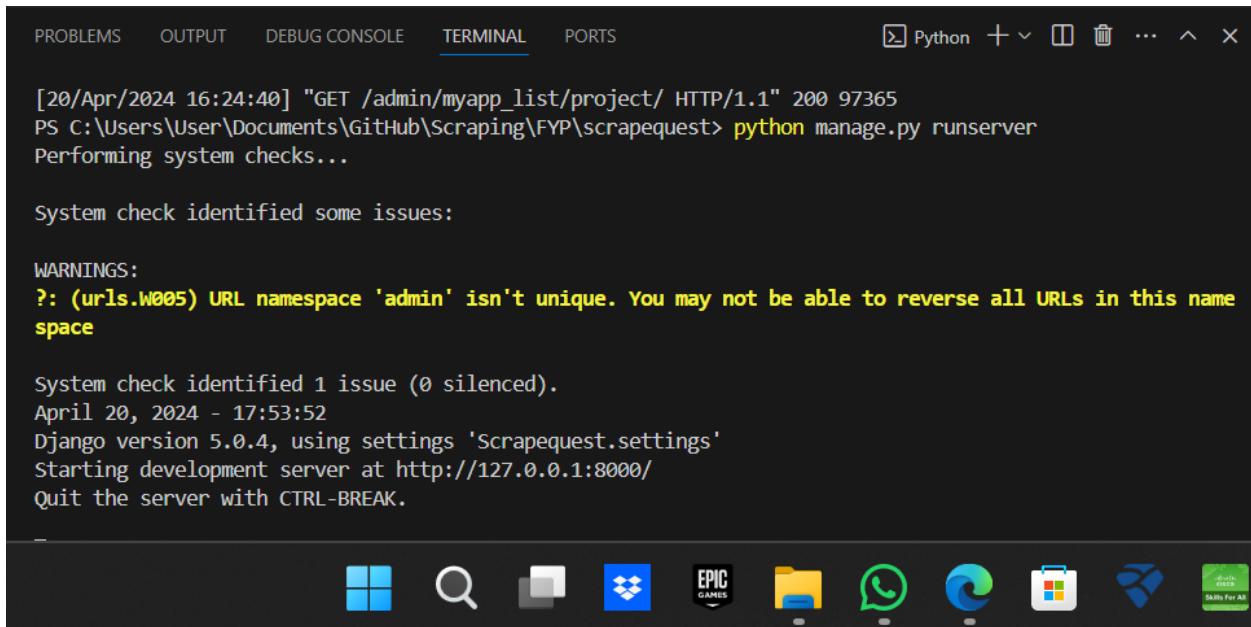
```
sharumss@sharumss-virtual-machine:~/Desktop/docker$ sudo docker load < scrapequest.tar
1f00ff201478: Loading layer [=====] 77.83MB/77.83MB
bfc9081d1eb2: Loading layer [=====] 9.552MB/9.552MB
d9ac86592bf9: Loading layer [=====] 32.92MB/32.92MB
55b572ec4bd1: Loading layer [=====] 5.12kB/5.12kB
438781090b57: Loading layer [=====] 12.88MB/12.88MB
1bc01cff5374: Loading layer [=====] 1.536kB/1.536kB
fdf24c222fad: Loading layer [=====] 2.56kB/2.56kB
6040bdf6a7a4: Loading layer [=====] 240.5MB/240.5MB
33eee93cf9d3: Loading layer [=====] 6.341MB/6.341MB
Loaded image: scrapequest:latest
sharumss@sharumss-virtual-machine:~/Desktop/docker$ sudo docker image ls
REPOSITORY          TAG      IMAGE ID   CREATED    SIZE
scrapequest         latest   06023190be01  2 days ago  363MB
```

Figure 167 Loading Docker Image into another system.

4.3 System Testing

Test Case 1	
Objective	To run Scrapequest system in Windows Platform.
Action	Run Django Server.
Expected Test Result	Scrapequest shall be run in Windows platform, every functionality shall work within the app.
Actual Test Result	Scrapequest system ran in Windows platform, and every functionality within the app worked.
Conclusion	Test Successful.

Table 41 System Testing Test Case 1



```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
[20/Apr/2024 16:24:40] "GET /admin/myapp_list/project/ HTTP/1.1" 200 97365
PS C:\Users\User\Documents\GitHub\Scraping\FYP\scrapequest> python manage.py runserver
Performing system checks...

System check identified some issues:

WARNINGS:
?: (urls.W005) URL namespace 'admin' isn't unique. You may not be able to reverse all URLs in this name space

System check identified 1 issue (0 silenced).
April 20, 2024 - 17:53:52
Django version 5.0.4, using settings 'Scrapequest.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.

```

The screenshot shows a terminal window with the following details:

- Header tabs: PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (selected), PORTS.
- Toolbar icons: Python, +, □, 🗑, ..., ^, X.
- Output content:
 - Log entry: [20/Apr/2024 16:24:40] "GET /admin/myapp_list/project/ HTTP/1.1" 200 97365
 - Command: PS C:\Users\User\Documents\GitHub\Scraping\FYP\scrapequest> python manage.py runserver
 - Message: Performing system checks...
 - Warning: System check identified some issues:
 - Warning details: WARNINGS: ?: (urls.W005) URL namespace 'admin' isn't unique. You may not be able to reverse all URLs in this name space
 - Summary: System check identified 1 issue (0 silenced).
 - Date and time: April 20, 2024 - 17:53:52
 - Django version: Django version 5.0.4, using settings 'Scrapequest.settings'
 - Server start: Starting development server at http://127.0.0.1:8000/
 - Exit command: Quit the server with CTRL-BREAK.
- Taskbar icons: File Explorer, Search, Task View, Microsoft Edge, File Explorer, WhatsApp, Microsoft Edge, File Explorer, Microsoft Word, Microsoft Teams, Microsoft Edge.

Figure 168 Windows Platform Running Scrapequest System

Test Case 2	
Objective	To run Scrapequest system in Linux Platform using docker image.
Action	Run Docker Image.
Expected Test Result	Scrapequest shall be run in Linux platform, every functionality shall work within the app.
Actual Test Result	Scrapequest system ran in Linux platform, and every functionality within the app worked.
Conclusion	Test Successful.

Table 42 System Testing Test Case 2

```
sharumss@sharumss-virtual-machine:~/Desktop/Scrapequest$ sudo docker run scrapequest:latest
Performing system checks...

System check identified some issues:

WARNINGs:
?: (urls.W005) URL namespace 'admin' isn't unique. You may not be able to reverse all URLs in this namespace

System check identified 1 issue (0 silenced).
April 20, 2024 - 11:23:48
Django version 5.0.4, using settings 'Scrapequest.settings'
Starting development server at http://0.0.0.0:8000/
Quit the server with CONTROL-C.
```

Figure 169 Linux Platform Running Scrapequest System

Test Case 3	
Objective	To test ability to view the analytics dashboard easily.
Action	Login to the system using valid credentials and access the home page.
Expected Test Result	User shall be navigated to home page after logging in successfully.
Actual Test Result	User was navigated to home page after logging in successfully.
Conclusion	Test Successful.

Table 43 System Testing Test Case 3

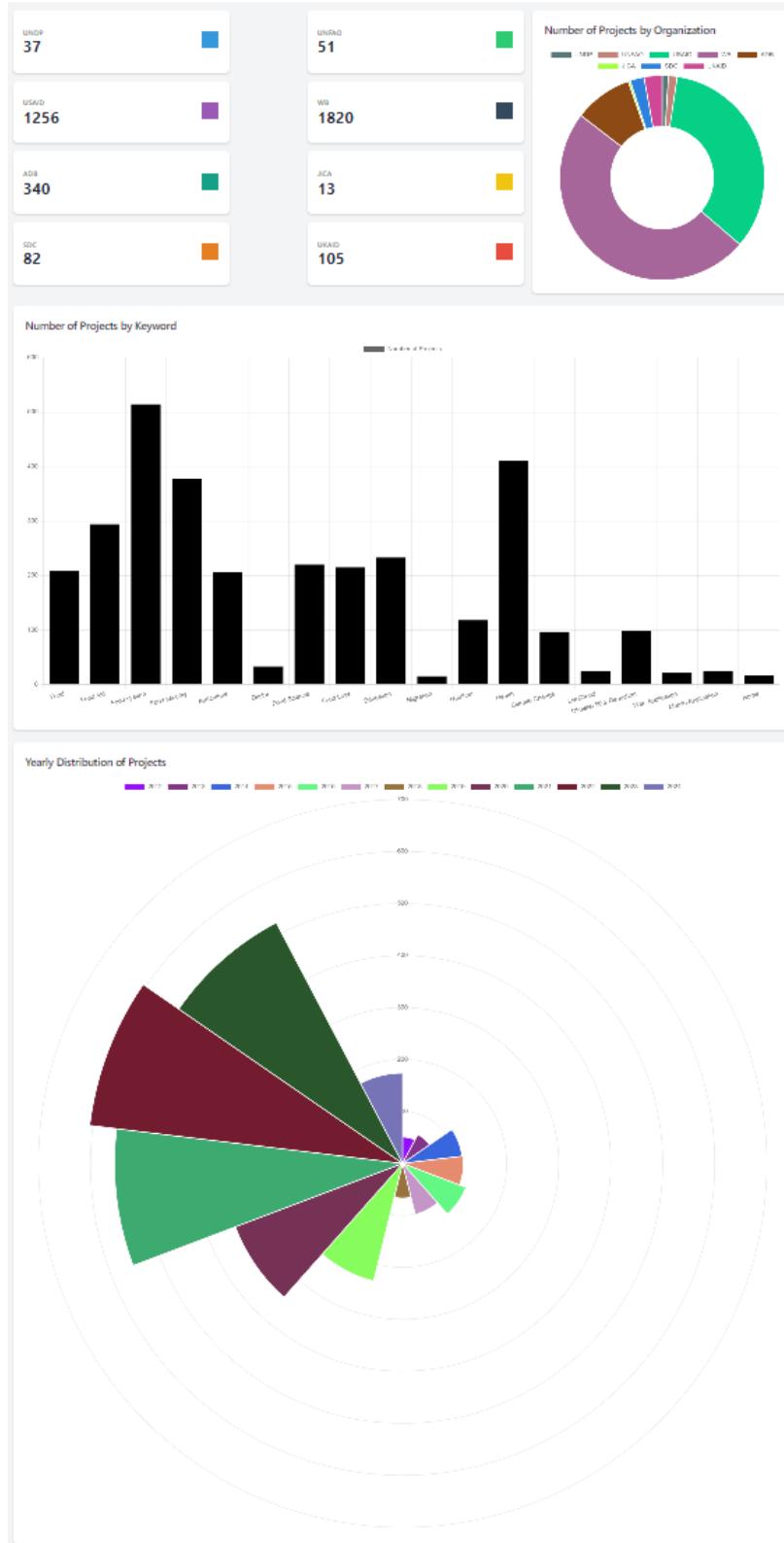


Figure 170 Displayed Analytics Dashboard

Test Case 4	
Objective	To test ability to search projects and view details accurately & easily.
Action	Login to the system and enter search query on search button in the navbar.
Expected Test Result	User shall be displayed accurate relevant results based on his search query.
Actual Test Result	User shall be displayed accurate relevant results based on his search query.
Conclusion	Test Successful.

Table 44 System Testing Test Case 4

The screenshot shows a user interface for managing international development projects. At the top, there's a search bar with the placeholder "digital". To the right, a message says "Hi, adm". Below the search bar, there are tabs for "UNDP Projects", "UNFAO Projects", and "USAID Projects". The "WB Projects" tab is currently active. A sidebar on the left lists "Organizations", "Filter via Keywords", and "Saved Projects". The main content area displays four project cards:

- UNDP Projects:** T-Digital Technical Assistance-Digital Frontiers (Project ID P154548)
- UNFAO Projects:** Digital Nepal Acceleration (DNA) Project - P176543
- USAID Projects:** Digital Nepal Acceleration (DNA) Project - P176543
- WB Projects:** Strengthening Systems for Social Protection and Civil Registration Project - P154548

Each card provides details like the name, project ID, a "View" link, procurement details, and the date of publication. Red boxes highlight specific fields: the search term "digital", the project ID "P154548" in the UNDP card, and the project ID "P176543" in both UNFAO and USAID cards.

Figure 171 Accurate and Relevant Project Details being displayed.

Test Case 5	
Objective	To test ability to view the page displaying projects filtered via organization, easily.
Action	Login to the system and click on respective button on the navbar.
Expected Test Result	User shall be displayed projects filtered via organization.
Actual Test Result	User was displayed projects filtered via organization.
Conclusion	Test Successful.

Table 45 System Testing Test Case 5

The screenshot shows the SCRAPEQUEST application's project listing page. At the top, there is a navigation bar with the title "SCRAPEQUEST", a search bar, and user authentication links ("Hi, admin!" and "Logout"). Below the navigation bar are three filter buttons: "Filter via Organizations", "Filter via Keywords", and "Saved Projects". A horizontal navigation bar below these filters contains buttons for various organizations: UNDP, UNFAO, USAID, JICA, ADB, WB, UKAID, and SDC. The main content area displays a grid of project cards. Each card represents a project and includes its name, a "View Project" link, and a "Save Project" button. The projects listed under the UNDP filter are:

- Renewable Energy for Resilient Agro-Food Systems (RERAS)
- Recovery & Resilience
- Enhancing human security through local climate actions
- European Union Support to Inclusive Federalism (EUSIF)
- Sambodhan: Temporary Basic Income
- Renewable Energy for Rural Livelihood
- Value Chain Development of Fruit and Vegetables Project (VCDP)
- Leaving No One Behind in Nepal's Green, Resilient, and Inclusive COVID-19 Recovery
- Strengthening Urban Preparedness, Earthquake Preparedness and Response in Western Regions of Nepal

Figure 172 Projects Filtered Via Organization being displayed.

Test Case 6	
Objective	To test ability to view the page displaying projects filtered via keywords, easily.
Action	Login to the system and click on respective button on the navbar.
Expected Test Result	User shall be displayed projects filtered via keywords.
Actual Test Result	User was displayed projects filtered via keywords.
Conclusion	Test Successful.

Table 46 System Testing Test Case 6

The screenshot shows the SCRAPEQUEST application interface. At the top, there is a navigation bar with a search bar, a 'Hi, admin!' message, and a 'Logout' button. Below the navigation bar are three filter buttons: 'Filter via Organizations', 'Filter via Keywords', and 'Saved Projects'. Underneath these are two rows of category buttons. The first row includes 'Food', 'Food Aid', 'Food system', 'Food security', 'Agriculture', 'Digital', 'Food Science', and 'Food Loss'. The second row includes 'Education', 'Migration', 'Nutrition', 'Health', 'Climate Change', 'Livelihood', and 'Disaster Risk Reduction'. Below these categories are three buttons: 'Web Application', 'Mobile Application', and 'Home'. The main content area has a title 'Showing Results For Food Aid'. It displays six project cards arranged in a 2x3 grid. Each card contains project details and a 'Save Project' button. The projects listed are:

- Name of the Project: Commodity Cost of Food Aid under Food For Education Program
- Duration of the Project:
- Sector: Basic Education
- Organization: USAID
- Name of the Project: ITSH Freight Cost of USAID Title II Food Aid for Other Emergency
- Duration of the Project:
- Sector: Emergency Response
- Organization: USAID
- Name of the Project: Commodity Cost of USAID Title II Food Aid for Other Emergency
- Duration of the Project:
- Sector: Emergency Response
- Organization: USAID
- Name of the Project: Commodity Cost of USAID Title II Food Aid for Sri Lank Prepositioning of Food Aid
- Duration of the Project:
- Sector: Emergency Response
- Organization: USAID
- Name of the Project: Section 202e Cost of USAID Title II Food Aid for Section 202e: Section of USAID's P.L. 480 authorizing funds to support Title II activities
- Duration of the Project:
- Sector: Emergency Response
- Organization: USAID
- Name of the Project: ITSH Freight Cost of USAID Title II Food Aid for Direct Support Cost
- Duration of the Project:
- Sector: Emergency Response
- Organization: USAID

Figure 173 Projects Filtered Via Keywords being displayed.

Test Case 7	
Objective	To test ability to view user saved projects, easily.
Action	Login to the system and click on respective button on the navbar.
Expected Test Result	User shall be displayed list of their saved projects.
Actual Test Result	User was displayed list of their saved projects.
Conclusion	Test Successful.

Table 47 System Testing Test Case 7

The screenshot shows the SCRAPEQUEST application interface. At the top, there is a navigation bar with the logo 'SCRAPEQUEST', a search bar containing 'Search...', and user account links 'Hi, admin!' and 'Logout'. Below the navigation bar, there are three filtering options: 'Filter via Organizations', 'Filter via Keywords', and 'Saved Projects'. The main content area is titled 'Saved Projects' and displays two project cards. The first project card is for 'UNJP/NEP/078/UNJ: Accelerating Progress towards Rural Women's Economic Empowerment (JP RWEE) Phase II', with a duration from '21-Jun-2022 - 24-May-2027' and a 'View Project' link. The second project card is for '4109-NEP: Electricity Grid Modernization Project-Additional Financing [PMD/EGMPAF/CPCUGTLP-079/80-02]', with a 'View Project' link, a status of 'Closed', and a description mentioning '54107-002; Nepal; Energy; Posting date: 08 May 2023'.

Figure 174 List of Admin's Saved Projects

Test Case 8	
Objective	To test admin's ability to view admin dashboard, and access exclusive features.
Action	Login to the admin panel and view dashboard.
Expected Test Result	Admin shall be displayed with the list of exclusive features to manage the system.
Actual Test Result	Admin was displayed with the list of exclusive features to manage the system.
Conclusion	Test Successful.

Table 48 System Testing Test Case 8

The screenshot shows the Django Admin interface for managing users. The top navigation bar includes links for 'WELCOME, ADMIN', 'VIEW SITE / CHANGE PASSWORD / LOG OUT', and a gear icon. Below the header, the breadcrumb navigation shows 'Home > Authentication and Authorization > Users'. On the left, a sidebar lists 'AUTHENTICATION AND AUTHORIZATION' with 'Groups' and 'Users' (which is highlighted in yellow) options, and 'MYAPP_LIST' with 'Keyword Uploads' and 'Projects' options. A search bar at the top left says 'Start typing to filter...'. The main content area is titled 'Select user to change' and contains a table with three rows. The columns are labeled 'Action', 'USERNAME', 'EMAIL ADDRESS', and 'FIRST NAME'. The rows show entries for 'admin', 'sharams' (with email 'sharamskunwar.sk@gmail.com'), and 'sharams1'. To the right of the table is a 'FILTER' sidebar with sections for 'Show counts', 'By staff status' (with 'All', 'Yes', and 'No' options), 'By superuser status' (with 'All', 'Yes', and 'No' options), and 'By active' (with 'All', 'Yes', and 'No' options). A 'ADD USER +' button is located at the top right of the main content area.

Figure 175 Admin Dashboard to Manage Users

The screenshot shows the Django admin interface for the 'Edit Keywords' page. The left sidebar lists 'Groups' and 'Users' under 'AUTHENTICATION AND AUTHORIZATION', and 'Keyword Uploads' and 'Projects' under 'MYAPP_LIST'. A search bar at the top says 'Start typing to filter...'. The main area has a title 'Edit Keywords' and a text input field containing a list of keywords separated by commas: 'Food, Food Aid, Food system, Food security, Agriculture, Digital, Food Science, Food Loss, Education, Migration, Nutrition, Health, Climate Change, Livelihood, Disaster Risk Reduction, Web Application, Mobile Application, Home'. Below the input field is a button labeled 'Save Changes'.

Figure 176 Admin Dashboard to Modify Keywords

The screenshot shows the Django admin interface for the 'Projects' list page. The left sidebar lists 'Groups' and 'Users' under 'AUTHENTICATION AND AUTHORIZATION', and 'Keyword Uploads' and 'Projects' under 'MYAPP_LIST'. A search bar at the top says 'Start typing to filter...'. The main area has a title 'Select project to change' and a table listing projects. The table includes columns for 'PROJECT DETAILS' (checkbox), 'USER' (sharams), and project details. The projects listed are: 'Recovery & Resilience' (View Project), 'Recovery & Resilience' (View Project), 'Japan's Official Development Assistance (ODA): Rolling Plan for Nepal (PDF)' (View Project), 'Name of the Project: Food and Nutrition Security Enhancement Project - P164319' (Link to the project: View, Procurement Details: Procurement of improved maize seed, Link to the Procurement: View, Date of Publication: March 4, 2024, Organization: WB), 'Enhancing human security through local climate actions' (View Project), and 'Recovery & Resilience' (View Project). Buttons at the top right include 'EXPORT TO CSV', 'EXPORT TO PDF', and 'ADD PROJECT'.

Figure 177 Admin Dashboard to View User Saved Projects and Export them.

Test Case 9	
Objective	To test whether the app supports debug mode.
Action	Navigate to gibberish path and review results.
Expected Test Result	System shall throw an error page instead of redirecting to debug mode.
Actual Test Result	System threw a 404-error page instead of redirecting to debug mode.
Conclusion	Test Successful.

Table 49 System Testing Test Case 9

← → ⌂ 127.0.0.1:8000/asadjaskdja

Not Found

The requested resource was not found on this server.

Figure 178 An Error Page

Test Case 10	
Objective	To test whether the app is responsive and cross-platform compatible.
Action	Inspect the webpage and change dimensions.
Expected Test Result	App shall be visually consistent across all screen sizes and there shall not be any layout issues upon resizing screen.
Actual Test Result	App was visually consistent across all screen sizes and there weren't any layout issues upon resizing screen.
Conclusion	Test Successful.

Table 50 System Testing Test Case 10

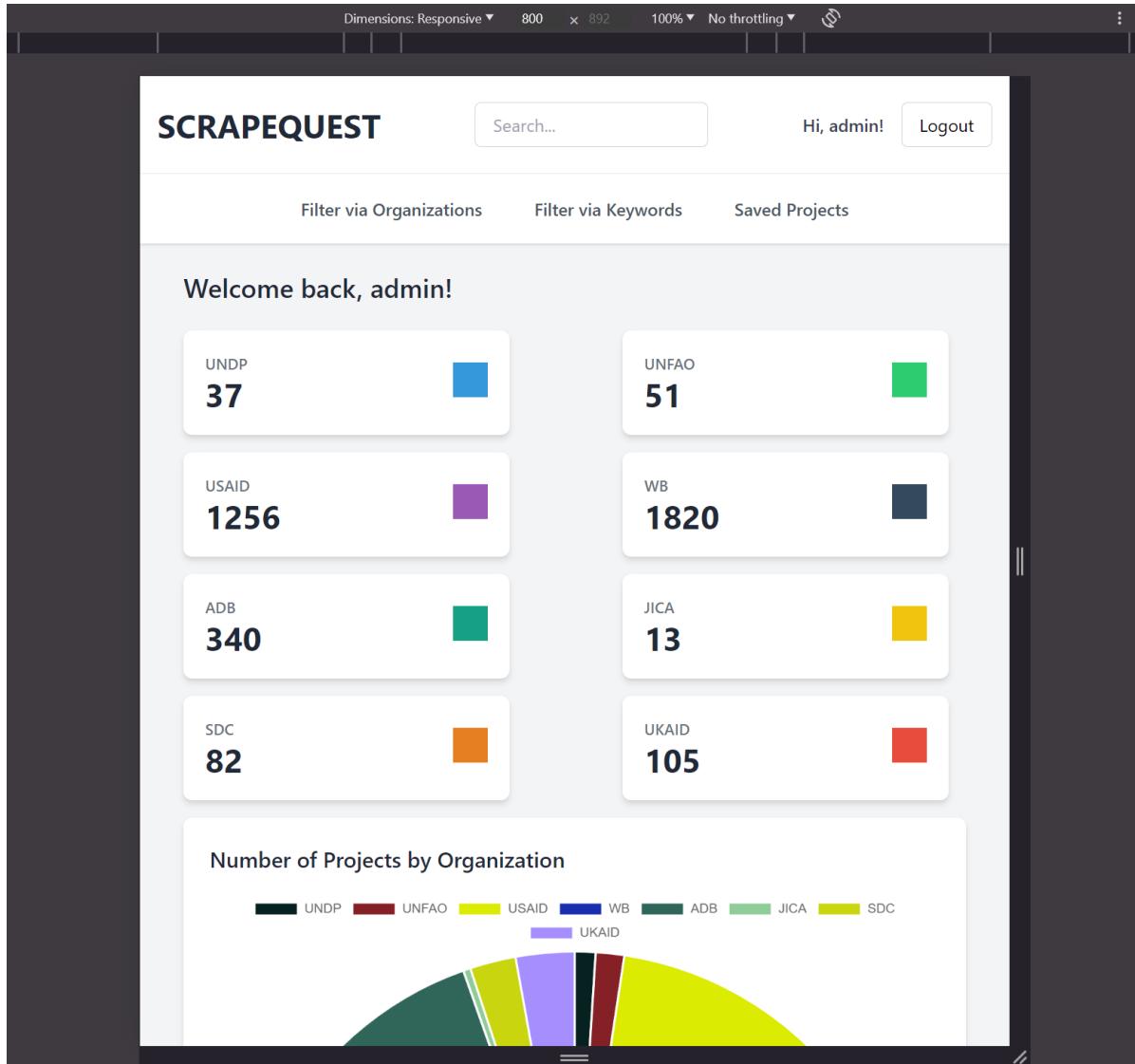


Figure 179 Responsiveness of the App

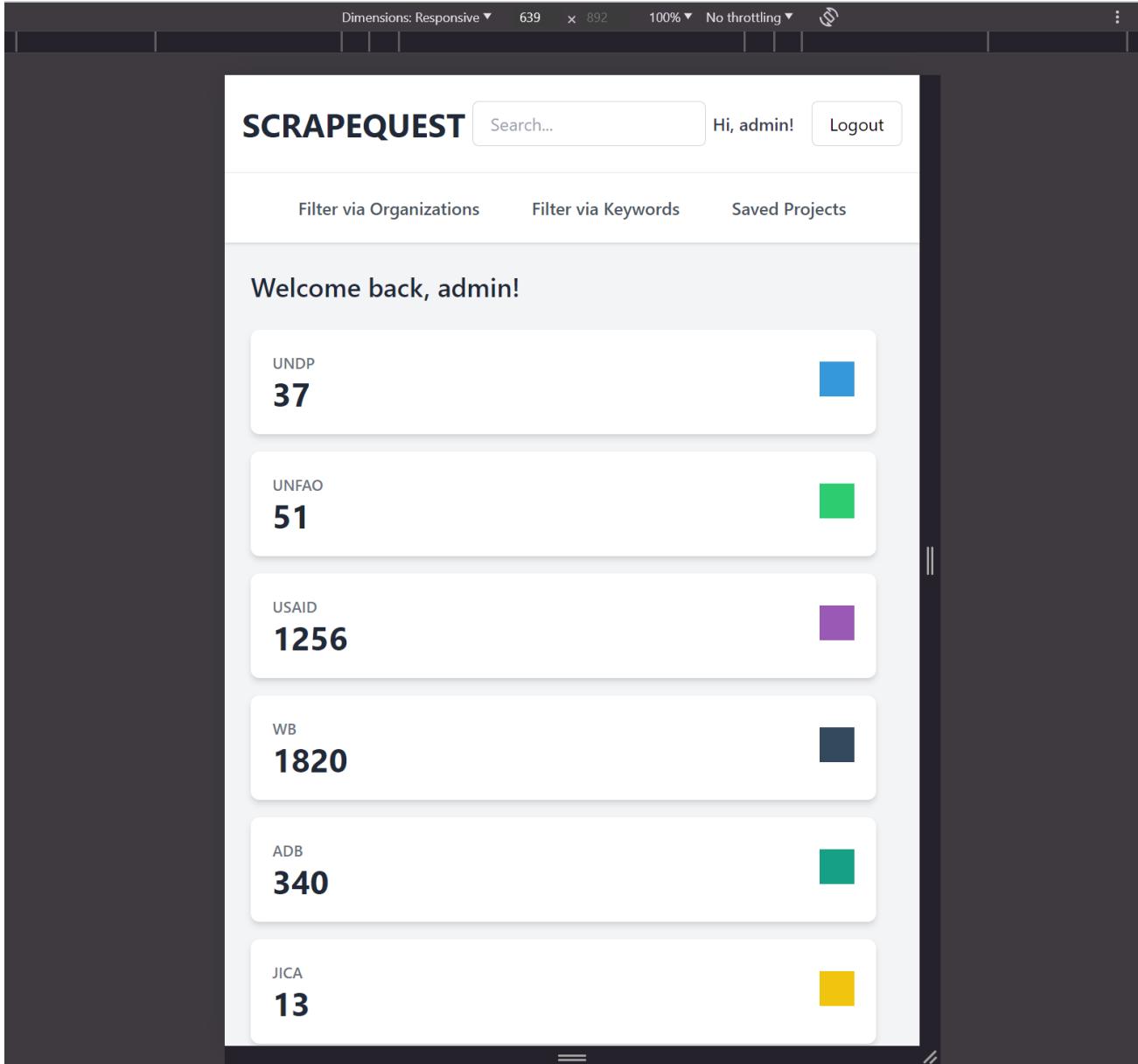


Figure 180 Responsiveness of the App

4.4 Critical Analysis

4.4.1 Test Summary

Despite encountering teething problems during the testing phases, these errors were methodically resolved while the system was being developed correspondingly, ensuring the system's functionality met the set objectives. All project components successfully underwent designated unit and system tests, verifying the system's operational status and adherence to the features proposed in the proposal. A user acceptance testing for the beta version was conducted to enable direct interaction between the client and the system. Moreover, extensive demonstrations to potential clients also effectively captured the system's capabilities and its limitations. Based on gathered user reviews, a significant limitation identified was the limited scope of project data, which, if expanded, could greatly enhance the project. Features for visualizing and filtering data received high praise, but the project data's slow loading time was viewed unfavorably.

4.4.2 Evaluation

The evaluation of the system is done by comparing its success against its intended purpose. The main purpose of this project is to create automated solutions that eliminate the need to manually go through large amounts of project data and improve the project discovery process. The system is highly efficient and accurate because it uses data scraping technologies, making it a better choice. It also integrates easily with current IT systems and requires only two commands to operate. With a proper internet connection and a computer, all aspects of the system work as planned. The system also includes simple yet effective security features and handles all potential exceptions smoothly. Further evaluations of the system are mentioned below:

4.4.2.1 Evaluation of Project Deliverables

The key outcomes of the project report include System Requirement Specifications, System Design, Practical Implementation, Test Cases with results, and a final, fully functional system. All these deliverables were successfully completed during the different stages of the project.

4.4.2.2 System Evaluation

All the client's functional requirements have been successfully met. Each specified criterion and initially promised feature has been fully accomplished. The system integrates hardware and software from multiple vendors, tailored to the client's specifications. This includes the implementation of scraping bots and a migration script, all of which function cohesively with the Scrapequest system. Additionally, the Scrapequest system has been containerized to streamline the delivery process. All project tasks and milestones were completed on schedule.

5 Conclusion

As IT organizations strive for efficiency and precision in project discovery, the development and implementation of "Scrapequest," a web scraping tool designed specifically for this purpose, marks a significant advancement. The project's methodology, primarily based on the Rational Unified Process (RUP), provided a robust framework that facilitated a detailed and adaptable approach to development, ensuring that the system not only met but exceeded the initial requirements laid out at the project's inception.

"Scrapequest" distinguishes itself through its unique capability to automate the project discovery process effectively, contrasting sharply with traditional methods that often involve cumbersome and error-prone manual searches. The system's ability to integrate and analyze data from diverse sources ensures that organizations are equipped with the most relevant and accurate information, leading to improved decision-making processes. This tool can be particularly tailored to cater to any organization with a click of a button, helping them identify projects that are not only relevant but align precisely with their strategic goals and operational capabilities.

In essence, the development of "Scrapequest" has successfully addressed the critical challenges faced by organizations in project discovery. By leveraging automated web scraping techniques, the system reduces the time and resources traditionally required for project identification and evaluation. The successful implementation of this project not only underscores the benefits of automation and advanced data management in project discovery but also sets a precedent for future innovations in this field. Looking forward, the continued refinement and adaptation of tools like Scrapequest will be vital in maintaining a competitive edge and enhancing operational efficiencies in project discovery.

5.1 Legal, Social and Ethical Issues

The section below discusses the legal, social, and ethical issues that could arise during the use of the system and how the system is equipped to handle these issues.

5.1.1 Legal Issues

Global data protection laws, like the GDPR in the European Union, require careful handling of personal data, with violations possible if data is scraped without consent. "Scrapequest" mitigates this by only accessing publicly available data and enhancing security features to anonymize and secure personal data. Additionally, web scraping may infringe on copyright, but Scrapequest checks copyright status before scraping and uses data solely for analysis, planning future implementations like citing sources to avoid infringement issues. Compliance with robots.txt and site Terms of Service is critical, as non-compliance can be seen as a breach of contract or unauthorized access, with potential legal ramifications such as "trespass to chattels" when scraping affects server performance. Scrapequest respects these guidelines and uses rate-limiting to prevent performance degradation. In Nepal, where data protection laws are evolving, staying informed and adapting to legal changes is essential for compliance. The system uses only authorized software and complies with the Electronic Transactions Act, ensuring it meets both global and local legal standards, with no illegal websites or resources accessed during project development.

[Note: Refer to [appendix](#) for detailed explanation.]

5.1.2 Social Issues

Web scraping may inadvertently collect personal data, raising privacy concerns. Scrapequest addresses this by targeting only publicly available, non-personal data and implementing data filters and security measures. Accuracy issues in web scraping can lead to misinformation, especially if used for critical decisions. Scrapequest counters this with validation algorithms to ensure data reliability and permits regular updates for maintaining data accuracy. High-frequency scraping can degrade website performance; hence, Scrapequest mimics human interactions and adheres to rate limiting to minimize server impact. The project is free, open-source, and designed to positively impact society by automating traditional manual tasks, without containing any content that could harm religious or political sensibilities or individual self-esteem, ensuring it does not breach social norms.

[Note: Refer to [appendix](#) for detailed explanation.]

5.1.3 Ethical Issues

The automation of data collection through Scrapequest can potentially displace jobs in manual data gathering and analysis. However, Scrapequest is designed to complement human workers by enhancing decision-making and fostering opportunities for innovative business practices. It employs ethical scraping practices such as rate limiting and adhering to robots.txt directives to minimize the impact on website operations. There's also a risk of dependency on automated tools for decision-making; Scrapequest addresses this by functioning as a decision-support tool requiring human verification, thus maintaining the importance of human judgment. To prevent market distortion from exclusive access to Scrapequest, the tool is made broadly accessible, with scaled solutions for various business sizes, promoting fair competition. The system is free from malware, adheres strictly to data rights and intellectual property laws, and has been developed without any harmful codes. It includes accurate citations and has been revised for content integrity, complying with all ethical standards and the regulations of London Metropolitan University.

[Note: Refer to [appendix](#) for detailed explanation.]

5.2 Advantages

The system Scrapequest, designed to automate and enhance project discovery processes for IT organizations, brings several key advantages that can significantly improve efficiency, accuracy, and strategic decision-making in corporate environments. Here are the primary advantages of the Scrapequest system:

- Aids in landing international projects which in turn can benefit the nation's economy by welcoming revenue to the country via foreign investments.
- Automates repetitive tasks of searching for project information.
- Gathers vast amounts of data quickly, enabling quick response to market changes.
- Identifies and extracts relevant data with high precision, reducing human error.
- Provides the most current information for timely decisions.
- Reduces manpower costs and minimizes opportunity costs.
- Helps identify projects aligning with strategic goals and capabilities.
- Can handle increasing data without increasing resource allocation.
- Enhances data visualization and reporting.
- Complies with legal standards and respects website integrity.
- Features a user-friendly interface and comprehensive user guides and training materials.

[Note: Explained in detail in [Appendix](#)]

5.3 Limitations

Scrapequest is just a prototype made for a college project because of which it has a considerable number of limitations. Few of the limitations are:

- **Limited Scope:**

The range of project data currently covered by Scrapequest is limited. Expanding the scope to include a broader array of projects could enhance its utility and appeal.

- **Optimization Needs:**

The system experiences slow page loading times. Optimizing the system to improve response times and efficiency is crucial for user satisfaction and system performance.

- **Adaptability to Website Changes:**

Scrapequest may struggle to adapt automatically to changes in website layouts or data presentation. Frequent updates to websites can disrupt data collection, requiring manual adjustments to scraping algorithms.

- **Overhead from Data Staleness:**

Data may become outdated or less relevant over time. The system needs mechanisms to regularly update and validate data, adding to its complexity and operational overhead.

- **Dependency on External Data Sources:**

The functionality of Scrapequest heavily relies on external data sources. Any disruptions or changes in these sources could adversely affect the system's performance and the quality of data it provides. Regular maintenance could solve this problem.

Furthermore, as it scales, there may be numerous other limitations, which have been discussed in [appendix](#).

5.4 Future Work

5.4.1 Cloud-Based Auto-Scaling

Objective: Enhance system scalability by utilizing Docker containers to manage Scrapequest deployments easily across any cloud platform for seamless scalability and better resource management.

Implementation Plan:

1. Choose a cloud provider based on cost, reliability, and available features.
2. Use cloud auto-scaling services to dynamically adjust the number of instances based on real-time load metrics.
3. Define scaling policies that trigger based on specific metrics.
4. Conduct stress testing to ensure that the system scales up and down smoothly under various load conditions.
5. Continuously monitor performance during peak loads and optimize the scaling parameters as needed to balance cost and performance.

5.4.2 Automated Compliance Checking System

Objective: Ensure compliance with international data protection regulations and website-specific terms of service.

Implementation Plan:

1. Create a rules engine that can interpret and apply the terms and conditions from various websites using (NLP) to parse and understand legal texts.
2. Connect the compliance engine with the scraping system to automatically check compliance before scraping data.
3. Keep the rules engine updated with the latest legal changes by subscribing to legal updates.
4. Implement logging and auditing mechanisms to track compliance checks and actions taken by the system.

5.4.3 Expand Project Scope

Objective: Broaden the data coverage and functionality of Scrapequest to encompass a wider array of industries.

Implementation Plan:

1. Conduct extensive market research to identify additional industries and sectors where Scrapequest's capabilities could be beneficial.
2. Design and develop industry-specific modules tailored to the needs identified in the market research phase.
3. Expand the variety of data sources from which Scrapequest can extract information.
4. Ensure that the new features and projects data integrated into Scrapequest meet the real-world needs of professionals in those industries.
5. Enhance the ability of Scrapequest to process and analyse the collected data in ways that are meaningful to new industries.
6. Thoroughly test the expanded functionalities to ensure robustness and reliability.

5.4.4 Multilingual Data Handling Capability

Objective: Enhance Scrapequest's capabilities to handle and process data from websites in multiple languages, broadening its applicability.

Implementation Plan:

1. Integrate advanced Natural Language Processing (NLP) tools to accurately scrape, interpret, and categorize content from different languages.
2. Implement automatic language detection and use translation APIs to convert non-English data into English for consistent processing and analysis.
3. Develop algorithms that not only translate words but also understand and adapt to cultural and contextual nuances important for accurate data interpretation.
4. Offer a localized user interface that supports multiple languages, enhancing usability for non-English speaking users.

5.4.5 Resource Optimization Framework

Objective: Reduce the computational and bandwidth resources required by Scrapequest, making it more efficient and cost-effective, especially at scale.

Implementation Plan:

1. Conduct a thorough analysis of current resource usage to identify bottlenecks and inefficiencies.
2. Introduce caching of web pages and data that do not change frequently to reduce redundant scrapes and decrease load times.
3. Utilize data compression techniques for data transmission and optimize the number of requests made by batching them or querying only necessary data fields.
4. Employ cloud services that offer auto-scaling and on-demand resources to efficiently manage load without over-provisioning.

5.4.6 Adaptive Scraping System for Dynamic Websites

Objective: Improve Scrapequest's ability to adapt automatically to changes in website layouts or data presentation to maintain data accuracy and scraping efficiency.

Implementation Plan:

1. Create a scraper that uses machine learning algorithms to understand and adapt to changes in HTML/CSS structures dynamically.
2. Utilize a DOM observer that detects changes in the webpage structure in real-time and adjusts scraping scripts accordingly.
3. Integrate a feedback loop where the system learns from its successes and failures, updating its parsing algorithms based on recent changes detected across websites it interacts with.
4. Regularly test the scraper on a wide range of websites to ensure robustness and adapt its performance based on feedback.

5.4.7 User Experience and System Responsiveness Enhancement

Objective: Improve the interface and responsiveness of Scrapequest to enhance user satisfaction.

Implementation Plan:

1. Redesign the front-end to create a more responsive, asynchronous user interface that can handle real-time data updates without reloading pages.
2. Convert the front-end into a Progressive Web App (PWA) to allow offline access and improve load times.
3. Optimize backend processes, including query optimization and the use of efficient data structures to speed up data retrieval and processing.

5.4.8 Data Accuracy and Verification System

Objective: Implement systems to verify the accuracy and timeliness of the data collected.

Implementation Plan:

1. Develop a feature to cross-verify the scraped data with multiple independent sources.
2. Implement a timestamp for each data entry, recording when the data was last verified or updated.
3. Allow users to report inaccuracies in the data, which can be used to prioritize data for re-verification.

6 Chapter 6: References

- Abaroa, J., 2019. *Unified Software Process*. [Online] Available at: <https://medium.com/@a01631154/unified-software-process-b288bd89c39a> [Accessed 2023].
- Abdo, O., 2023. *The Docker's Impact: Transforming the Landscape of Application Development and Deployment*. [Online] Available at: <https://blog.stackademic.com/the-dockers-impact-transforming-the-landscape-of-application-development-and-deployment-7116d4c86bee> [Accessed 2023].
- Awati, R., 2023. *requirements analysis (requirements engineering)*. [Online] Available at: <https://www.techtarget.com/searchsoftwarequality/definition/requirements-analysis> [Accessed 2023].
- Barcaroli, G. & Nurra, A., 2014. *Use of web scraping and text mining techniques in the Istat survey on "Information and Communication Technology in enterprises"*. Vienna, s.n.
- bartleby, 2023. *Variants Of Unified Process*. [Online] Available at: <https://www.bartleby.com/essay/Variants-Of-Unified-Process-PCQT83SVKU> [Accessed 2023].
- Bloch, M. & Blumberg, S., 2012. *Delivering large-scale IT projects on time, on budget, and on value*. [Online] Available at: <https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/delivering-large-scale-it-projects-on-time-on-budget-and-on-value> [Accessed 2023].
- BolpatraNepal, 2023. *One Procurement Portal Covering All Bidding Eco-System*. [Online] Available at: <https://bolpatranePAL.com/> [Accessed 2023].

BolpatraNepal, 2023. *One Procurement Portal Covering All Bidding Eco-System*. [Online] Available at: <https://bolpatranePAL.com/> [Accessed 2023].

compucademy, 2023. *UNRAVELING THE BEAUTY OF BEAUTIFULSOUP PYTHON*. [Online] Available at: <https://compucademy.net/unraveling-the-beauty-of-beautifulsoup-python/> [Accessed 2023].

Daruwalla, S., 2018. *GDPR compliance for web scrapers: The step-by-step guide*. [Online] Available at: <https://www.zyte.com/blog/web-scraping-gdpr-compliance-guide/> [Accessed 2024].

Devaraj, K., 2023. *Waterfall Model In Software Testing*. [Online] Available at: <https://testsigma.com/blog/waterfall-model-in-software-testing/> [Accessed 2023].

Dongo, I. & Cardinale, Y., 2021. A qualitative and quantitative comparison between Web scraping and API methods for Twitter credibility analysis. *International Journal of Web Information Systems*, 17(1744-0084), pp. 580-606.

geeksforgeeks, 2023. *Software Prototyping Model and Phases*. [Online] Available at: <https://www.geeksforgeeks.org/software-prototyping-model-and-phases/> [Accessed 2023].

geeksforgeeks, 2023. *Spiral Model – Software Engineering*. [Online] Available at: <https://www.geeksforgeeks.org/software-engineering-spiral-model/> [Accessed 2023].

Gladun, S., 2023. *Content may not be copied, reproduced, distributed, displayed, downloaded or otherwise used for any purpose without the prior written consent of owners*. [Online] Available at: <https://agilie.com/blog/what-is-the-software-development-life-cycle-sdlc-and-how-does-it-work> [Accessed 2023].

Gurendo, D., 2023. *Software Development Life Cycle (SDLC). Spiral Model.* [Online] Available at: <https://xbsoftware.com/blog/software-development-life-cycle-spiral-model/> [Accessed 2023].

Heller, M., 2023. *What is Visual Studio Code? Microsoft's extensible code editor.* [Online] Available at: <https://www.infoworld.com/article/3666488/what-is-visual-studio-code-microsofts-extensible-code-editor.html> [Accessed 2023].

IGleads, 2024. *Web Scraping and the CFAA: What You Need to Know.* [Online] Available at: <https://www.ig-leads.com/cfaa-regulations> [Accessed 2024].

InformationWeek, 2022. *InformationWeek State of IT - Resource Drain Study.* [Online] Available at: <https://hub.teamdynamix.com/insights/information-week-state-of-it-resource-drain-market-study> [Accessed 2023].

Kaur, N. & Himanshu, A., 2017. A Novel Semantically-Time-Referrer based Approach of Web Usage Mining for Improved Sessionization in Pre-Processing of Web Log. *International Journal of Advanced Computer Science and Applications*, 8(1).

Levin, 2022. *An Overview Of RUP Methodology Interview Questions For Assessing Candidate Skills And Experience.* [Online] Available at: <https://woodwardavenue.org/an-overview-of-rup-methodology-interview-questions-for-assessing-candidate-skills-and-experience/> [Accessed 2023].

Lutkevich, B., 2023. *waterfall model.* [Online] Available at: <https://www.techtarget.com/searchsoftwarequality/definition/waterfall-model> [Accessed 2023].

Martin, M., 2023. *Prototype Model in Software Engineering.* [Online] Available at: <https://www.guru99.com/software-engineering-prototyping-model.html> [Accessed 2023].

Minott, C., 2022. *What is RUP Methodology?*. [Online] Available at: <https://study.com/learn/lesson/rational-unified-process-methodology-examples.html> [Accessed 2023].

MongoDB, 2023. *MongoDB Features.* [Online] Available at: <https://www.mongodb.com/features#:~:text=Documents%20in%20MongoDB%20are%20stored,videos%2C%20and%20other%20binary%20data.> [Accessed 2023].

Mozilla, 2023. *Django introduction.* [Online] Available at: <https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django/Introduction> [Accessed 2023].

Munappy, A. R., Mattos, D. I. & Bosch, J., 2020. *From Ad-Hoc Data Analytics to DataOps*. Seoul Republic of Korea , Association for Computing Machinery, pp. 165-174.

Nolle, T., 2023. *data flow diagram (DFD).* [Online] Available at: <https://www.techtarget.com/searchdatamanagement/definition/data-flow-diagram-DFD> [Accessed 2023].

Osis, J. & Donins, U., 2017. *Unified process.* [Online] Available at: <https://www.sciencedirect.com/topics/computer-science/unified-process> [Accessed 2023].

PureStorage, 2024. *What Is MongoDB?* [Online] Available at: <https://www.purestorage.com/uk/knowledge/what-is-mongodb.html#:~:text=Real%2DTIme%20Analytics, and%20accessed%20on%20the%20fly.> [Accessed 2024].

Python, 2023. *What is Python? Executive Summary.* [Online] Available at: <https://www.python.org/doc/essays/blurb/> [Accessed 2023].

Riley, K. C., 2019. *Data Scraping as a Cause of Action: Limiting Use of the CFAA and Trespass in Online Copying Cases.* [Online]
 Available at: <https://core.ac.uk/download/pdf/216959269.pdf>
 [Accessed 2024].

Simajuntak, H. & Traigan, D., 2022. Weighted Hybrid Recommendation System for Toba Tourism Based on Google Review Data. *IEEE International Conference of Computer Science and Information Technology*, pp. 1-8.

teamstage, 2023. *Project Management Statistics: Trends and Common Mistakes in 2023.* [Online]

Available at: <https://teamstage.io/project-management-statistics/>
 [Accessed 2023].

TechTarget, 2023. *use case diagram (UML use case diagram).* [Online]

Available at: <https://www.techtarget.com/whatis/definition/use-case-diagram>
 [Accessed 2023].

TenderNotice, 2023. *Welcome to Tender Notice.* [Online]

Available at: <https://tendernotice.com.np/>
 [Accessed 2023].

TEPC, 2006. *The Electronic Transactions Act, 2063 (2008).* [Online]

Available at: <http://www.tepc.gov.np/uploads/files/12the-electronic-transaction-act55.pdf>
 [Accessed 2024].

TradeNep, 2023. नेपाल भरीबाट प्रकाशित सबै प्रकारका टेच्चर तथा बोलपत्र सँग सम्बन्धित सुचनाहरु ;
Tender Notice in Nepal. [Online]
 Available at: <https://tradenep.com/tender/>
 [Accessed 2023].

tryqa, 2023. *What is Waterfall model- Examples, advantages, disadvantages & when to use it?.* [Online]

Available at: <https://tryqa.com/what-is-waterfall-model-advantages-disadvantages-and-when-to->

use-it/#Disadvantages_of_waterfall_model

[Accessed 2023].

VertexSpecial, 2023. *Vertex*. [Online] Available at: <https://vertexspecial.com/> [Accessed 2023].

WebscrapeAI, 2023. *Selenium Python Web Scraping: A Comprehensive Guide*. [Online] Available at: <https://webscrapeai.com/blog/selenium-python-web-scraping-a-comprehensive-guide> [Accessed 2023].

workamajig, 2022. *Project Management Statistics: 45 Stats You Can't Ignore*. [Online] Available at: <https://www.workamajig.com/blog/project-management-statistics> [Accessed 2023].

7 Chapter 7: Bibliography

Amaral, o. & Abualhaija, S., 2022. NLP-based Automated Compliance Checking of Data Processing Agreements against GDPR. *JOURNAL OF LATEX*, 14(8), pp. 1-26.

Asgari, B. & Ghobaei-Arani, M., 2016. An Efficient Approach for Resource Auto-Scaling in Cloud Environments. *IJECE*, Volume 6, pp. 2415-2424.

Belogay, A. & Koeva, S., n.d. *Harnessing NLP Techniques in the Processes of Multilingual Content Management*. [Online]

Available at: <https://aclanthology.org/E12-2002.pdf>
[Accessed 2024].

Cai, Z. & Li, J., 2020. Research on Performance Optimization of Web Application System based on JAVA EE. *Journal of Physics Conference Series*, 1437(1).

Carle, V., 2020. *Web Scraping using Machine*. [Online]
Available at: <https://www.diva-portal.org/smash/get/diva2:1468583/FULLTEXT01.pdf>
[Accessed 2024].

IBM, 2024. *What is auto scaling?* [Online]
Available at: <https://www.ibm.com/topics/autoscaling>
[Accessed 2024].

Kozal, J. & Zyblewski, P., 2022. *Lifelong Learning Natural Language Processing Approach for Multilingual Data Classification*. [Online]

Available at:
researchgate.net/publication/361500367_Lifelong_Learning_Natural_Language_Processing_Approach_for_Multilingual_Data_Classification
[Accessed 2024].

Mogouie, Khosro ; Shamsi, Mahboubeh, 2015. A Novel Approach for Optimization Auto-Scaling in Cloud Computing Environment. *International Journal of Computer Network and Information Security*, 7(11).

Patnaik, S. K. & Baby, N., 2021. Intelligent and Adaptive Web Data Extraction System Using Convolutional and Long Short-Term Memory Deep Learning Networks. *BIG DATA MINING AND ANALYTICS*, 4(4), pp. 279-297.

Sousa, G. & Rudametkin, W., 2016. Automated Setup of Multi-Cloud Environments for Microservices Applications. *HAL*, pp. 327-334.

Vihervaara, J. & Loula, P., n.d. *Performance Gains from Web Performance Optimization*. [Online]

Available at: <https://www.scitepress.org/papers/2016/59036/59036.pdf>
[Accessed 2024].

web.dev, 2024. *Progressive Web Apps*. [Online]
Available at: <https://web.dev/explore/progressive-web-apps>
[Accessed 2024].

Zhang, J. & El-Gohary, N. M., 2015. Semantic NLP-based Information Extraction from Construction Regulatory Documents for Automated Compliance Checking. *J. Comput. Civ. Eng.*, pp. 1-42.

8 Chapter 8: Appendix

8.1 Appendix A: Pre-Survey

8.1.1 Pre-Survey Form

Pre-Survey Form - Scrapequest

B I U ↲ ✖

This survey aims to gather insights from employees of selected companies working in the field of international development projects on the current project discovery process to identify areas of improvement and to assess the need for a new tool like Scrapequest.

Tailored Solution for Your Needs



SCRAPEQUEST

Select the company you work in: *

1. Vertex Special Technologies
2. Sunaula Khimti Construction Company
3. Kshamadevi Group
4. Dhukuchu Construction Company

How involved are you in the project discovery process? *

- Not involved
- Somewhat involved
- Very involved
- I lead the process

On a scale of 1 to 5, how would you rate the current project discovery process at our company? (1 being very inefficient, 5 being very efficient) *

- 1
- 2
- 3
- 4
- 5

3. Approximately how much time do you spend weekly on project discovery-related tasks?

- Less than 5 hours
- 5 to 10 hours
- 10 to 15 hours
- More than 15 hours

*
Do you feel the time you spend on project discovery is:

- Just right
- Somewhat excessive
- Far too excessive
- Not enough

What do you find most challenging about the project discovery process? (Select all that apply) *

- Identifying projects that align with our strategy
- Sifting through large volumes of data
- Verifying the reliability of project information
- Keeping track of new and updated projects
- Other...

What tools or resources do you currently use for project discovery? (Select all that apply) *

- Manual searching through websites and databases
- Commercial project discovery databases
- Industry networking events and conferences
- Internal databases and spreadsheets
- Other...

In your opinion, what features would be most beneficial in a project discovery tool? *

- Real-time updates on new projects
- Data filtering to match projects with our strategy
- Automated verification of data reliability
- Integration with our current IT systems
- Cost-effectiveness
- User-friendly interface

How does the current project discovery process impact your daily workload and morale? *

- Positively
- No significant impact
- Somewhat negatively
- Very negatively

What improvements do you hope to see in the project discovery process within the next year? *

- Faster project identification
- Better project-strategy alignment
- Enhanced data management and filtering
- Reduction in manual tasks
- Other...

Figure 181 Pre-Survey Form

8.1.2 Sample of Filled Pre-Survey Forms

Responses cannot be edited

Pre-Survey Form - Scrapequest

This survey aims to gather insights from employees of selected companies working in the field of international development projects on the current project discovery process to identify areas of improvement and to assess the need for a new tool like Scrapequest.

* Indicates required question

Tailored Solution for Your Needs



SCRAPEQUEST

Select the company you work in: *

Vertex Special Technologies



How involved are you in the project discovery process? *

- Not involved
- Somewhat involved
- Very involved
- I lead the process

On a scale of 1 to 5, how would you rate the current project discovery process at our company? (1 * being very inefficient, 5 being very efficient)

- 1
- 2
- 3
- 4
- 5

3. Approximately how much time do you spend weekly on project discovery-related tasks?

- Less than 5 hours
- 5 to 10 hours
- 10 to 15 hours
- More than 15 hours

*
Do you feel the time you spend on project discovery is:

- Just right
- Somewhat excessive
- Far too excessive
- Not enough

What do you find most challenging about the project discovery process? (Select all that apply) *

- Identifying projects that align with our strategy
- Sifting through large volumes of data
- Verifying the reliability of project information
- Keeping track of new and updated projects
- Other: _____

What tools or resources do you currently use for project discovery? (Select all that apply) *

- Manual searching through websites and databases
- Commercial project discovery databases
- Industry networking events and conferences
- Internal databases and spreadsheets
- Other: _____

In your opinion, what features would be most beneficial in a project discovery tool? *

- Real-time updates on new projects
- Data filtering to match projects with our strategy
- Automated verification of data reliability
- Integration with our current IT systems
- Cost-effectiveness
- User-friendly interface

How does the current project discovery process impact your daily workload and morale? *

- Positively
- No significant impact
- Somewhat negatively
- Very negatively

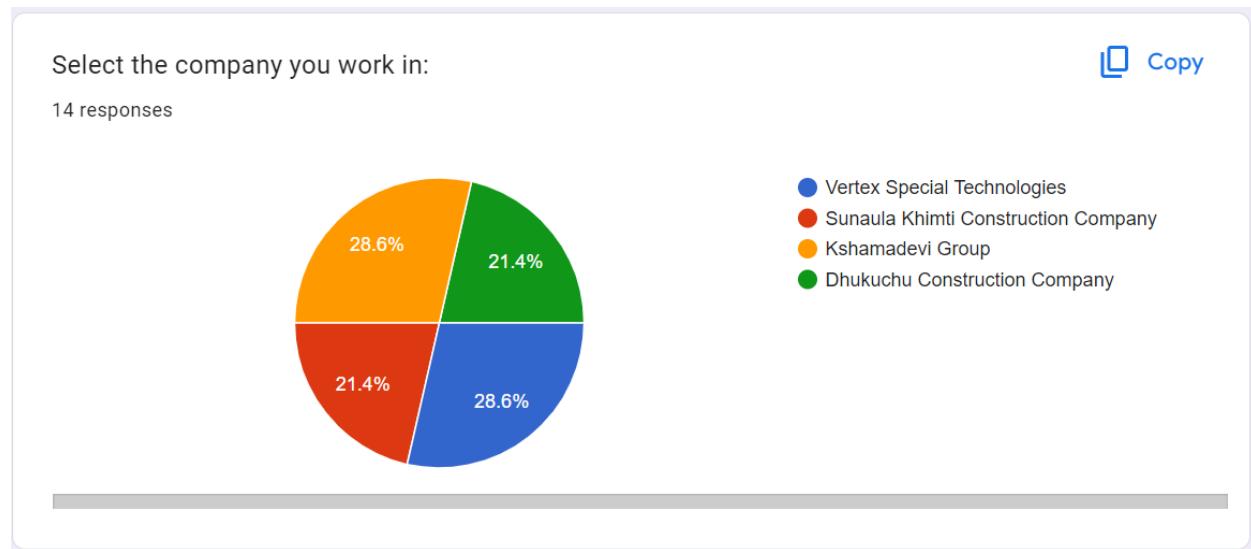
What improvements do you hope to see in the project discovery process within the next year? *

- Faster project identification
- Better project-strategy alignment
- Enhanced data management and filtering
- Reduction in manual tasks
- Other: _____

Figure 182 Sample of Filled Pre-Survey Form

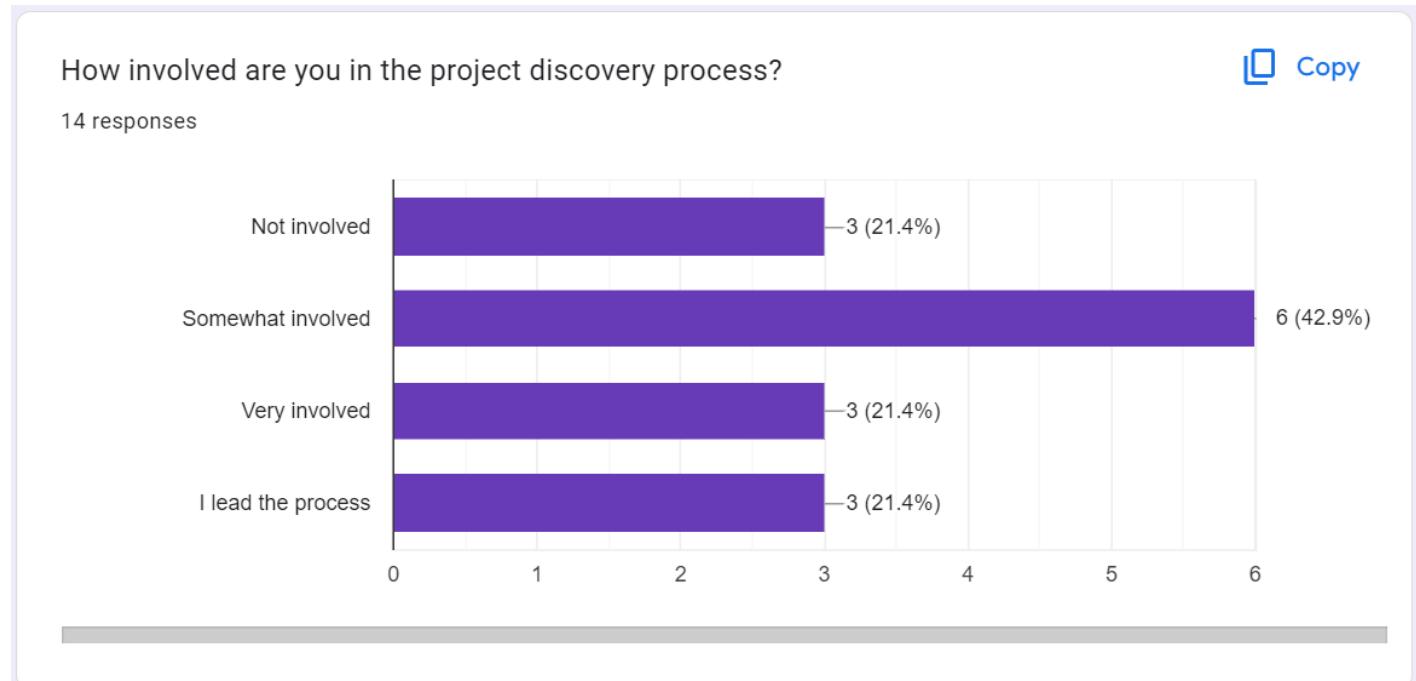
8.1.3 Pre-Survey Result

Respondent Companies:



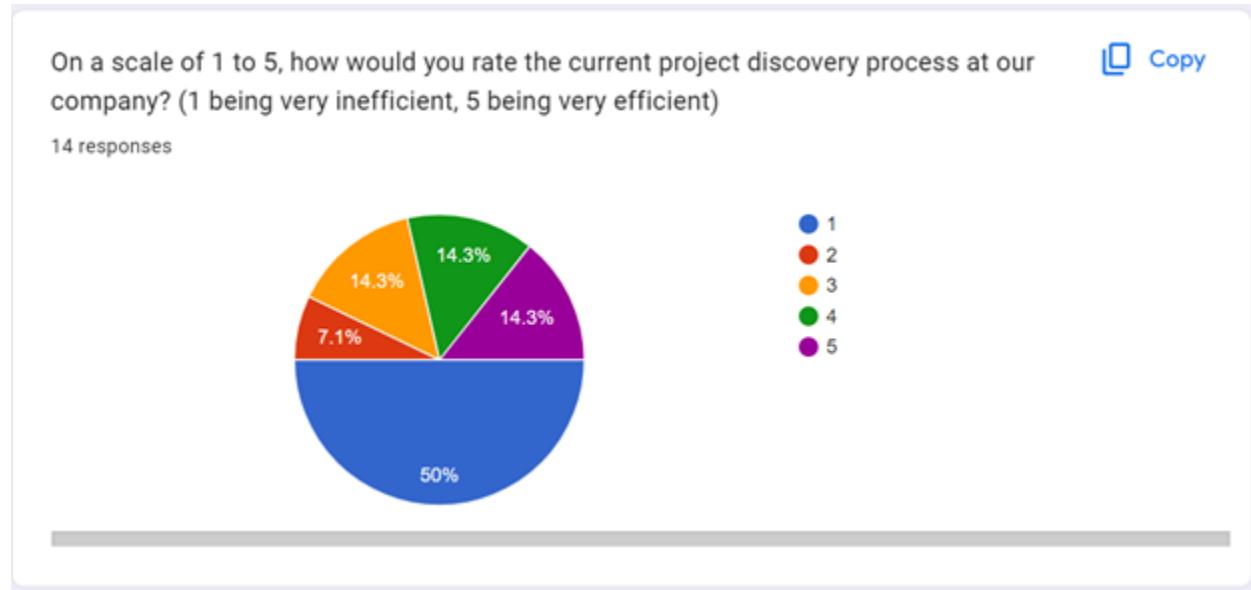
The respondents are from four different companies: Vertex Special Technologies, Sunaula Khimti Construction Company, Kshamadevi Group, and Dhukuchu Construction Company. The distribution among these companies was fairly even, suggesting a diverse group of participants.

Involvement in Project Discovery:



Employees reported varying levels of involvement in the project discovery process. Notably, 42.9% stated that they are 'somewhat involved', and an equal proportion are 'very involved' or 'lead the process', each category capturing 21.4% of the responses. This indicates a relatively even distribution across the spectrum of involvement, suggesting that the respondents of survey were in some form of managerial role.

Perceived Efficiency:



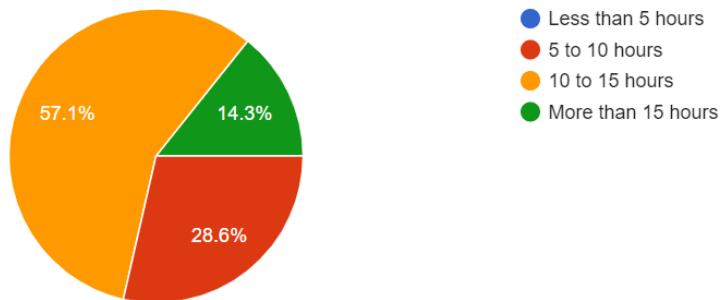
When asked to rate the efficiency of the current process on a scale from 1 (very inefficient) to 5 (very efficient), 50% of the respondents rated it as moderate (3). However, there are concerns about inefficiency as 28.6% rated it below 3. This suggests room for improvement in the process's efficiency.

Time Spent and Perception:

3. Approximately how much time do you spend weekly on project discovery-related tasks?

 Copy

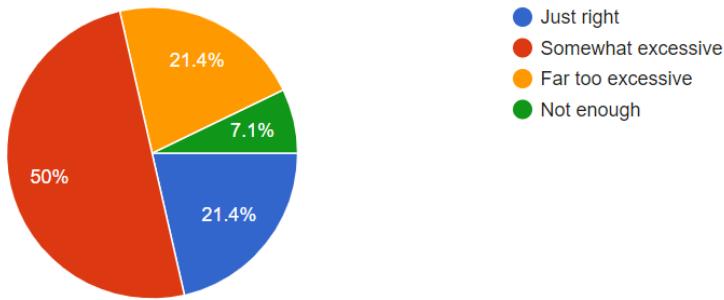
14 responses



 Copy

Do you feel the time you spend on project discovery is:

14 responses



The majority of employees (57.1%) spend 5 to 10 hours per week on project discovery-related tasks, while 28.6% spend more than 15 hours, which indicates a significant time commitment for some.

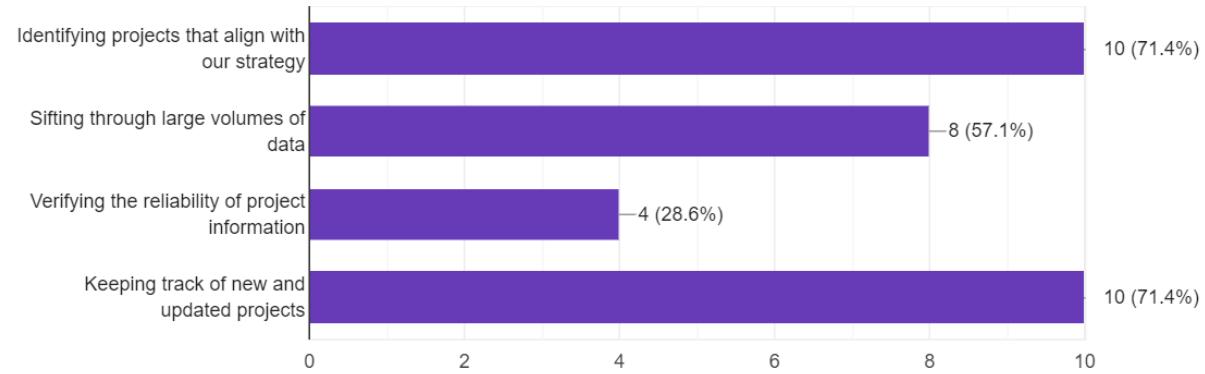
Half of the respondents feel the time spent is just right, yet 21.4% find it somewhat excessive, and another 21.4% feel it is far too excessive. This indicates a significant portion of the workforce perceives a misalignment in time allocation versus value received.

Challenges Faced:

[Copy](#)

What do you find most challenging about the project discovery process? (Select all that apply)

14 responses



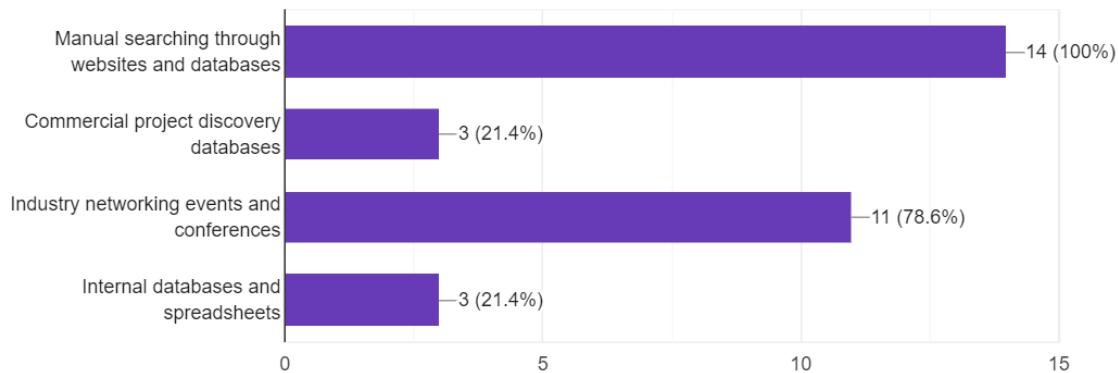
Identifying projects that align with the company's strategy and keeping track of new and updated projects were cited as major challenges by 71.4% of the participants. Additionally, sifting through large volumes of data is a significant hurdle for 57.1% of the staff.

Tools and Resources Used:

[Copy](#)

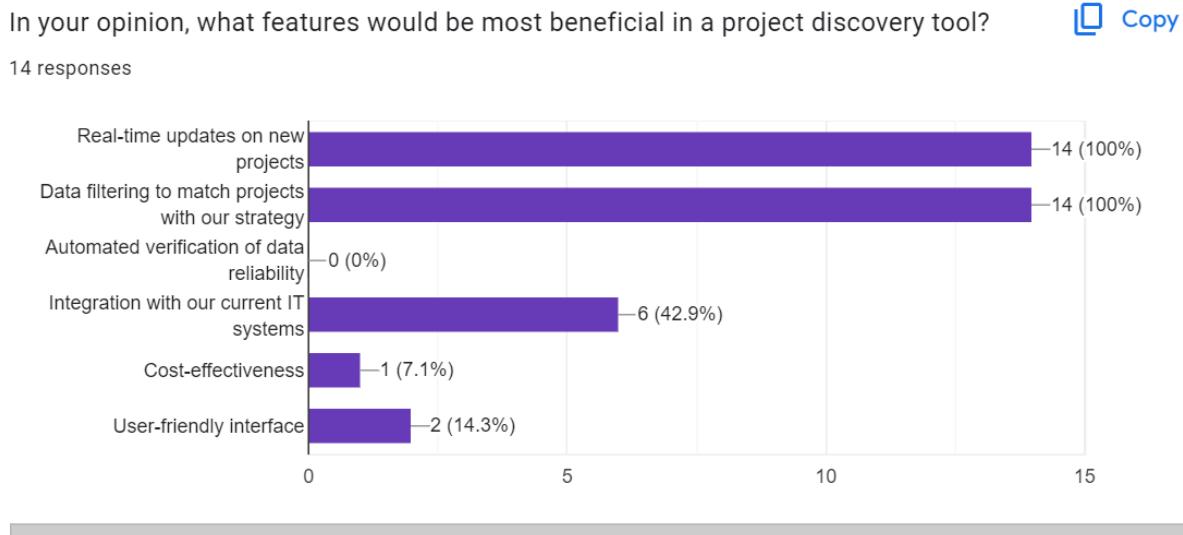
What tools or resources do you currently use for project discovery? (Select all that apply)

14 responses



All respondents use manual searching through websites and databases, indicating a high reliance on manual methods. Networking events and internal databases are also frequently used, by 78.6% and 21.4% of respondents, respectively.

Desired Features in Discovery Tools:



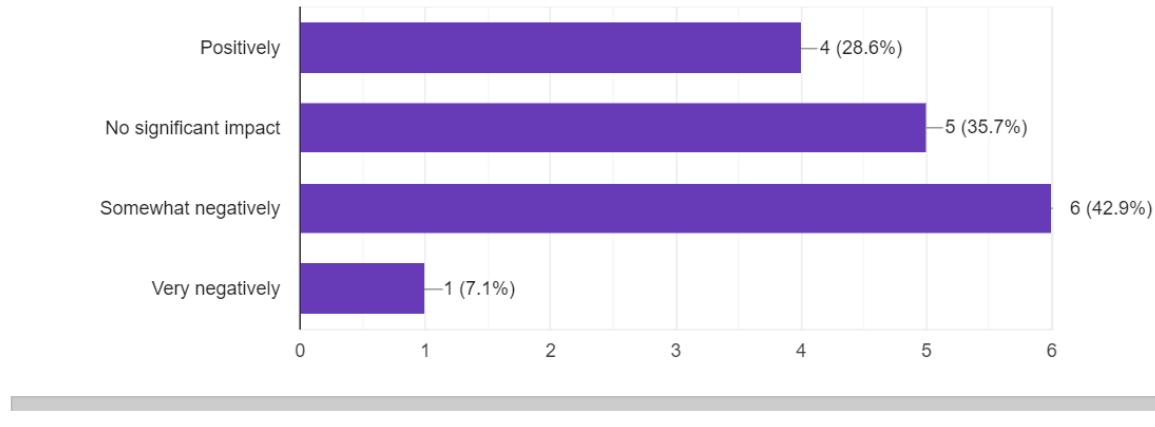
There is a unanimous demand for real-time updates on new projects and data filtering to match projects with the company's strategy. However, automated verification of data reliability seems less prioritized (0% interest), which might be due to specific trust and verification standards already in place.

Impact on Workload and Morale:

How does the current project discovery process impact your daily workload and morale?

[Copy](#)

14 responses



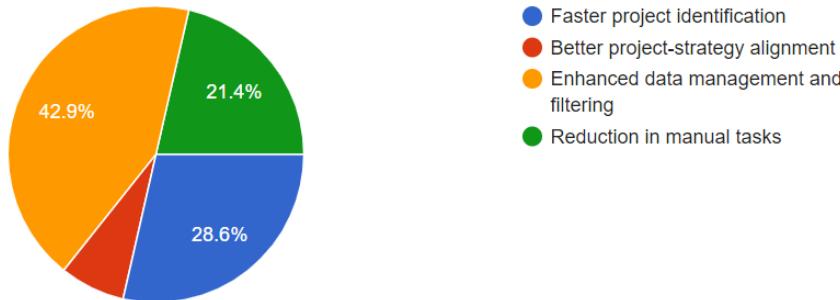
The impact of the current process on daily workload and morale is mixed. While 28.6% see a positive impact, a concerning 42.9% feel it impacts them somewhat negatively, which could affect overall job satisfaction and efficiency.

Hopes for Future Improvements:

What improvements do you hope to see in the project discovery process within the next year?

 Copy

14 responses



Employees hope for faster project identification, better project-strategy alignment, enhanced data management and filtering, and a reduction in manual tasks within the next year. This suggests a strong desire for a more streamlined, efficient, and technologically integrated discovery process.

The pre-survey results strongly indicate a need for a new tool like Scrapequest in the international development sector, as the current project discovery process is perceived as only moderately effective and inefficient. Employees face significant challenges, including aligning projects with strategic goals and managing large data volumes, which underscores the inefficiencies in the existing process. The widespread desire for features such as real-time updates and strategic data filtering highlights specific gaps in the current tools. Moreover, the considerable time spent by employees on discovery tasks, coupled with its negative impact on their workload and morale, accentuates the urgency for a more efficient, user-friendly tool that can streamline the discovery process, enhance productivity, and improve overall employee satisfaction.

[Back to Report](#)

8.2 Appendix B: Post-Survey

8.2.1 Post-Survey Form

Post-Survey Form - Scrapequest

B I U ↲ ✖

This survey aims to capture feedback on the effectiveness, usability, and overall impact of the tool on the project discovery process.

Tailored Solution for Your Needs

 SCRAPEQUEST

Select the company you work in: *

1. Vertex Special Technologies
2. Sunaula Khimti Construction Company
3. Kshamadevi Group
4. Dhukuchu Construction Company

What is your initial impression of Scrapequest? *

- Very positive
- Somewhat positive
- Neutral
- Somewhat negative
- Very negative

How intuitive did you find the interface of Scrapequest? *

- Extremely intuitive
- Very intuitive
- Moderately intuitive
- Slightly intuitive
- Not intuitive at all

Which of the the following features of Scrapequest did you like?

- Real-time project updates
- Data filtering according to strategic goals
- Automated verification of data
- Integration with current IT systems

* Did you find the features of Scrapequest particularly innovative or beneficial?

- Yes
- No

Were there any features you felt were missing from Scrapequest that would be beneficial to include? *

Please Specify

- No
- Other...

Do you believe Scrapequest would save you time in the project discovery process? *

- Yes, a significant amount of time
- Yes, a moderate amount of time
- Unsure
- No, minimal time
- No, it would not save any time

In your opinion, will Scrapequest improve the accuracy of project discovery? *

- Greatly improve
- Somewhat improve
- No change
- Somewhat worsen
- Greatly worsen

How comfortable do you feel using Scrapequest following the demonstration? *

- Very comfortable
- Comfortable
- Neutral
- Uncomfortable
- Very uncomfortable

What type of training or resources would help you to use Scrapequest effectively? (Select all that apply) *

In-depth user manual
 Online tutorials
 Interactive webinars
 On-site training sessions
 Q&A sessions with the developers

Based on what you've seen, how likely are you to recommend the implementation of Scrapequest? *

Very likely
 Likely
 Neutral
 Unlikely
 Very unlikely

Please provide any additional comments or suggestions regarding Scrapequest:

Long-answer text

Figure 183 Post Survey Form

8.2.2 Sample of Filled Post-Survey Forms

Responses cannot be edited

Post-Survey Form - Scrapequest

This survey aims to capture feedback on the effectiveness, usability, and overall impact of the tool on the project discovery process.

* Indicates required question

Tailored Solution for Your Needs

 SCRAPEQUEST

Select the company you work in: *

Vertex Special Technologies ▾

What is your initial impression of Scrapequest? *

- Very positive
- Somewhat positive
- Neutral
- Somewhat negative
- Very negative

How intuitive did you find the interface of Scrapequest? *

- Extremely intuitive
- Very intuitive
- Moderately intuitive
- Slightly intuitive
- Not intuitive at all

Which of the the following features of Scrapequest did you like?

- Real-time project updates
- Data filtering according to strategic goals
- Automated verification of data
- Integration with current IT systems

Did you find the features of Scrapequest particularly innovative or beneficial? *

- Yes
- No

Were there any features you felt were missing from Scrapequest that would be beneficial to include? *

Please Specify



No



Other:

Do you believe Scrapequest would save you time in the project discovery process? *

Yes, a significant amount of time

Yes, a moderate amount of time

Unsure

No, minimal time

No, it would not save any time

In your opinion, will Scrapequest improve the accuracy of project discovery? *

- Greatly improve
- Somewhat improve
- No change
- Somewhat worsen
- Greatly worsen

How comfortable do you feel using Scrapequest following the demonstration? *

- Very comfortable
- Comfortable
- Neutral
- Uncomfortable
- Very uncomfortable

What type of training or resources would help you to use Scrapequest effectively? (Select all that apply) *

In-depth user manual
 Online tutorials
 Interactive webinars
 On-site training sessions
 Q&A sessions with the developers

Based on what you've seen, how likely are you to recommend the implementation of Scrapequest? *

Very likely
 Likely
 Neutral
 Unlikely
 Very unlikely

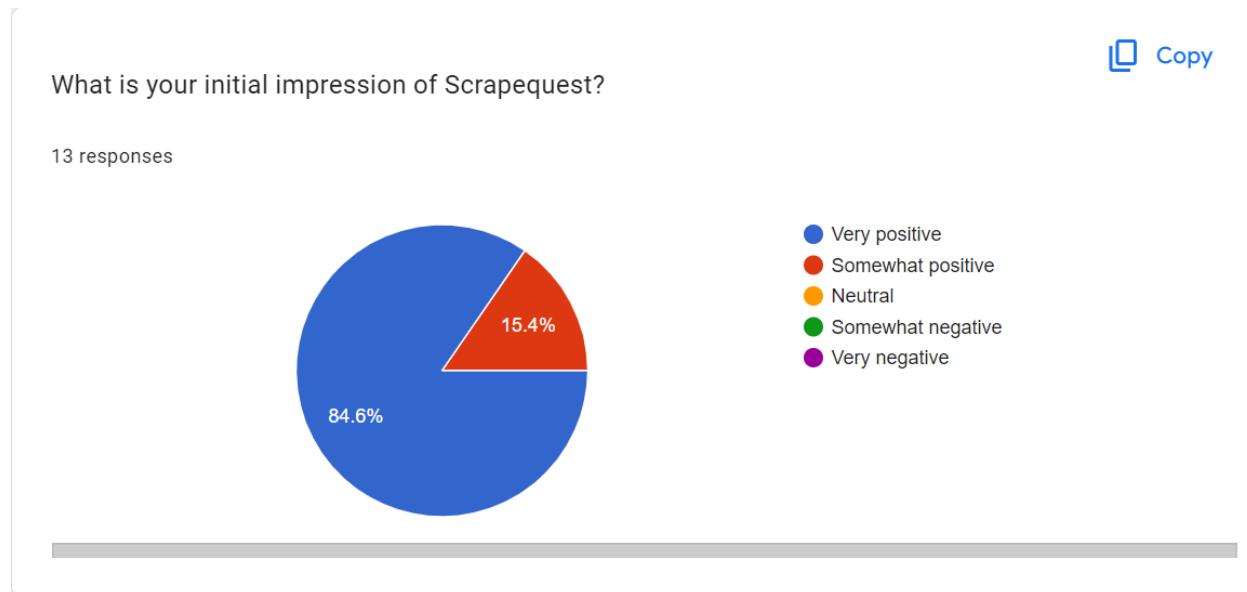
Please provide any additional comments or suggestions regarding Scrapequest:

Figure 184 Sample of Filled Post-Survey Form

8.2.3 Post-Survey Result

The survey results provide a comprehensive overview of the respondents' feedback on the tool Scrapequest, focusing on its effectiveness, usability, and impact on the project discovery process. Here are the detailed results from the survey:

Initial Impressions:



The initial impressions of Scrapequest were largely positive, with 84.6% of respondents indicating a very positive sentiment, demonstrating high initial acceptance and satisfaction.

Usability

How intuitive did you find the interface of Scrapequest?

 Copy

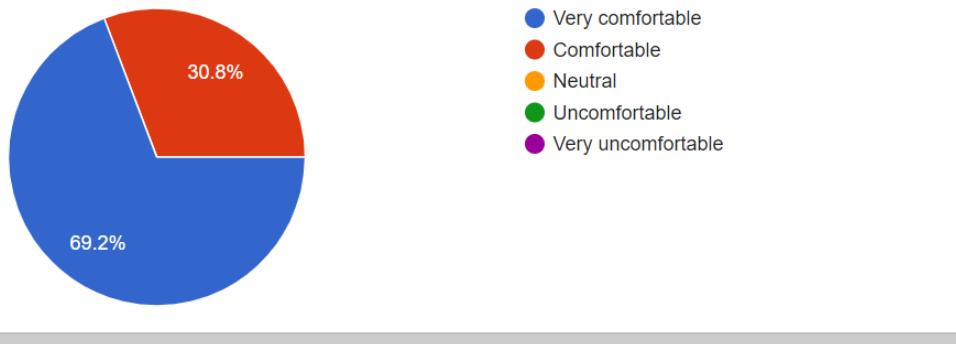
13 responses



How comfortable do you feel using Scrapequest following the demonstration?

 Copy

13 responses



The majority of respondents found the interface of Scrapequest to be intuitive:

- 15.4% found it extremely intuitive.
- 84.6% rated it from moderately to very intuitive.

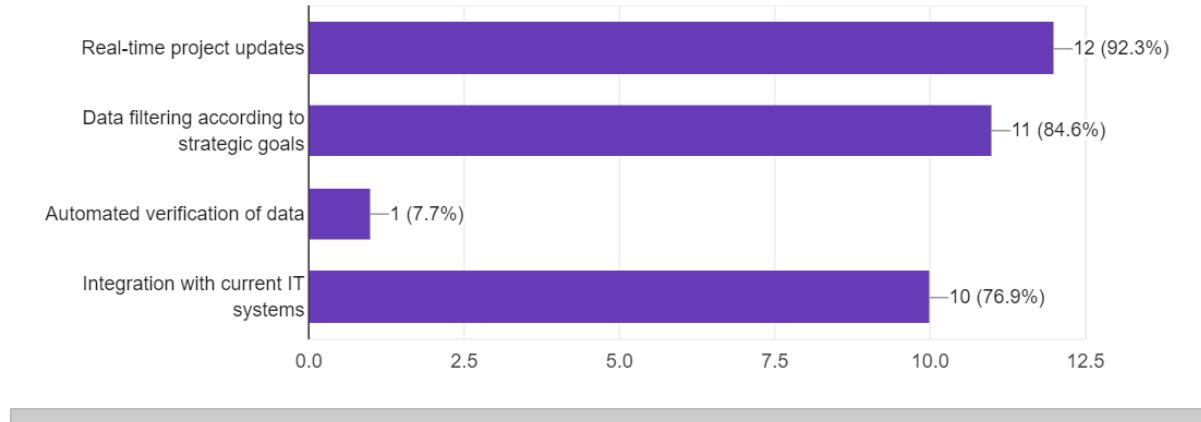
Post-demonstration, 100% of respondents felt comfortable using Scrapequest, with 30.8% feeling very comfortable and 69.2% comfortable.

Features and Innovations:

 Copy

Which of the the following features of Scrapequest did you like?

13 responses

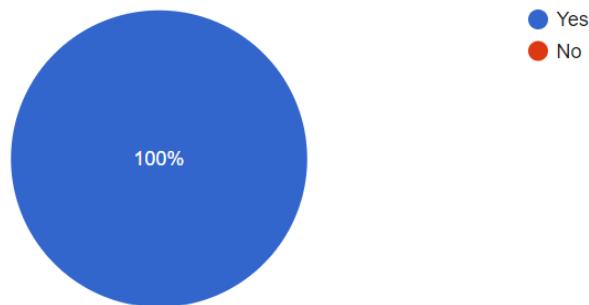


Key features like real-time project updates, data filtering, automated verification of data, and integration with current IT systems were well-received, with over 92% approval for each feature.

 Copy

Did you find the features of Scrapequest particularly innovative or beneficial?

13 responses

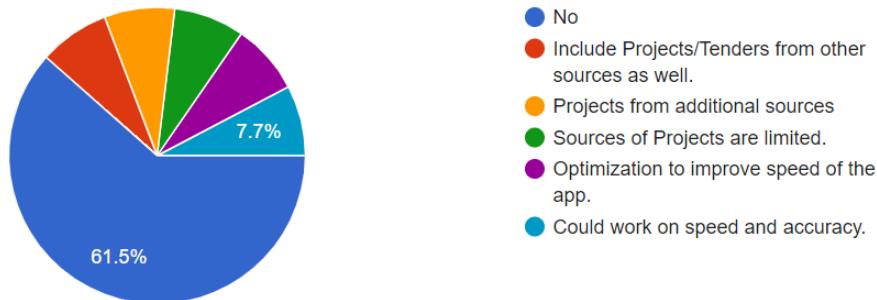


All respondents (100%) found the features of Scrapequest to be innovative or beneficial, indicating strong approval of the tool's capabilities and potential impact.

 Copy

Were there any features you felt were missing from Scrapequest that would be beneficial to include?

13 responses



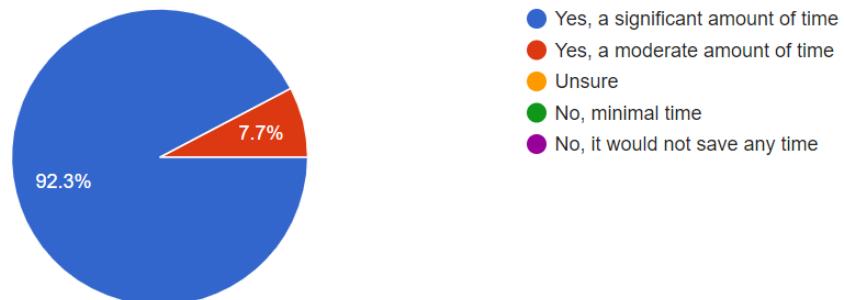
Some users suggested additional features such as including projects/tenders from other sources and optimizations to improve the app's speed and accuracy.

Impact on Project Discovery

Do you believe Scrapequest would save you time in the project discovery process?

 Copy

13 responses

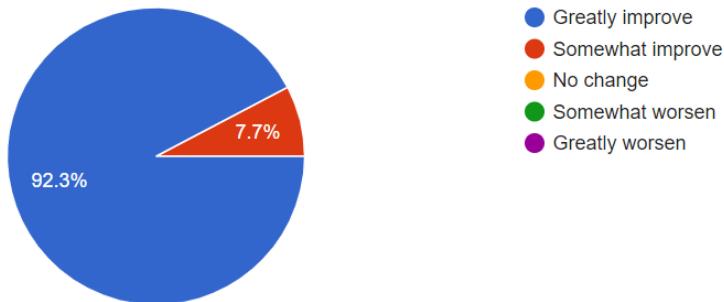


92.3% of the respondents believe Scrapequest would save them time in the project discovery process, indicating its efficiency and potential for productivity improvement.

In your opinion, will Scrapequest improve the accuracy of project discovery?

 Copy

13 responses



Similarly, 92.3% of the respondents think that Scrapequest will improve the accuracy of project discovery, highlighting its perceived effectiveness.

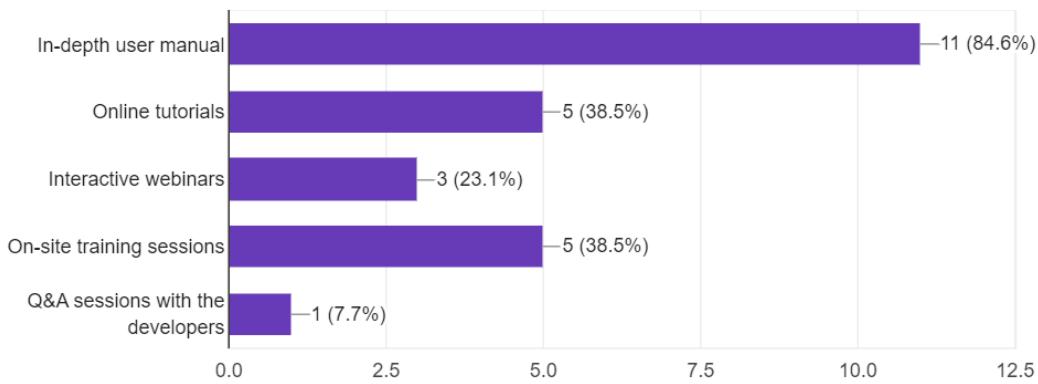
Training and Resources

What type of training or resources would help you to use Scrapequest effectively?

 Copy

(Select all that apply)

13 responses



To use Scrapequest effectively, respondents showed interest in a variety of training resources:

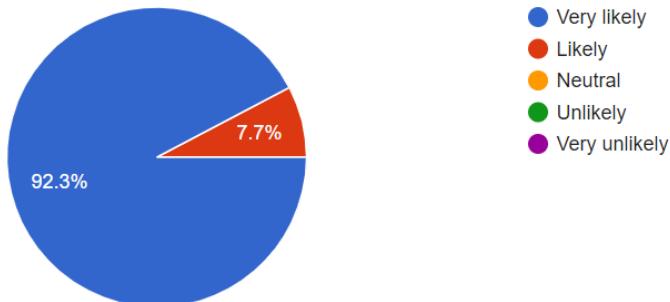
- 84.6% desired an in-depth user manual and online tutorials.
- About 38.5% wanted on-site training sessions and interactive webinars.
- A smaller portion (7.7%) would appreciate Q&A sessions with the developers.

Recommendation Likelihood

Based on what you've seen, how likely are you to recommend the implementation of Scrapequest?

 Copy

13 responses



A significant majority (92.3%) of respondents are likely to recommend the implementation of Scrapequest, showing strong support for its broader adoption within their organizations.

In essence, the survey results indicate that Scrapequest is well-received in terms of usability and effectiveness, with specific strengths in intuitive design, innovative features, and perceived improvements in project discovery processes. However, there is room for enhancement, particularly in terms of broadening the data sources and improving app speed and accuracy. Training resources are also highlighted as an area to bolster user confidence and maximize the utility of Scrapequest.

[Back to Report](#)

8.3 Appendix C: Sample Codes

8.3.1 Sample Code of the UI

8.3.1.1 Web Application

8.3.1.1.1 User Login/Signup Page

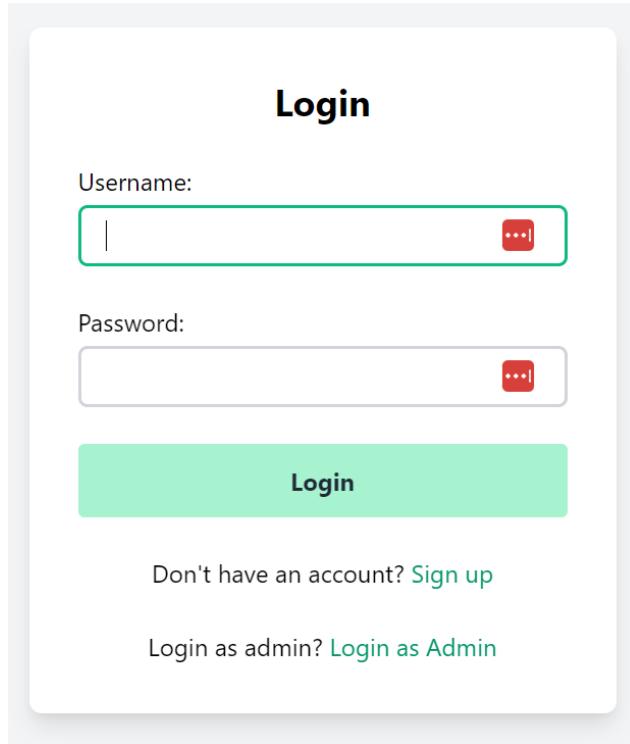


Figure 185 Login UI

Sign Up

Username:

Required. 150 characters or fewer. Letters, digits and @./+/-/_ only.

Email:

Required. Inform a valid email address.

Password:

Your password can't be too similar to your other personal information.
Your password must contain at least 8 characters.
Your password can't be a commonly used password.
Your password can't be entirely numeric.

Password confirmation:

Enter the same password as before, for verification.

Sign up

Already have an account? [Login](#)

Figure 186 Signup Page

```
from django import forms
from django.contrib.auth.forms import UserCreationForm
from django.contrib.auth.models import User

##login/signup
class SignUpForm(UserCreationForm):
    email = forms.EmailField(max_length=254, help_text='Required. Inform a valid email address.')

    class Meta:
        model = User
        fields = ('username', 'email', 'password1', 'password2', )

class LoginForm(forms.Form):
    username = forms.CharField()
    password = forms.CharField(widget=forms.PasswordInput)
```

Figure 187 Login/Signup Forms

Above pasted form definitions do following:

- It imports the required modules for forms and authentication.
- Defines a SignUpForm that inherits from Django's UserCreationForm, adding an email field.
- Specifies that the SignUpForm will apply to the User model and include fields for the username, email, password1, and password2.
- Defines a LoginForm with fields for a username and a password, using Django's forms functionality.

```

def signup(request):
    if request.method == 'POST':
        form = SignUpForm(request.POST)
        if form.is_valid():
            user = form.save(commit=False)
            user.is_active = False # User will be inactive until approved
            user.save()
            return redirect('login') # Redirect to login page after signup
    else:
        form = SignUpForm()
    return render(request, 'signup.html', {'form': form})

def user_login(request):
    if request.method == 'POST':
        form = LoginForm(request.POST)
        if form.is_valid():
            username = form.cleaned_data.get('username')
            password = form.cleaned_data.get('password')
            user = authenticate(username=username, password=password)
            if user is not None:
                if user.is_active:
                    login(request, user)
                    return redirect('home') # Redirect to a home page or dashboard
                else:
                    messages.error(request, 'Your account has not been approved yet.')
            else:
                messages.error(request, 'Invalid username or password.')
        else:
            for field in form.errors:
                form[field].field.widget.attrs['class'] += ' border-red-500'
            messages.error(request, 'Please correct the errors below.')
    else:
        form = LoginForm()
    return render(request, 'login.html', {'form': form})

```

Figure 188 Login/Signup Backend

Above pasted view functions achieve the following:

- The **signup** view handles the user registration process. It creates an instance of **SignUpForm** when a POST request is received. If the form is valid, it creates a new User object but doesn't commit it to the database (commit=False). It sets **user.is_active** to False, meaning the user will remain inactive until approved. After saving the user, it redirects to the login page.
- The **user_login** view manages the login process. It also creates an instance of **LoginForm** when a POST request is received. If the form is valid, it retrieves the username and password from the form, tries to authenticate the user, and if successful, logs the user in and redirects them to the home page. If the user is inactive, it shows an error message that the account hasn't been approved yet. For invalid credentials, it shows an error message.

```
<form method="post" class="space-y-6">
    {% csrf_token %}
    {{ form.non_field_errors }}
    {% if messages %}
        <div class="space-y-2">
            {% for message in messages %}
                <p class="bg-red-100 border-l-4 border-red-500 text-red-700 p-2" role="alert">
                    | {{ message }}
                </p>
            {% endfor %}
        </div>
    {% endif %}
    <div class="space-y-1">
        {{ form.username.label_tag }}
        {{ form.username|add_class:"block w-full border-2 border-gray-300 rounded-md py-2 px-4 mb-3 leading-tight focus:outline-none focus:border-green-500" }}
        {% if form.username.errors %}
            <p class="text-red-500 text-xs italic">{{ form.username.errors }}</p>
        {% endif %}
    </div>
    <div class="space-y-1">
        {{ form.password.label_tag }}
        {{ form.password|add_class:"block w-full border-2 border-gray-300 rounded-md py-2 px-4 mb-3 leading-tight focus:outline-none focus:border-green-500" }}
        {% if form.password.errors %}
            <p class="text-red-500 text-xs italic">{{ form.password.errors }}</p>
        {% endif %}
    </div>
    <button type="submit" class="w-full bg-green-200 hover:bg-green-300 text-gray-800 font-bold py-3 px-4 rounded focus:outline-none focus:shadow-outline transition">Don't have an account? <a href="{% url 'signup' %}" class="text-green-600 hover:text-green-800">Sign up</a></p>
    <p class="text-center">Login as admin? <a href="/admin/" class="text-green-600 hover:text-green-800">Login as Admin</a></p>
</form>
```

Figure 189 Styled Login Form for error validation.

```
<form method="post" class="space-y-6">
    {% csrf_token %}
    {% if form.non_field_errors %}
        <div class="bg-red-100 border-l-4 border-red-500 text-red-700 p-3 mb-3" role="alert">
            {{ form.non_field_errors }}
        </div>
    {% endif %}
    <div class="space-y-1">
        {{ form.username.label_tag }}
        {{ form.username|add_class:"block w-full border-2 border-gray-300 rounded-md py-2 px-4 mb-3 leading-tight focus:outline-none focus:border-green-500" }}
        {% if form.username.errors %}
            <p class="text-red-500 text-xs italic">{{ form.username.errors }}</p>
        {% endif %}
        <p class="text-gray-600 text-xs italic">{{ form.username.help_text }}</p>
    </div>
    <div class="space-y-1">
        {{ form.email.label_tag }}
        {{ form.email|add_class:"block w-full border-2 border-gray-300 rounded-md py-2 px-4 mb-3 leading-tight focus:outline-none focus:border-green-500" }}
        {% if form.email.errors %}
            <p class="text-red-500 text-xs italic">{{ form.email.errors }}</p>
        {% endif %}
        <p class="text-gray-600 text-xs italic">{{ form.email.help_text }}</p>
    </div>
    <div class="space-y-1">
        {{ form.password1.label_tag }}
        {{ form.password1|add_class:"block w-full border-2 border-gray-300 rounded-md py-2 px-4 mb-3 leading-tight focus:outline-none focus:border-green-500" }}
        {% if form.password1.errors %}
            <p class="text-red-500 text-xs italic">{{ form.password1.errors }}</p>
        {% endif %}
        <p class="text-gray-600 text-xs italic">{{ form.password1.help_text }}</p>
    </div>
    <div class="space-y-1">
        {{ form.password2.label_tag }}
        {{ form.password2|add_class:"block w-full border-2 border-gray-300 rounded-md py-2 px-4 mb-3 leading-tight focus:outline-none focus:border-green-500" }}
        {% if form.password2.errors %}
            <p class="text-red-500 text-xs italic">{{ form.password2.errors }}</p>
        {% endif %}
        <p class="text-gray-600 text-xs italic">{{ form.password2.help_text }}</p>
    </div>
    <button type="submit" class="w-full bg-green-200 hover:bg-green-300 text-gray-800 font-bold py-3 px-4 rounded focus:outline-none focus:shadow-outline transition">Already have an account? <a href="{% url 'login' %}" class="text-green-600 hover:text-green-800">Login</a></p>
</form>
```

Figure 190 Styled Signup Form for error validation.

Above pasted HTML templates achieve following:

- These snippets show the HTML template for the login and signup forms, respectively, using Django templating language.
- They include CSRF protection tokens which are necessary for security in POST forms to prevent Cross-Site Request Forgery attacks.
- The templates display form fields and error messages styled with CSS classes. For instance, if there are any non-field errors, they are displayed in a div with a red background and border.
- It uses the messages framework to display any flash messages that have been added in the view, for example, an error message if the login fails.
- At the bottom, there are links for users to navigate to the signup page if they don't have an account, or to the admin login page if applicable.

8.3.1.1.2 Home Page

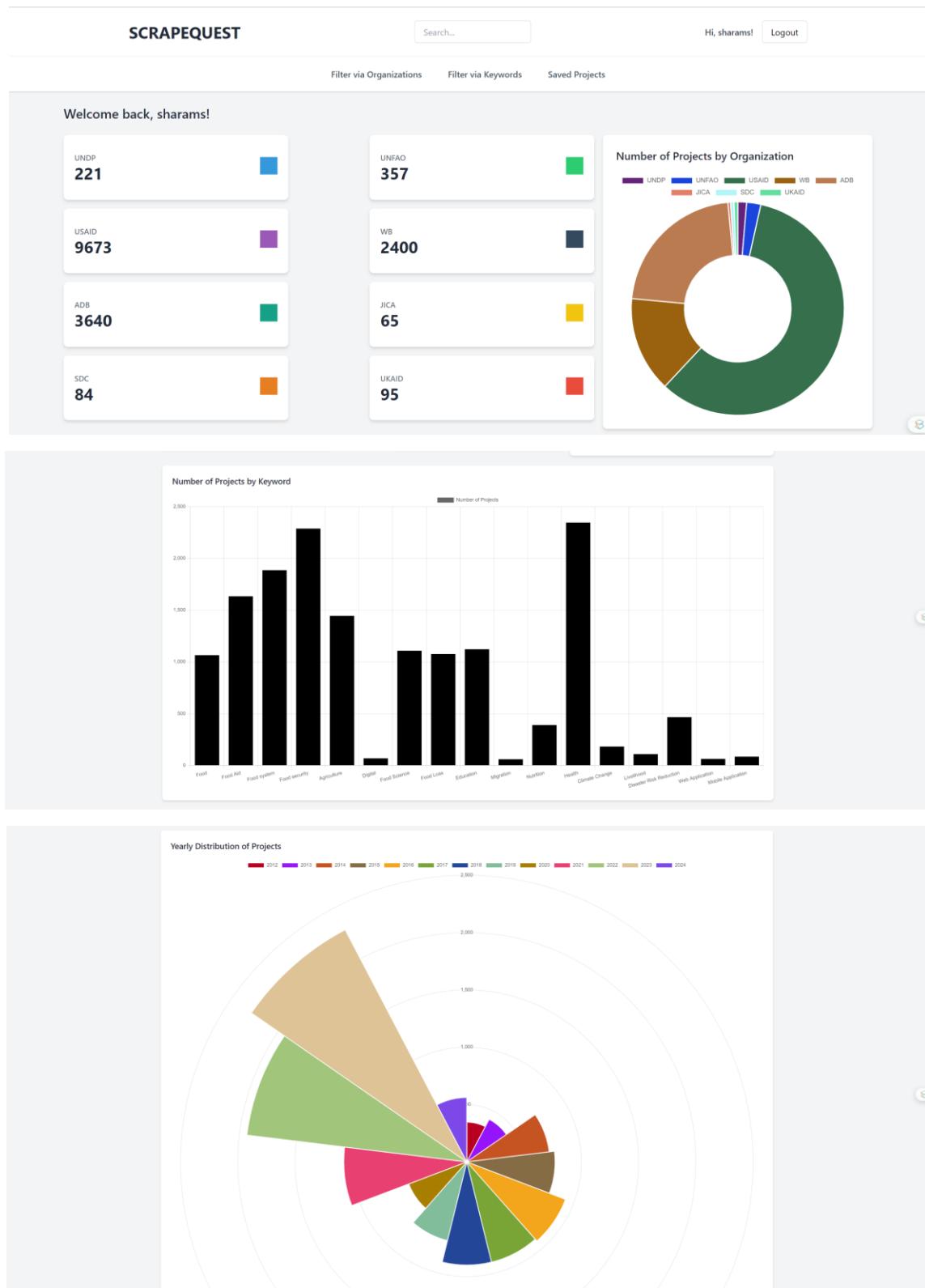


Figure 191 Homepage.

8.3.1.1.2.1 Navbar

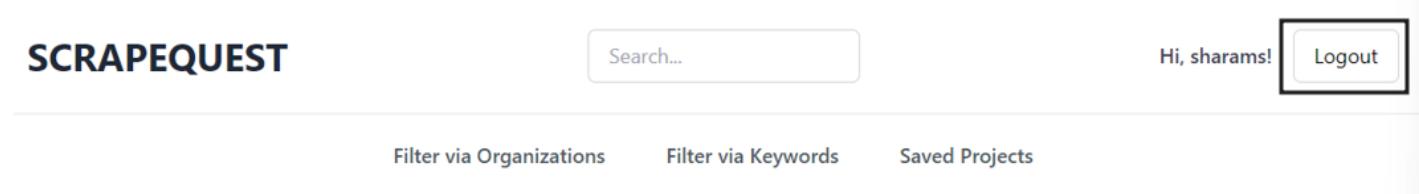


Figure 192 Screenshot of Navbar

```

<header class="bg-white shadow">
  <div class="max-w-7xl mx-auto px-4 flex justify-between items-center py-6">
    <!-- SCRAPEQUEST Text -->
    <div class="flex items-center">
      <a href="{% url 'homepage' %}" class="text-3xl font-bold text-gray-800 hover:text-gray-600 id="home">SCRAPEQUEST</a>
    </div>

    <!-- Search Container -->
    <div class="flex-1 flex justify-center">
      <div id="search-container" class="search-container relative" data-search-url="{% url 'search_bar' %}">
        <input type="text" id="search-input" class="border border-gray-300 rounded-md py-2 px-4" placeholder="Search..." onfocus="toggleSearchResults(true)" oninput="fetchSearchResults()">
        <div id="search-results" class="absolute z-50 bg-white shadow-md rounded px-2 py-1 text-sm w-56 hidden"></div>
      </div>
    </div>

    <!-- Greeting and Logout -->
    <div class="flex items-center">
      <span class="text-gray-700 font-medium mr-4">Hi, {{ user.username }}!</span>
      <div id="logout">
        <form action="/logout/" method="post">
          {% csrf_token %}
          <button type="submit" class="border border-gray-300 rounded-md py-2 px-4 hover:bg-gray-100 focus:outline-none">Logout</button>
        </form>
      </div>
    </div>
  </div>

  <!-- Lower Navigation Links -->
  <!-- Lower Navigation Links -->
<div class="border-t border-gray-200">
  <nav class="max-w-7xl mx-auto px-4 flex justify-center space-x-6 py-3">
    <a href="{% url 'project_list' %}" class="text-gray-600 hover:bg-gray-100 px-3 py-2 rounded-md text-base font-medium" id="project-list">Filter via Organizations</a>
    <a href="{% url 'search_by_keyword' %}" class="text-gray-600 hover:bg-gray-100 px-3 py-2 rounded-md text-base font-medium" id="show-project-details">Filter via Keywords</a>
    <a href="{% url 'view_saved_project' %}" class="text-gray-600 hover:bg-gray-100 px-3 py-2 rounded-md text-base font-medium" id="show-saved-project">Saved Projects</a>
  </nav>
</div>

```

Figure 193 HTML & CSS for Navbar

The navbar structure and style are defined using HTML and Tailwind CSS. The header element uses Tailwind CSS classes for styling. Various flex container elements are used, including a homepage link, a search container with input, a greeting and logout form, and lower navigation links for additional page navigation.

```

from django.contrib.auth import logout
from django.shortcuts import redirect

def logout_view(request):
    if request.method == 'POST':
        logout(request)
        return redirect('login')
    else:
        # Optionally, handle incorrect usage:
        return HttpResponseRedirect(['POST'])

```

Figure 194 Backend function to handle Logout.

Above pasted Logout View achieves following:

- It handles the logout functionality.
- The function checks if the request method is POST. If it is, the user is logged out using the logout function imported from **django.contrib.auth**, and then redirected to the login page.
- If the request method is not POST, the view returns an **HttpResponseNotAllowed** object, which essentially blocks any non-POST requests to this endpoint, ensuring that logout can only be performed with a POST request which prevents CSRF attacks.

8.3.1.1.2.2 Search Button

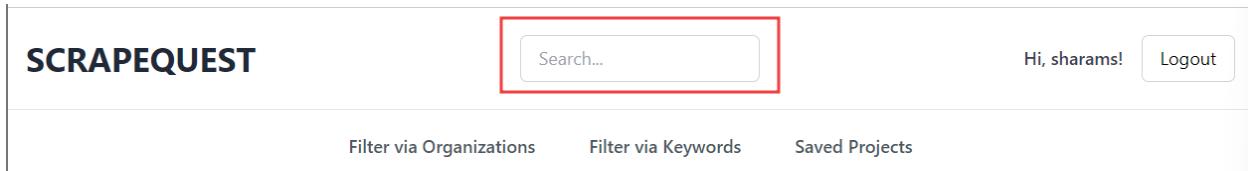


Figure 195 Search Button

2020

UNDP Projects

UNFAO Projects

Name of the Project: GCP /NEP/076/GCF – Building a Resilient Churia Region in Nepal (BRCRN)
Duration of the Project: May 2020 to May 2027
Link to the project: [View](#)

WB

Name of the Project: OSRO/NEP/001: Immediate TA for animal health system to address emerging / priority zoonotic diseases & health threat
Duration of the Project: October 2020 to September 2022
Link to the project: [View](#)

IICA

Name of the Project: OSRO/NEP/001: Immediate TA for animal health system to address emerging / priority zoonotic diseases & health threat
Duration of the Project: October 2020 to September 2022
Link to the project: [View](#)

Name of the Project: GCP /NEP/076/GCF – Building a Resilient Churia Region in Nepal (BRCRN)
Duration of the Project: May 2020 to May 2027
Link to the project: [View](#)

Name of the Project: GCP /NEP/076/GCF – Building a Resilient Churia Region in Nepal (BRCRN)
Duration of the Project: May 2020 to May 2027
Link to the project: [View](#)

Figure 196 Search Button in Action

```

<% if results %>
<% for organization, projects in results.items %>


<h2 class="text-xl font-bold text-gray-800 mb-4">{{ organization }} Projects</h2>
  <div class="project-container space-y-4">
    {% for project in projects %}
      <div class="project bg-white border border-gray-300 rounded-lg p-4 hover:shadow-lg transition-shadow duration-300">
        <h3 class="text-lg font-semibold text-gray-700">{{ project.Name_of_the_Project }}</h3>
        {% for field, value in project.items %}
          {% if field != 'Name_of_the_Project' and field != 'Date' and value != 'N/A' %}
            <p class="text-gray-600">
              {{ field }}:
              {% if field == 'Link to the project' or field == 'Link to the Procurement' %}
                <a href="{{ value }}" target="_blank" class="text-blue-600 hover:text-blue-800">View</a>
              {% else %}
                {{ value }}
              {% endif %}
            </p>
          {% endif %}
        {% endfor %}
      </div>
    {% endfor %}
  </div>
<% endfor %>
{% else %}
  <p class="text-gray-600">No results found.</p>
{% endif %}


```

Figure 197 HTML Page for Search Button

```

#search-results {
  display: none; /* Initially hidden */
  width: 320%; /* Width adjusted to 100% to fit the container */
  max-height: 800px; /* Set max height for visible area */
  background-color: #rgba(200, 200, 200, 0.5); /* Semi-transparent grey */
  box-shadow: 0 4px 8px #rgba(0, 0, 0, 0.1);
  position: absolute;
  left: -100%;
  top: 162%; /* Align directly below the search input */
  z-index: 1000;
  overflow-y: auto; /* Allow vertical scrolling */
  border-radius: 5px; /* Adds rounded corners */
  transition: all 0.5s ease-out; /* Smooth transition for expanding */
  padding: 20px;
}

#search-results a {
  --tw-text-opacity: 1;
  color: #2978A6; /* This is Tailwind's blue-600 color */
}

a:hover {
  --tw-text-opacity: 1;
  color: #30A6C8; /* This is Tailwind's blue-700 color */
}

```

Figure 198 CSS for toggling Search Button

The first snippet is part of a Django template that generates a list of projects within organizations based on user search query. It loops through results, displays the project lists in detail.

The second snippet provides CSS styling for search results. The CSS style #search-results ID and a: hover style provides visual feedback.

```
// Search button functionality
function toggleSearchResults(expand) {
    if (expand) {
        $('#search-results').addClass('expanded').fadeIn('slow');
    } else {
        $('#search-results').removeClass('expanded').fadeOut('slow');
    }
}

function fetchSearchResults() {
    var query = $("#search-input").val();
    var searchUrl = $('#search-container').data('search-url');

    if (query.length > 0) {
        if (!$('#search-results').hasClass('expanded')) {
            toggleSearchResults(true);
        }
        $.ajax({
            url: searchUrl,
            data: { 'search': query },
            success: function(data) {
                $('#search-results').html(data.html);
            }
        });
    } else {
        toggleSearchResults(false);
    }
}

// Close the search results if the user clicks outside
$(document).click(function(event) {
    if (!$(event.target).closest('#search-container').length) {
        toggleSearchResults(false);
    }
});

function performSearch(keyword) {
    $("#search-input").val(keyword);
    fetchSearchResults();
}
```

Figure 199 JS for handling Search Query

Above code snippet handles search query within navbar. The function **toggleSearchResults(expand)** adds a class to the search results container and fades it in or out using jQuery's fadeIn or fadeOut functions. **fetchSearchResults()** fetches search results based on user query input, updates the container's HTML, or hides them if the query is empty. The document click event handler hides search results when the user clicks outside the container. The function **performSearch(keyword)** sets the search input value to a given keyword and executes the search.

```

##search functionality
search_collections = {
    'UNDP': ['Name of the Project'],
    'UNFAO': ['Name of the Project', 'Duration of the Project'],
    'USAID': ['Name of the Project', 'Duration of the Project', 'Sector'],
    'WB': ['Name of the Project', 'Procurement Details', 'Date of Publication'],
    'ADB': ['Name of the Project', 'Project Status', 'Project Description'],
    'JICA': ['Name of the Project'],
    'SDC': ['Name of the Project', 'Project Details', 'Duration of the Project'],
    'UKAID': ['Name of the Project', 'Project Details', 'Start-Date'],
}

for collection_name, fields in search_collections.items():
    # Create a list of tuples for the fields to be indexed
    index_fields = [(field, 'text') for field in fields]

    # Get the collection object
    collection = db[collection_name]

    # Get current text indexes on the collection
    current_indexes = collection.index_information()

    # Prepare a set of fields for the new index for comparison
    new_index_fields = set(index_fields)

    # Flag to determine if an equivalent index exists
    index_exists = False

    # Check each existing index
    for index in current_indexes.values():
        if 'key' in index and set(index['key']) == new_index_fields:
            # An equivalent index exists
            index_exists = True
            break

    # If an equivalent index does not exist, create it
    if not index_exists:
        collection.create_index(index_fields)

from django.http import HttpResponseRedirect, JsonResponse
from django.template.loader import render_to_string

def custom_search(request):
    search_query = request.GET.get('search', '').strip()
    results = {}

    if search_query:
        for collection_name, fields in search_collections.items():
            if collection_name in db.list_collection_names():
                collection = db[collection_name]
                # Perform a text search
                projects = collection.find(
                    {"$text": {"$search": search_query}},
                    {'_id':0}
                ).sort([{'textScore': {'$meta': 'textScore'}}])
                results[collection_name] = list(projects)

    if request.headers.get('X-Requested-With') == 'XMLHttpRequest':
        html = render_to_string('partials/search_results.html', {'results': results})
        return JsonResponse({'html': html})
    else:
        return render(request, 'base.html', {'results': results})

```

Figure 200 Backend for Search Functionality

The above snippet is responsible for handling search functionality within the application, providing a way to search across multiple MongoDB collections and return the results to the user either as a full page reload or dynamically via AJAX.

The **search_collections** dictionary maps collection names to searchable fields within those collections. It creates a list of **index_fields** for each collection, retrieves the collection object, fetches current index information, prepares a new set, and checks if an equivalent index exists. The **custom_search** function extracts the search query from the request's GET parameters, loops through the **search_collections**, performs a text search within the collection, sorts results based on a text score, and adds the search results to the results dictionary.

8.3.1.1.2.3 Dashboard with Analytics

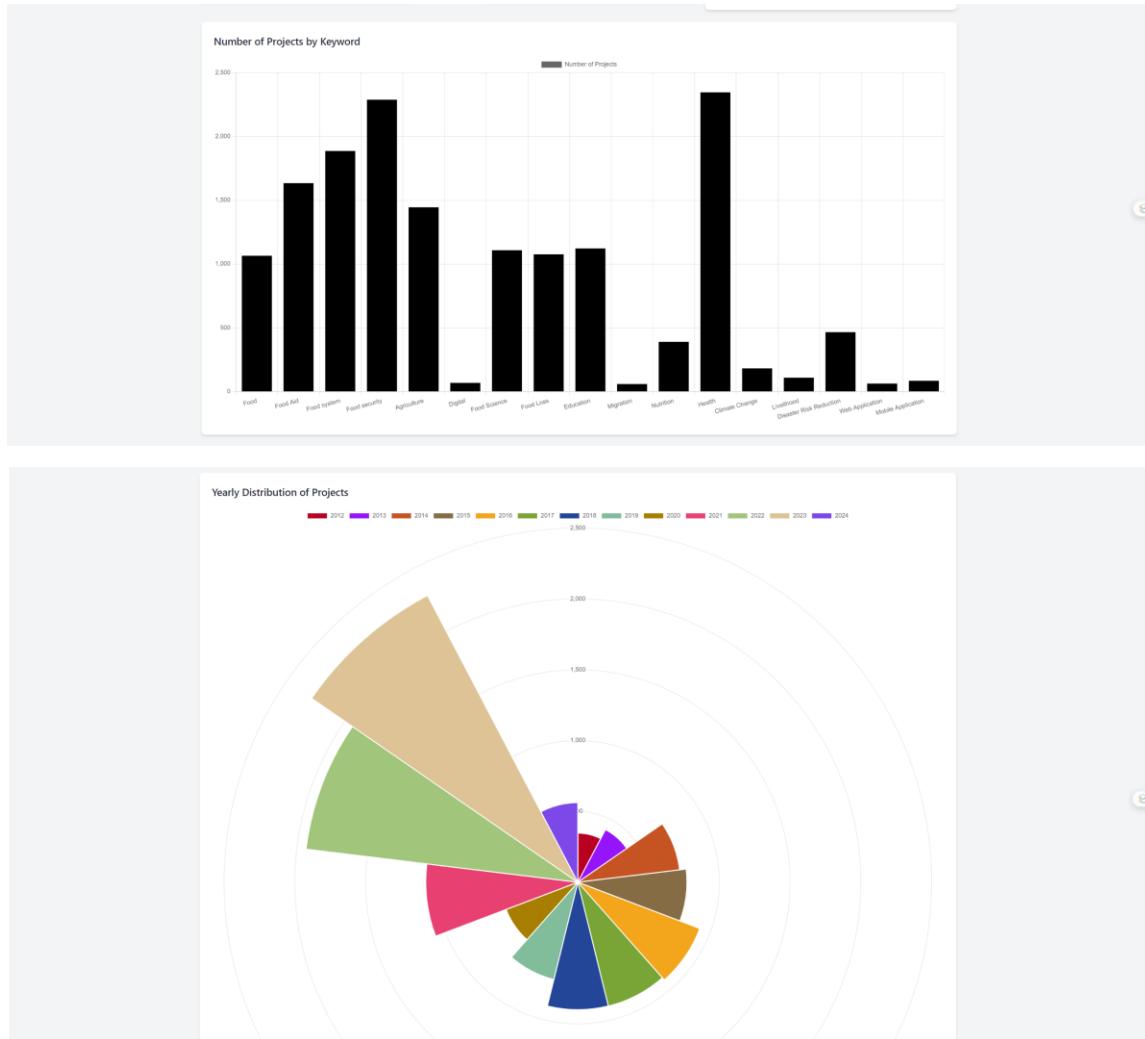


Figure 201 Dashboard with Analytics

```

<!-- Stats Cards and Organization Doughnut Chart -->


<!-- Stats Cards -->
    <div class="w-full lg:w-2/3 flex flex-wrap justify-between">
        {% for org, count in project_counts.items %}
            <div class="bg-white p-5 rounded-lg shadow-md flex items-center justify-between mb-4 mr-4 w-full sm:w-5/12">
                <div>
                    <div class="text-sm font-medium text-gray-500">{{ org }}</div>
                    <div class="text-3xl font-bold text-gray-800">{{ count }}</div>
                </div>
                <div class="text-2xl">
                    <!-- Colored square for organization -->
                    <span class="inline-block w-8 h-8" style="background-color: {{ org|color_for_organization }}"></span>
                </div>
            </div>
        {% endfor %}
    </div>

    <!-- Doughnut Chart for Organization -->
    <div class="w-full lg:w-1/3">
        <div class="bg-white p-6 rounded-lg shadow-md">
            <h3 class="text-xl font-semibold text-gray-800 mb-4">Number of Projects by Organization</h3>
            <canvas id="orgDoughnutChart"></canvas>
        </div>
    </div>
</div>

<!-- Bar Chart for Keyword -->
<div class="bg-white p-6 rounded-lg shadow-md mb-6">
    <h3 class="text-xl font-semibold text-gray-800 mb-4">Number of Projects by Keyword</h3>
    <canvas id="keywordBarChart"></canvas>
</div>

<!-- Another Chart for Yearly Distribution -->
<div class="bg-white p-6 rounded-lg shadow-md mb-6">
    <h3 class="text-xl font-semibold text-gray-800 mb-4">Yearly Distribution of Projects</h3>
    <canvas id="yearlyDistributionChart"></canvas>
</div>

<!-- Retrieve the data for charts -->
{{ project_counts|json_script:"projectCountsData" }}
{{ keyword_counts|json_script:"keywordCountsData" }}
{{ yearly_counts|json_script:"yearlyCountsData" }}


```

Figure 202 HTML code for Dashboard.

HTML Dashboard Page:

- The HTML snippet shows the structure of a dashboard with statistics cards and placeholders for charts (Doughnut, Bar, and Polar Area) to visualize data.
- The statistics cards iterate through organizational stats and display counts alongside organization names and colors.
- The canvas elements are placeholders for the charts to be rendered on the webpage.

```

<script>
    // Retrieve the data from the JSON script tag and parse it
    const projectCounts = JSON.parse(document.getElementById('projectCountsData').textContent);
    const keywordCounts = JSON.parse(document.getElementById('keywordCountsData').textContent);
    const yearlyCounts = JSON.parse(document.getElementById('yearlyCountsData').textContent);

    // Colors for the organization chart
    const orgColors = Object.keys(projectCounts).map(org => getRandomColor()); // Function to generate random colors

    // Doughnut Chart for Organization
    const orgDoughnutChart = new Chart(document.getElementById('orgDoughnutChart'), {
        type: 'doughnut',
        data: {
            labels: Object.keys(projectCounts),
            datasets: [
                {
                    data: Object.values(projectCounts),
                    backgroundColor: orgColors,
                }
            ]
        },
        options: {
            animation: {
                animateScale: true
            }
        }
    });

    // Bar Chart for Keywords
    const keywordBarChart = new Chart(document.getElementById('keywordBarChart'), {
        type: 'bar',
        data: {
            labels: Object.keys(keywordCounts),
            datasets: [
                {
                    label: 'Number of Projects',
                    data: Object.values(keywordCounts),
                    backgroundColor: funColors(keywordCounts.length), // Function to generate an array of fun colors
                }
            ],
            options: {
                animation: {
                    duration: 2000,
                    easing: 'easeOutElastic',
                },
                scales: {
                    yAxes: [
                        {
                            ticks: {
                                beginAtZero: true
                            }
                        }
                    ]
                }
            }
        }
    });

    // Yearly Distribution Chart - Polar Area
    const yearlyDistributionChart = new Chart(document.getElementById('yearlyDistributionChart'), {
        type: 'polarArea',
        data: {
            labels: Object.keys(yearlyCounts),
            datasets: [
                {
                    label: 'Number of Projects',
                    data: Object.values(yearlyCounts),
                    backgroundColor: funColors(Object.keys(yearlyCounts).length),
                }
            ],
            options: {
                animation: {
                    animateRotate: true
                }
            }
        }
    });

    // Functions to generate colors
    function getRandomColor() {
        const letters = '0123456789ABCDEF';
        let color = '#';
        for (let i = 0; i < 6; i++) {
            color += letters[Math.floor(Math.random() * 16)];
        }
        return color;
    }

    function funColors(count) {
        const colors = [];
        for (let i = 0; i < count; i++) {
            colors.push(getRandomColor());
        }
        return colors;
    }

```

Figure 203 JS for Generating and Styling Charts

JavaScript Code:

- The JavaScript snippet is using the Chart.js library to create charts on the dashboard.
- It retrieves data for the charts from elements within the HTML that are populated via Django's template system (the JSON script tags).
- getRandomColor() and funColors() are utility functions to generate random colors for the chart elements.
- The script then initializes a Doughnut Chart, a Bar Chart, and a Polar Area Chart with the retrieved data.

```
#visualization for homepage
from collections import Counter
from django.shortcuts import render
from django.contrib.auth.decorators import login_required

@login_required
def dashboard_view(request):
    collections = ['UNDP', 'UNFAO', 'USAID', 'WB', 'ADB', 'JICA', 'SDC', 'UKAID']
    project_counts = {collection: db[collection].count_documents({}) for collection in collections}

    keywords = load_keywords()

    # Aggregate data for keywords across all collections
    keyword_counts = Counter()
    for collection in collections:
        for keyword in keywords:
            keyword_counts[keyword] += db[collection].count_documents({'$text': {'$search': keyword}})

    # Count projects for 2023 and 2024
    years = [str(year) for year in range(2012, 2025)] # Years from 2012 to 2024
    yearly_counts = {year: 0 for year in years}

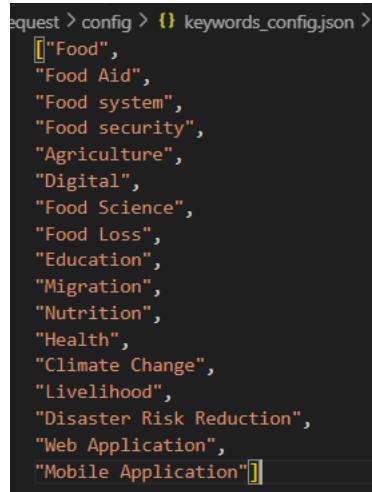
    for year in years:
        for collection_name in collections:
            collection = db[collection_name]
            # Use the text index to search for the year as a keyword
            count = collection.count_documents({'$text': {'$search': year}})
            yearly_counts[year] += count

    context = {
        'project_counts': project_counts,
        'keyword_counts': dict(keyword_counts),
        'yearly_counts': yearly_counts,
    }
    return render(request, 'home.html', context)
```

Figure 204 Backend for generating data in Chart.

View Function:

- This Python snippet is the view function that prepares the data for the dashboard.
- It requires a logged-in user (@login_required).
- The function aggregates project counts from various collections, keyword counts across all collections, and yearly counts for projects.
- It then passes this data into the context for rendering in the template, allowing the JavaScript to pick it up and generate the visualizations.



```
request > config > { keywords_config.json >
  ["Food",
   "Food Aid",
   "Food system",
   "Food security",
   "Agriculture",
   "Digital",
   "Food Science",
   "Food Loss",
   "Education",
   "Migration",
   "Nutrition",
   "Health",
   "Climate Change",
   "Livelihood",
   "Disaster Risk Reduction",
   "Web Application",
   "Mobile Application"]]
```

A screenshot of a code editor showing a JSON file named 'keywords_config.json'. The file contains a single array of strings representing keywords. The keywords listed are: Food, Food Aid, Food system, Food security, Agriculture, Digital, Food Science, Food Loss, Education, Migration, Nutrition, Health, Climate Change, Livelihood, Disaster Risk Reduction, Web Application, and Mobile Application.

Figure 205 JSON file to load Keywords.

JSON File for Keywords:

- This JSON file contains an array of keywords that the system uses to categorize or tag projects.
- The backend view uses this list to aggregate keyword counts, as shown in the backend snippet.

8.3.1.1.3 Filter via Organizations

The screenshot shows the SCRAPEQUEST interface. At the top, there is a search bar and a user profile with 'Hi, sharams!' and 'Logout'. Below the header, there are three filter buttons: 'Filter via Organizations' (highlighted with a red box), 'Filter via Keywords', and 'Saved Projects'. A horizontal navigation bar below the filters includes links for UNDP, UNFAO, USAID, JICA, ADB, WB, UKAID, and SDC. The main content area is titled 'UNDP Projects' and displays a grid of nine project cards. Each card contains a project title, a 'View Project' link, and a 'Save Project' button.

Project Title	Description	Action
Recovery & Resilience	Enhancing human security through local climate actions	View Project Save Project
Sambodhan: Temporary Basic Income	European Union Support to Inclusive Federalism (EUSIF)	View Project Save Project
Leaving No One Behind in Nepal's Green, Resilient, and Inclusive COVID-19 Recovery	Renewable Energy for Rural Livelihood	View Project Save Project
Sustainable Tourism for Livelihood Recovery	Value Chain Development of Fruit and Vegetables Project (VCDP)	View Project Save Project
Developing Climate Resilient Livelihoods in the Vulnerable Watershed in Nepal	Strengthening Urban Preparedness, Earthquake Preparedness and Response in Western Regions of Nepal	View Project Save Project
	Support to Knowledge and Lifelong Learning Skills (SKILLS) Programme	View Project Save Project
		View Project Save Project
		View Project Save Project
		View Project Save Project

Figure 206 Page to Display Projects Filtered Via Organizations

```

<div class="container mx-auto p-6">
  <!-- Project Buttons -->
  <div class="flex flex-wrap gap-2 justify-center p-4">
    <button onclick="showProjects('UNDP')" class="bg-gray-200 hover:bg-gray-400 text-gray-800 font-semibold py-2 px-4 rounded">UNDP</button>
    <button onclick="showProjects('UNFAO')" class="bg-gray-200 hover:bg-gray-400 text-gray-800 font-semibold py-2 px-4 rounded">UNFAO</button>
    <button onclick="showProjects('USAID')" class="bg-gray-200 hover:bg-gray-400 text-gray-800 font-semibold py-2 px-4 rounded">USAID</button>
    <button onclick="showProjects('JICA')" class="bg-gray-200 hover:bg-gray-400 text-gray-800 font-semibold py-2 px-4 rounded">JICA</button>
    <button onclick="showProjects('ADB')" class="bg-gray-200 hover:bg-gray-400 text-gray-800 font-semibold py-2 px-4 rounded">ADB</button>
    <button onclick="showProjects('WB')" class="bg-gray-200 hover:bg-gray-400 text-gray-800 font-semibold py-2 px-4 rounded">WB</button>
    <button onclick="showProjects('UKAID')" class="bg-gray-200 hover:bg-gray-400 text-gray-800 font-semibold py-2 px-4 rounded">UKAID</button>
    <button onclick="showProjects('SDC')" class="bg-gray-200 hover:bg-gray-400 text-gray-800 font-semibold py-2 px-4 rounded">SDC</button>
  </div>

  <!-- UNDP Projects -->
  <div id="UNDP" class="project-list hidden">
    <h1 class="text-2xl font-bold text-center text-gray-800 mb-4">UNDP Projects</h1>
    <div class="projects-container grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3 gap-4">
      <% for UNDPproject in UNDPprojects %>
        <div class="project bg-white p-6 rounded-lg shadow-md transition duration-300 ease-in-out">
          <form action="{% url 'save_project' %}" method="post" onsubmit="handleFormSubmit(this, event)" class="space-y-4">
            <% csrf_token %>
            <h2 class="text-lg font-semibold">{{ UNDPproject.Name }}</h2>
            <div>
              <a href="{{ UNDPproject.Link }}" target="_blank" class="text-blue-500 hover:text-blue-600 transition duration-300">View Project</a>
            </div>
            <button type="button" class="inline-block bg-green-500 hover:bg-green-600 text-white font-bold py-2 px-4 rounded focus:outline-none focus:shadow-outline">
              Save Project
            </button>
          </form>
        </div>
      <% endfor %>
    </div>
  </div>

```



```

  <!-- UNFAO Projects -->
  <div id="UNFAO" class="project-list hidden">
    <h1 class="text-2xl font-bold text-center text-gray-800 mb-4">UNFAO Projects</h1>
    <div class="projects-container grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3 gap-4">
      <% for UNFAOproject in UNFAOprojects %>
        <div class="project bg-white p-6 rounded-lg shadow-md transition duration-300 ease-in-out">
          <form action="{% url 'save_project' %}" method="post" onsubmit="handleFormSubmit(this, event)" class="space-y-3">
            <% csrf_token %>
            <h2 class="text-lg font-semibold">{{ UNFAOproject.Name }}</h2>
            <p class="text-gray-600">Duration: {{ UNFAOproject.Duration }}</p>
            <div>
              <a href="{{ UNFAOproject.Link }}" target="_blank" class="text-blue-500 hover:text-blue-600 transition duration-300">View Project</a>
            </div>
            <button type="button" class="bg-green-500 hover:bg-green-600 text-white font-bold py-2 px-4 rounded focus:outline-none focus:shadow-outline">
              Save Project
            </button>
          </form>
        </div>
      <% endfor %>
    </div>
  </div>

  <!-- USAID Projects -->
  <div id="USAID" class="project-list hidden">
    <h1 class="text-2xl font-bold text-center text-gray-800 mb-4">USAID Projects</h1>
    <div class="projects-container grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3 gap-4">
      <% for USAIDproject in USAIDprojects %>
        <div class="project bg-white p-6 rounded-lg shadow-md transition duration-300 ease-in-out">
          <form action="{% url 'save_project' %}" method="post" onsubmit="handleFormSubmit(this, event)" class="space-y-3">
            <% csrf_token %>
            <h2 class="text-lg font-semibold">{{ USAIDproject.Name }}</h2>
            <p class="text-gray-600">Duration: {{ USAIDproject.Duration }}</p>
            <p class="text-gray-600">Sector: {{ USAIDproject.Sector }}</p>
            <div class="flex justify-between items-center mt-4">
              <button type="button" class="bg-green-500 hover:bg-green-600 text-white font-bold py-2 px-4 rounded focus:outline-none focus:shadow-outline">
                Save Project
              </button>
            </div>
          </form>
        </div>
      <% endfor %>
    </div>
  </div>

```

Figure 207 HTML Page for Filtered Via Organizations Page

```

<!-- WB Projects -->
<div id="WB" class="project-list hidden">
  <h1 class="text-2xl font-bold text-center text-gray-800 mb-4">WB Projects</h1>
  <div class="projects-container grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3 gap-4">
    {% for WBproject in WBprojects %}
      <div class="project bg-white p-6 rounded-lg shadow-md transition duration-300 ease-in-out">
        <form action="{% url 'save_project' %}" method="post" onsubmit="handleFormSubmit(this, event)" class="space-y-3">
          |  {% csrf_token %}<br>
          |  <h2 class="text-lg font-semibold">{{ WBproject.Name }}</h2>
          |  <a href="{{ WBproject.Project_Link }}" target="_blank" class="text-blue-500 hover:text-blue-600 transition duration-300">View Project</a>
          |  <p class="text-gray-600">{{ WBproject.Details }}</p>
          |  <a href="{{ WBproject.Procurement_Link }}" target="_blank" class="text-blue-500 hover:text-blue-600 transition duration-300">Procurement Details</a>
          |  <p class="text-gray-600">Published: {{ WBproject.Published_Date }}</p>
          |  <button type="button" class="bg-green-500 hover:bg-green-600 text-white font-bold py-2 px-4 rounded focus:outline-none focus:shadow-outline transition duration-300 ease-in-out">Save Project</button>
        </form>
      </div>
    {% endfor %}
  </div>
</div>

<!-- ADB Projects -->
<div id="ADB" class="project-list hidden">
  <h1 class="text-2xl font-bold text-center text-gray-800 mb-4">ADB Projects</h1>
  <div class="projects-container grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3 gap-4">
    {% for ADBproject in ADBprojects %}
      <div class="project bg-white p-6 rounded-lg shadow-md transition duration-300 ease-in-out">
        <form action="{% url 'save_project' %}" method="post" onsubmit="handleFormSubmit(this, event)" class="space-y-3">
          |  {% csrf_token %}<br>
          |  <h2 class="text-lg font-semibold">{{ ADBproject.Name }}</h2>

          <a href="{{ ADBproject.Link }}" target="_blank" class="text-blue-500 hover:text-blue-600 transition duration-300">View Project</a>
          <p class="text-gray-600">Status: {{ ADBproject.Status }}</p>
          <p class="text-gray-600">Description: {{ ADBproject.Desc }}</p>
          <button type="button" class="bg-green-500 hover:bg-green-600 text-white font-bold py-2 px-4 rounded focus:outline-none focus:shadow-outline transition duration-300 ease-in-out">Edit Project</button>
        </form>
      {% endfor %}
    </div>
  </div>
</div>

<!-- JICA Projects -->
<div id="JICA" class="project-list hidden">
  <h1 class="text-2xl font-bold text-center text-gray-800 mb-4">JICA Projects</h1>
  <div class="projects-container grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3 gap-4">
    {% for JICApject in JICApjcts %}
      <div class="project bg-white p-6 rounded-lg shadow-md transition duration-300 ease-in-out">
        <form action="{% url 'save_project' %}" method="post" onsubmit="handleFormSubmit(this, event)" class="space-y-3">
          |  {% csrf_token %}<br>
          |  <h2 class="text-lg font-semibold">{{ JICApject.Name }}</h2>

          <a href="{{ JICApject.Link }}" target="_blank" class="text-blue-500 hover:text-blue-600 transition duration-300">View Project</a>
          <button type="button" class="bg-green-500 hover:bg-green-600 text-white font-bold py-2 px-4 rounded focus:outline-none focus:shadow-outline transition duration-300 ease-in-out">Edit Project</button>
        </form>
      {% endfor %}
    </div>
  </div>
</div>

<div id="UKAID" class="project-list hidden">
  <h1 class="text-2xl font-bold text-center text-gray-800 mb-4">UKAID Projects</h1>
  <div class="projects-container grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3 gap-4">
    {% for UKAIDpjct in UKAIDpjcts %}
      <div class="project bg-white p-6 rounded-lg shadow-md transition duration-300 ease-in-out">
        <form action="{% url 'save_project' %}" method="post" onsubmit="handleFormSubmit(this, event)" class="space-y-3">
          |  {% csrf_token %}<br>
          |  <h2 class="text-lg font-semibold">{{ UKAIDpjct.Name }}</h2>

          <a href="{{ UKAIDpjct.Link }}" target="_blank" class="text-blue-500 hover:text-blue-600 transition duration-300">View Project</a>
          <p class="text-gray-600">Project Details: {{ UKAIDpjct.Desc }}</p>
          <p class="text-gray-600">Start Date: {{ UKAIDpjct.Date }}</p>

          <button type="button" class="bg-green-500 hover:bg-green-600 text-white font-bold py-2 px-4 rounded focus:outline-none focus:shadow-outline transition duration-300 ease-in-out">Edit Project</button>
        </form>
      {% endfor %}
    </div>
  </div>
</div>

```

Figure 208 HTML Page for Filtered Via Organizations Page

```

<div id="SDC" class="project-list hidden">
  <h1 class="text-2xl font-bold text-center text-gray-800 mb-4">SDC Projects</h1>
  <div class="projects-container grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3 gap-4">
    {% for SDCproject in SDCprojects %}
      <div class="project bg-white p-6 rounded-lg shadow-md transition duration-300 ease-in-out">
        <form action="{{ url 'save_project' }}" method="post" onsubmit="handleFormSubmit(this, event)" class="space-y-3">
          |  {% csrf_token %}
          <h2 class="text-lg font-semibold">{{ SDCproject.Name }}</h2>

          <a href="{{ SDCproject.Link }}" target="_blank" class="text-blue-500 hover:text-blue-600 transition duration-300">View Project</a>
          <p class="text-gray-600">Project Details: {{ SDCproject.Desc }}</p>
          <p class="text-gray-600">Duration: {{ SDCproject.Duration }}</p>

          <button type="button" class="bg-green-500 hover:bg-green-600 text-white font-bold py-2 px-4 rounded focus:outline-none focus:shadow-sm">Edit</button>
        </div>
      {% endfor %}
    </div>
  </div>

```

Figure 209 HTML Page for Filtered Via Organizations Page

HTML Pages:

- They contain the markup for a series of buttons corresponding to different organizations like UNDP, UNFAO, USAID, etc.
- Each button has an onclick event that calls the showProjects JavaScript function with an identifier as an argument.
- There are sections for each organization's projects, initially hidden, that include a list of projects. Each project has a title, duration, and a link to view more details.
- Each project can be saved using a form submission with CSRF protection.

```
function showProjects(id) {
    // Hide all project lists
    var projectLists = document.getElementsByClassName("project-list");
    for (var i = 0; i < projectLists.length; i++) {
        projectLists[i].classList.add("hidden");
    }

    // Show the selected project list
    var selectedProject = document.getElementById(id);
    if (selectedProject) {
        selectedProject.classList.remove("hidden");
    }
}
```

Figure 210 JS to toggle Organization Buttons

JavaScript Function:

- This function is designed to show the project list for the selected organization.
- It first hides all project lists and then reveals the one corresponding to the clicked button.

```

#fetching data from MongoDB
#UNDP
UNDP = db['UNDP']
UNFAO = db['UNFAO']
USAID = db['USAID']
WB = db['WB']
ADB = db['ADB']
JICA = db['JICA']
UKAID = db['UKAID']
SDC = db['SDC']

from django.contrib.auth.decorators import login_required

@login_required
def display_data(request):

    # Fetch data from the collection
    UNDPprojects = list(UNDP.find({}, {'_id': 0}))

    for UNDPproject in UNDPprojects:
        UNDPproject['Name'] = UNDPproject.pop('Name of the Project')
        UNDPproject['Link'] = UNDPproject.pop('Link to the project')
    pass

    # Fetch data from the collection
    UNFAOprojects = list(UNFAO.find({}, {'_id': 0}))
    for UNFAOproject in UNFAOprojects:
        UNFAOproject['Name'] = UNFAOproject.pop('Name of the Project')
        UNFAOproject['Duration'] = UNFAOproject.pop('Duration of the Project')
        UNFAOproject['Link'] = UNFAOproject.pop('Link to the project')
    pass

    # Fetch data from the collection
    USAIDprojects = list(USAID.find({}, {'_id': 0}))
    for USAIDproject in USAIDprojects:
        USAIDproject['Name'] = USAIDproject.pop('Name of the Project')
        USAIDproject['Duration'] = USAIDproject.pop('Duration of the Project')
        USAIDproject['Sector'] = USAIDproject.pop('Sector')
    pass

    # Fetch data from the collection
    WBprojects = list(WB.find({}, {'_id': 0}))
    for WBproject in WBprojects:
        WBproject['Name'] = WBproject.pop('Name of the Project')
        WBproject['Project_Link'] = WBproject.pop('Link to the project')
        WBproject['Details'] = WBproject.pop('Procurement Details')
        WBproject['Procurement_Link'] = WBproject.pop('Link to the Procurement')
        WBproject['Published_Date'] = WBproject.pop('Date of Publishment')
    pass

    # Fetch data from the collection
    ADBprojects = list(ADB.find({}, {'_id': 0}))
    for ADBproject in ADBprojects:
        ADBproject['Name'] = ADBproject.pop('Name of the Project')
        ADBproject['Link'] = ADBproject.pop('Link to the project')
        ADBproject['Status'] = ADBproject.pop('Project Status')
        ADBproject['Desc'] = ADBproject.pop('Project Description')
    pass

```

Figure 211 Backend Function for Filter Via Organization Page

```

# Fetch data from the collection
JICAprojects = list(JICA.find({}, {'_id': 0}))
for JICaproject in JICAprojects:
    JICaproject['Name'] = JICaproject.pop('Name of the Project')
    JICaproject['Link'] = JICaproject.pop('Link to the project')
pass
# Fetch data from the collection
UKAIDprojects = list(UKAID.find({}, {'_id': 0}))
for UKAIDproject in UKAIDprojects:
    UKAIDproject['Name'] = UKAIDproject.pop('Name of the Project')
    UKAIDproject['Link'] = UKAIDproject.pop('Link to the project')
    UKAIDproject['Date'] = UKAIDproject.pop('Start Date')
    UKAIDproject['Desc'] = UKAIDproject.pop('Project Details')
pass
# Fetch data from the collection
SDCprojects = list(SDC.find({}, {'_id': 0}))
for SDCproject in SDCprojects:
    SDCproject['Name'] = SDCproject.pop('Name of the Project')
    SDCproject['Link'] = SDCproject.pop('Link to the project')
    SDCproject['Duration'] = SDCproject.pop('Duration of the Project')
    SDCproject['Desc'] = SDCproject.pop('Project Details')
pass
# Pass data to the template
context = {'UNFAOprojects': UNFAOprojects,
           'UNDPprojects': UNDPprojects,
           'USAIDprojects': USAIDprojects,
           'WBprojects': WBprojects,
           'ADBprojects': ADBprojects,
           'JICAprojects': JICAprojects,
           'UKAIDprojects': UKAIDprojects,
           'SDCprojects': SDCprojects,
           }
}

return render(request, 'List.html', context)

```

Figure 212 Backend Function for Filter Via Organization Page

Backend View Functions:

- These functions are responsible for fetching data from MongoDB collections corresponding to each organization.
- The data for each project is prepared by fetching and modifying the project entries from the collections, such as adding names, links, and other details.
- The prepared data is then passed to the Django template via the context.

8.3.1.3.1 Save Project

Recovery & Resilience

[View Project](#)

Save Project

Figure 213 Save Project Button

Recovery & Resilience

[View Project](#)

Project has been saved successfully.

Save Project

Figure 214 Save Project Button in Action

```
<button type="button" class="bg-green-500 hover:bg-green-600 text-white font-bold py-2 px-4 rounded focus:outline-none focus:shadow-outline transition duration-300" onclick="handleFormSubmit(this);>Save Project</button>
```

Figure 215 HTML & CSS code for "Save Project Button."

The button in the above snippet is designed with a class for styling and an onclick event that triggers a JavaScript function handleFormSubmit() when clicked.

```

function handleFormSubmit(clickedButton) {
  var projectDiv = clickedButton.closest('.project');
  if (!projectDiv) {
    console.error('Project div not found');
    return;
  }

  var projectDetailsHTML = projectDiv.innerHTML;

  var form = document.createElement('form');
  form.setAttribute('method', 'post');
  form.setAttribute('action', '{% url 'save_project' %}');

  var csrfInput = document.createElement('input');
  csrfInput.setAttribute('type', 'hidden');
  csrfInput.setAttribute('name', 'csrfmiddlewaretoken');
  csrfInput.setAttribute('value', document.querySelector('[name="csrfmiddlewaretoken"]').value);
  form.appendChild(csrfInput);

  var contentInput = document.createElement('input');
  contentInput.setAttribute('type', 'hidden');
  contentInput.setAttribute('name', 'content');
  contentInput.setAttribute('value', projectDetailsHTML);
  form.appendChild(contentInput);

  document.body.appendChild(form); // Add form to the DOM for submission

  var formData = new FormData(form);

  fetch(form.action, {
    method: 'POST',
    body: formData,
    credentials: 'same-origin'
  })
  .then(response => {
    if (!response.ok) {
      throw new Error('Network response was not ok: ' + response.statusText);
    }
    return response.json();
  })
  .then(data => {
    if (data.success) {
      displaySuccessMessage(clickedButton, "Project has been saved successfully.");
    }
  })
  .catch(error => {
    console.error('Error:', error);
  })
  .finally(() => {
    document.body.removeChild(form); // Ensure form is removed after operation
  });
}

```

Figure 216 JS to handle Form Submission

JavaScript for Form Submission:

- The JavaScript function creates a dynamic form when the 'Save Project' button is clicked.
- It first finds the closest project container and retrieves its HTML content.
- A new form is created with a hidden CSRF token and hidden input containing the project details.
- The form is appended to the DOM and submitted using the Fetch API with the appropriate credentials and POST method.

```

function displaySuccessMessage(button, message) {
    // Create a message element
    var messageDiv = document.createElement('div');
    messageDiv.classList.add(
        'success-message',
        'bg-green-100',
        'border',
        'border-green-400',
        'text-green-700',
        'px-4',
        'py-3',
        'rounded',
        'relative',
        'my-3',
        'transition',
        'duration-500',
        'ease-in-out',
        'transform',
        'opacity-0'
    );
    messageDiv.textContent = message;

    // Insert the message in the DOM
    button.parentNode.insertBefore(messageDiv, button);

    // Fade in the message
    setTimeout(() => {
        messageDiv.classList.remove('opacity-0');
        messageDiv.classList.add('opacity-100');
    }, 100); // Fades in the message

    // Fade out the message after some time
    setTimeout(() => {
        messageDiv.classList.remove('opacity-100');
        messageDiv.classList.add('opacity-0');
    }, 2900); // Starts fading out the message after 2.9 seconds

    // Remove the message after it fades out
    setTimeout(() => {
        messageDiv.remove();
    }, 5000);
}

```

Figure 217 JS to handle Confirmation Message

JavaScript for Confirmation Message:

- This function creates a message element and inserts it into the DOM before the button that was clicked.
- It displays a success message with an animation that fades in and out, indicating that the project has been saved.

```
# save_projects
from django.http import JsonResponse
from django.views.decorators.http import require_POST
from .models import Project
from django.views.decorators.csrf import csrf_exempt

@csrf_exempt
@require_POST
def save_project(request):
    content = request.POST.get('content')
    user = request.user

    # Create a new Project instance and save the form data
    try:
        project = Project(content=content, user=user)
        project.save()
        return JsonResponse({"success": True, "message": "Project saved successfully."})
    except Exception as e:
        # Log the error or handle it as per requirements
        return JsonResponse({"success": False, "message": "Failed to save the project."}, status=400)
```

Figure 218 Backend Function to Save Projects.

Backend View Function:

- The view function in Django is marked with `@csrf_exempt` to bypass CSRF verification, which is not recommended for production environments due to security concerns.
- It accepts only POST requests and extracts the content from the request to create a new Project instance associated with the current user.
- If the project is saved successfully, it returns a JSON response indicating success; otherwise, it catches any exceptions and returns a JSON response indicating failure.

8.3.1.1.4 Filter Via Keywords

The screenshot shows the SCRAPEQUEST interface. At the top, there is a search bar and a user greeting "Hi, sharams! Logout". Below the header, there are three main navigation tabs: "Filter via Organizations", "Filter via Keywords" (which is highlighted with a red box), and "Saved Projects". Underneath these tabs is a grid of filter categories represented by buttons. The "Food" button is also highlighted with a red box. Other buttons include Food Aid, Food system, Food security, Agriculture, Digital, Food Science, Food Loss, Education, Migration, Nutrition, Health, Climate Change, Livelihood, and Disaster Risk Reduction. Below the filter buttons are two application type buttons: "Web Application" and "Mobile Application". The main content area displays a heading "Showing Results For Food" (also highlighted with a red box). Three project cards are shown, each with a "Save Project" button at the bottom. The first project card contains the following details:

Name of the Project: Food and Nutrition Security Enhancement Project - P164319
Link to the project: [View](#)
Procurement Details: Procurement of improved maize seed.
Link to the Procurement: [View](#)
Date of Publication: March 4, 2024
Organization: WB
Save Project

The second project card has similar details:

Name of the Project: Food and Nutrition Security Enhancement Project - P164319
Link to the project: [View](#)
Procurement Details: Reconstruction and Maintenance of office rooms & Program hall
Link to the Procurement: [View](#)
Date of Publication: February 28, 2024
Organization: WB
Save Project

The third project card has similar details:

Name of the Project: Food and Nutrition Security Enhancement Project - P164319
Link to the project: [View](#)
Procurement Details: Procurement of Seasonable Summer Vegetables Seed (PCU Gorkha)
Link to the Procurement: [View](#)
Date of Publication: February 28, 2024
Organization: WB
Save Project

Below these cards, there is a section titled "Related Results For Food" containing one project card:

Name of the Project: GCP /GLO/505/ROK: Implementation of Codex standards to support containment and reduction of foodborne AMR
Duration of the Project: 20-Jul-2021 - 19-Jun-2026
Link to the project: [View](#)
Organization: UNFAO
Save Project

Figure 219 Page to Display Projects Filtered Via Keywords

```

<div id="keyword-buttons-container" class="flex justify-center py-6 mb-4 w-full">
  <div class="flex justify-center flex-wrap gap-6">
    {% for keyword in results.keys %}
      <button
        class="bg-gray-200 hover:bg-gray-300 text-gray-800 font-semibold py-2 px-4 rounded shadow"
        onclick="showProjects('{{ keyword | slugify }}')"
      >
        {{ keyword }}
      </button>
    {% endfor %}
  </div>
</div>

<div class="flex justify-center">
  <div class="w-full max-w-6xl px-4">
    {% for keyword, matches in results.items %}
      <div id="{{ keyword | slugify }}-full" class="project-list hidden">
        <h2 class="text-3xl font-bold text-center text-gray-800 mb-8"> Showing Results For {{ keyword }}</h2>

        <div class="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3 gap-6">
          {% for project in matches.full_matches %}
            <div class="project bg-white shadow-lg rounded-lg overflow-hidden">
              <form action="{{ url 'save_project' }}" method="post" onsubmit="handleFormSubmit(this.querySelector('.save-project-btn'), event)" class="p-6">
                {% csrf_token %}

                <h3 class="text-xl font-semibold mb-4">{{ project.Name_of_the_Project }}</h3>

                {% for field, value in project.items %}
                  {% if field != 'Name_of_the_Project' and field != 'Date' and value != 'N/A' %}
                    {% if field == 'Link to the project' or field == 'Link to the Procurement' %}
                      <p class="mb-3"><span class="font-semibold">{{ field }}</span> <a href="{{ value }}" target="_blank" class="text-blue-600 hover:text-blue-800">View</a></p>
                      <input type="hidden" name="link" value="{{ project.link_to_the_project }}">
                    {% else %}
                      <p class="mb-3"><span class="font-semibold">{{ field }}</span> {{ value }}</p>
                    {% endif %}
                  {% endif %}
                  {% endif %}
                <button type="button" class="save-project-btn bg-green-500 hover:bg-green-600 text-white font-bold py-2 px-4 rounded" onclick="handleFormSubmit(this);">Save Project</button>
              </form>
            </div>
          {% endfor %}
        </div>
      </div>
    {% endfor %}
  </div>

```

Figure 220 HTML Page to Display Projects Filtered Via Keywords (Full Matches)

```

<div id="{{ keyword | slugify }}-partial" class="project-list hidden">
  <div class="w-full border-t border-gray-300"></div>

  <h2 class="pt-10 text-3xl font-bold text-center text-gray-800 mb-8">Related Results For {{ keyword }}</h2>

  <div class="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3 gap-6">
    {% for project in matches.partial_matches %}
      <div class="project bg-white shadow-lg rounded-lg overflow-hidden">
        <form action="{{ url 'save_project' }}" method="post" onsubmit="handleFormSubmit(this.querySelector('.save-project-btn'), event)" class="flex flex-col justify-between h-full p-6">
          {% csrf_token %}

          <div>
            <h3 class="text-xl font-semibold mb-4">{{ project.Name_of_the_Project }}</h3>

            {% for field, value in project.items %}
              {% if field != 'Name_of_the_Project' and field != 'Date' and value != 'N/A' %}
                {% if field == 'Link to the project' or field == 'Link to the Procurement' %}
                  <p class="mb-3"><span class="font-semibold">{{ field }}</span> <a href="{{ value }}" target="_blank" class="text-blue-600 hover:text-blue-800">View</a></p>
                  <input type="hidden" name="link" value="{{ project.link_to_the_project }}">
                {% else %}
                  <p class="mb-3"><span class="font-semibold">{{ field }}</span> {{ value }}</p>
                {% endif %}
              {% endif %}
              {% endif %}
            </div>

            <button type="button" class="save-project-btn mt-4 bg-green-500 hover:bg-green-600 text-white font-bold py-2 px-4 rounded self-end" onclick="handleFormSubmit(this);">Save Project</button>
          </form>
        </div>
      </div>
    {% endfor %}
  </div>
<% endfor %>

```

Figure 221 HTML Page to Display Projects Filtered Via Keywords (Partial Matches)

HTML Pages:

- The HTML includes buttons for each keyword. When a button is clicked, a JavaScript function (showProjects) is triggered to display projects associated with that keyword.
- For each keyword, there's a container to hold the list of projects. This container is initially hidden and is displayed when the corresponding keyword button is clicked.
- Each project entry has a form with a CSRF token for security that allows users to save the project. The form submission is handled by a JavaScript function (handleFormSubmit).

```

document.addEventListener('DOMContentLoaded', (event) => {
  let buttons = document.querySelectorAll("[id^='keyword-button-']");
  buttons.forEach(button => {
    button.addEventListener('click', function() {
      showProjects(this.getAttribute('data-target'));
    });
  });
}

function showProjects(baseId) {
  // Hide all project lists
  var projectLists = document.getElementsByClassName('project-list');
  for (var i = 0; i < projectLists.length; i++) {
    projectLists[i].classList.add('hidden');
  }

  // Show the selected full and partial project lists
  var selectedFullProject = document.getElementById(baseId + "-full");
  var selectedPartialProject = document.getElementById(baseId + "-partial");

  if (selectedFullProject) {
    selectedFullProject.classList.remove('hidden');
  }
  if (selectedPartialProject) {
    selectedPartialProject.classList.remove('hidden');
  }
}

```

Figure 222 JS function to toggle Keywords buttons.

JavaScript to Toggle Keyword Buttons:

- Upon DOM content being fully loaded, an event listener is attached to each keyword button. Clicking a keyword button triggers the showProjects function.
- The showProjects function hides all project lists and then shows the selected keyword's full and partial project lists based on their IDs.

```

collections = ['UNDP', 'UNFAO', 'USAID', 'WB', 'ADB', 'JICA', 'SDC', 'UKAID']

organization_fields = {
    'UNDP': ['Name of the Project', 'Link to the project'],
    'UNFAO': ['Name of the Project', 'Duration of the Project', 'Link to the project'],
    'USAID': ['Name of the Project', 'Duration of the Project', 'Sector'],
    'WB': ['Name of the Project', 'Link to the project', 'Procurement Details', 'Link to the Procurement'],
    'ADB': ['Name of the Project', 'Link to the project', 'Project Status', 'Project Description'],
    'JICA': ['Name of the Project', 'Link to the project'],
    'SDC': ['Name of the Project', 'Link to the project', 'Duration of the Project', 'Project Details'],
    'UKAID': ['Name of the Project', 'Link to the project', 'Project Details', 'Start Date'],
}

def parse_date(project, organization):
    date_str = None
    formats = [] # Initialize formats to an empty list

    if organization == 'UNFAO':
        # Extract the start date from the duration using " - " as the separator
        duration = project.get('Duration of the Project', '')
        if " - " in duration:
            date_str = duration.split(" - ")[0]
        formats = ['%d-%b-%Y', '%b %Y'] # 20-Jun-2022, Jul 2019
    elif organization == 'USAID':
        # Check if the duration field is not empty and use " - " as the separator
        duration = project.get('Duration of the Project', '')
        if duration and " - " in duration:
            date_str = duration.split(" - ")[0]
        formats = ['%Y-%m-%d'] # 2006-09-29
    elif organization == 'WB':
        date_str = project.get('Date of Publication', '')
        formats = ['%B %d, %Y'] # March 12, 2023
    elif organization == 'SDC':
        date_str = project.get('Duration of the Project', '')
        formats = ['%d.%m.%Y'] # 31.12.2026
    elif organization == 'UKAID':
        date_str = project.get('Start Date', '')
        formats = ['%Y-%m-%d'] #2023-4-1
    elif organization == 'ADB':
        # Extract the date after "Posting date:"
        description = project.get('Project Description', '')
        marker = "Posting date:"
        if marker in description:
            start_index = description.find(marker) + len(marker)
            end_index = description.find(".", start_index)
            if end_index > start_index:
                date_str = description[start_index:end_index].strip()
        formats = ['%d %b %Y'] # 10 Aug 2022

    for fmt in formats:
        if date_str: # Ensure date_str is not None or empty
            try:
                return datetime.datetime.strptime(date_str, fmt)
            except ValueError:
                continue

    return None

```

Figure 223 Backend Function for Filter Via Keywords Page

```

def sort_projects(projects):
    # This function will place None values at the end when sorting in descending order
    return sorted(projects, key=lambda x: (x['Date'] is not None, x['Date']), reverse=True)

from django.contrib.auth.decorators import login_required
from django.conf import settings
import json

@login_required
def list_projects_by_keyword(request):
    # Load keywords from the configuration file
    keywords = load_keywords()

    # Handling requests
    keyword = request.GET.get('keyword', '') # Get the keyword from the request
    results = {}

    # Logic for fetching and organizing projects
    results = {keyword: {'full_matches': [], 'partial_matches': []} for keyword in keywords}

    for organization, fields in organization_fields.items():
        collection = db[organization]
        for keyword in keywords:
            # Define additional search fields based on the organization
            search_fields = ['Name of the Project']
            if organization == 'WB':
                search_fields.append('Procurement Details')
            elif organization == 'ADB':
                search_fields.append('Project Description')
            elif organization == 'SDC':
                search_fields.append('Project Details')
            elif organization == 'UKAID':
                search_fields.append('Project Details')

            # Perform searches on all designated fields
            for field in search_fields:
                full_keyword_query = {field: {"$regex": f"\b{keyword}\b", "$options": "i"}}
                projects = collection.find(full_keyword_query)

                for project in projects:
                    project_info = {field: project.get(field, 'N/A') for field in fields}
                    project_info['Organization'] = organization
                    project_info['Date'] = parse_date(project, organization) # Parse the date

                    if project_info not in results[keyword]['full_matches']:
                        results[keyword]['full_matches'].append(project_info)

            # Partial keyword search
            for word in keyword.split():
                partial_keyword_query = {field: {"$regex": f"(?={word})", "$options": "i"}}
                projects = collection.find(partial_keyword_query)

                for project in projects:
                    project_info = {field: project.get(field, 'N/A') for field in fields}
                    project_info['Organization'] = organization
                    project_info['Date'] = parse_date(project, organization) # Parse the date

                    if project_info not in results[keyword]['full_matches'] and project_info not in results[keyword]['partial_matches']:
                        results[keyword]['partial_matches'].append(project_info)

    for keyword in results:
        results[keyword]['full_matches'] = sort_projects(results[keyword]['full_matches'])
        results[keyword]['partial_matches'] = sort_projects(results[keyword]['partial_matches'])

    return render(request, 'project_details.html', {'results': results})

def load_keywords():
    with open(settings.BASE_DIR / 'config/keywords_config.json', 'r') as file:
        return json.load(file)

```

Figure 224 Backend Function for Filter Via Keywords Page

Backend View Function:

- The view function `list_projects_by_keyword` is responsible for fetching and displaying projects related to a specific keyword. It is a protected view that requires user login.
- The function reads the keyword from the request parameters and uses it to fetch relevant project data from a MongoDB database.
- It employs a function `parse_date` to standardize date formats across different organizations.
- Projects are then organized into full and partial matches based on the search terms, sorted, and passed into the context for rendering in the Django template.

8.3.1.1.4.1 Save Project Button

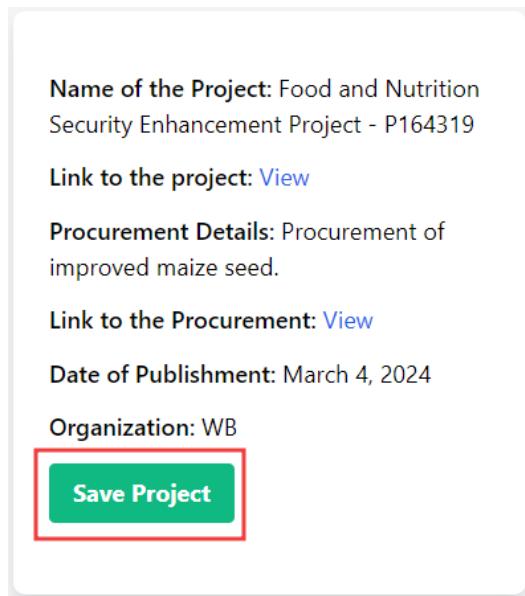


Figure 225 Save project Button.

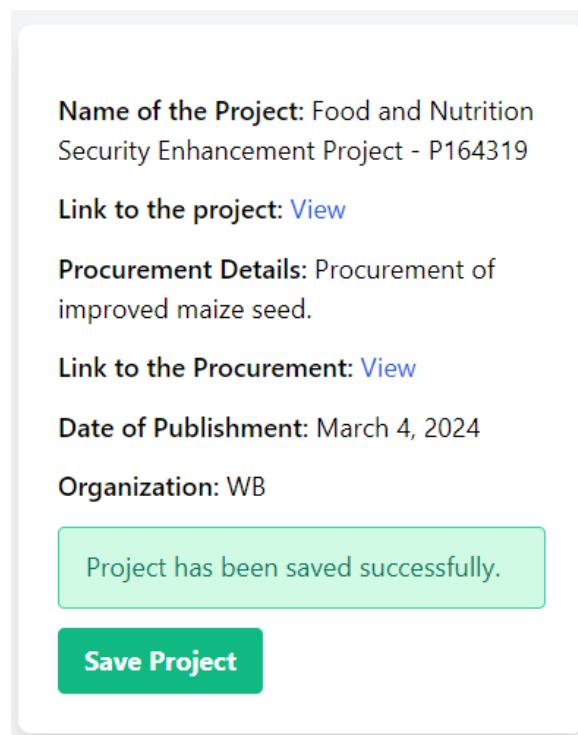


Figure 226 Save Project Button in Action

```

function handleFormSubmit(clickedButton) {
  var projectDiv = clickedButton.closest('.project');

  // Serialize the details from the project div
  var projectDetailsHTML = projectDiv.innerHTML;

  var form = document.createElement('form');
  form.setAttribute('method', 'post');
  form.setAttribute('action', '{% url 'save_project' %}');

  // Add the CSRF token and project details to the form
  var csrfInput = document.createElement('input');
  csrfInput.setAttribute('type', 'hidden');
  csrfInput.setAttribute('name', 'csrfmiddlewaretoken');
  csrfInput.setAttribute('value', document.querySelector('[name="csrfmiddlewaretoken"]').value);
  form.appendChild(csrfInput);

  var contentInput = document.createElement('input');
  contentInput.setAttribute('type', 'hidden');
  contentInput.setAttribute('name', 'content');
  contentInput.setAttribute('value', projectDetailsHTML);
  form.appendChild(contentInput);

  document.body.appendChild(form);

  // Create a FormData object and append the content
  var formData = new FormData(form);

  // Make the fetch API call to send the POST request
  fetch(form.action, {
    method: 'POST',
    body: formData,
    credentials: 'same-origin' // Include cookies and CSRF token
  })
  .then(response => {
    document.body.removeChild(form); // Clean up: remove the temporary form

    if (!response.ok) {
      throw new Error(`Network response was not ok: ${response.statusText}`);
    }
    return response.json(); // Or response.text() if not returning JSON
  })
  .then(data => {
    if(data.success) {
      displaySuccessMessage(clickedButton, "Project has been saved successfully.");
    }
  })
  .catch(error => {
    console.error('Error:', error);
    // Handle the error, possibly notify the user that the save action failed
  });
}

```

Figure 227 JS to handle form submission.

JavaScript for Form Submission:

- The JavaScript function creates a dynamic form when the 'Save Project' button is clicked.
- It first finds the closest project container and retrieves its HTML content.
- A new form is created with a hidden CSRF token and hidden input containing the project details.
- The form is appended to the DOM and submitted using the Fetch API with the appropriate credentials and POST method.

8.3.1.1.5 Saved Projects

SCRAPEQUEST

Search...

Hi, sharams! Logout

Filter via Organizations Filter via Keywords **Saved Projects**

Saved Projects

Recovery & Resilience
[View Project](#)

MCC Transmission Lines Activity
Duration: [redacted]
Sector: Energy

Name of the Project:
UNJP/NEP/084/UNJ Assessing the impact of the global crisis on the agriculture and food security situation in Nepal
Duration of the Project: 20-Jun-2022 - 31-Mar-2023
Link to the project: [View](#)
Organization: UNFAO

MiRiDew - Migrant Rights and Decent Work
[View Project](#)

Project Details: Remittances sent home by over 4 million migrant workers have significantly contributed to the economic development of Nepal. The proposed project will enhance the capacities of the Government of Nepal and strengthen mechanisms to better protect the rights of workers abroad. There will be a special focus on women's need and the adverse effects of climate change exacerbating the vulnerability of migrants. The project capitalises on Switzerland's longstanding engagement on labour migration in Nepal.

Duration: 01.07.2023 - 31.12.2026

2021 Population and Housing Census of Nepal
[View Project](#)

Project Details: The goal is to support the Government of Nepal to conduct the 2021 census in line with international standards to obtain high quality data, reflecting Nepal's social, ethnic, religious and linguistic diversity and which can be used to better respond to the needs of the different population groups. Given its track record and emphasis on federal state building, Switzerland has a key interest that the census is adapted to the federal context and gender equality and social inclusion principles are followed.

Duration: 01.07.2021 - 31.12.2024

Fleming Fund – Country and Regional Grants and Fellowships Programme
[View Project](#)

Project Details: The Fleming Fund helps low- and middle-income countries to fight antimicrobial resistance. A management agent has been appointed to deliver: country grants 24 low- and middle-income countries, regional grants in West Africa, East and Southern Africa, South Asia and South East Asia, and global fellowships programme. These initiatives aim to improve laboratory capacity and diagnosis as well as data and surveillance of antimicrobial resistance (AMR).

Start Date: 2016-10-10

Figure 228 Saved Projects Page.

```

{% block content %}

{%load static%}
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/tailwindcss@2.2.19/dist/tailwind.min.css" />

<div class="container mx-auto px-4 py-8">
    <h1 class="text-4xl font-bold text-center text-gray-800 mb-10">Saved Projects</h1>

    <div class="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3 gap-6">
        {% for project in projects %}
            <div class="project bg-white p-6 rounded-lg shadow-lg overflow-hidden">
                <div class="project-content text-gray-700 space-y-2">
                    <!-- The 'safe' filter will render the HTML -->
                    {{ project.content|safe }}
                </div>
            </div>
        {% endfor %}
    </div>
</div>

<script>
    document.addEventListener('DOMContentLoaded', (event) => {
        document.querySelectorAll('.save-project-btn').forEach(button => button.remove());
    });
</script>
{% endblock %}

```

Figure 229 HTML Page for Saved Projects

The HTML page features a content block for inserting HTML into a base template, using Tailwind CSS for styling. It has a grid layout with each project displayed as a card, adjusting for mobile and larger screens. Each content is marked safe to render, ensuring accurate display of database content.

```

@login_required
def view_saved_projects(request):
    projects = Project.objects.filter(user=request.user) # Fetch all projects from the database
    return render(request, 'view_saved_projects.html', {'projects': projects})

```

Figure 230 Backend Function Display Saved Projects

Backend Function (view_saved_projects):

- The function retrieves all Project objects from the database that belong to the logged-in user (request.user).
- It then passes these projects to the template view_saved_projects.html for rendering.

8.3.1.2 Admin Panel

The screenshot shows the Django Admin Panel interface. At the top, there's a header bar with "Django administration" on the left and "WELCOME, ADMIN. VIEW SITE / CHANGE PASSWORD / LOG OUT" on the right. Below the header, the main content area is titled "Site administration". It has two main sections: "AUTHENTICATION AND AUTHORIZATION" and "MYAPP_LIST".

AUTHENTICATION AND AUTHORIZATION:

- Groups**: + Add, Change
- Users**: + Add, Change

MYAPP_LIST:

- Keyword Uploads**: + Add, Change
- Projects**: + Add, Change

On the right side of the interface, there are two sidebar panels:

- Recent actions**: A list of recent operations, including additions and changes for users, admin, house, and sharams1.
- My actions**: A list of recent operations, including additions and changes for user, admin, house, sharams1, and sharams.

Figure 231 Admin Panel

8.3.1.2.1 Manage Users

Django administration

Home > Authentication and Authorization > Users > sharams

Start typing to filter...

AUTHENTICATION AND AUTHORIZATION

- Groups [+ Add](#)
- Users** [+ Add](#) (highlighted)

MYAPP_LIST

- Keyword Uploads [+ Add](#)
- Projects [+ Add](#)

Change user

sharams

Change Username

Username: sharams (highlighted)

Required: 150 characters or fewer. Letters, digits and @/./+/_- only.

Password:

algorithim: pbkdf2_sha256 **iterations:** 720000 **salt:** 2BQP3***** **hash:** chHips*****
Raw passwords are not stored, so there is no way to see this user's password, but you can change the password using this form.

Personal info

First name: (highlighted)

Last name: (highlighted)

Email address: sharamskunwar.sk@gmail.com

Modify Permissions, Privilege Management

Permissions

Active
Designates whether this user should be treated as active. Unselect this instead of deleting accounts.

Staff status
Designates whether the user can log into this admin site.

Superuser status
Designates that this user has all permissions without explicitly assigning them.

Figure 232 Admin Interface to Manage Users

8.3.1.2.2 Modify Keywords

Home

Start typing to filter...

AUTHENTICATION AND AUTHORIZATION

- Groups [+ Add](#)
- Users** [+ Add](#)

MYAPP_LIST

- Keyword Uploads** [+ Add](#) (highlighted)
- Projects [+ Add](#)

Edit Keywords

Keywords: Food, Food Aid, Food system, Food security, Agriculture, Digital, Food Science, Food Loss, Education, Migration, Nutrition, Health, Climate Change, Livelihood, Disaster Risk Reduction, Web Application, Mobile Application

Enter keywords separated by commas

Save Changes

Figure 233 Admin Panel to Modify Keywords

```
##keywords
from django import forms

class KeywordsForm(forms.Form):
    keywords = forms.CharField(widget=forms.Textarea, help_text='Enter keywords separated by commas')

    def __init__(self, *args, **kwargs):
        keywords = kwargs.pop('keywords', [])
        super().__init__(*args, **kwargs)
        self.fields['keywords'].initial = ', '.join(keywords)
```

Figure 234 Form to Modify Keywords

Form to Modify Keywords:

- The form's constructor (`__init__` method) has been overridden to accept an additional `keywords` argument, which is a list of keyword strings.
- Inside the constructor, it removes the `keywords` from the keyword arguments (`kwargs`) and sets the initial value of the `keywords` field to a comma-separated string made from the `keywords` list.

```
##keywords
from django.shortcuts import render
from django.http import HttpResponseRedirect
from .forms import KeywordsForm
from .models import KeywordUploadProxy
from django.urls import path
from django.conf import settings
import json
import os

@admin.register(KeywordUploadProxy)
class KeywordsConfigAdmin(admin.ModelAdmin):
    change_list_template = "admin/keywords_config_change_list.html"

    def get_urls(self):
        urls = super().get_urls()
        my_urls = [
            path('edit-keywords/', self.admin_site.admin_view(self.edit_keywords), name='edit_keywords'),
        ]
        return my_urls + urls

    def edit_keywords(self, request):
        # Load keywords from JSON file
        file_path = os.path.join(settings.BASE_DIR, 'config', 'keywords_config.json')
        with open(file_path, 'r') as file:
            keywords = json.load(file)

        if request.method == 'POST':
            form = KeywordsForm(request.POST, keywords=keywords)
            if form.is_valid():
                # Save updated keywords to JSON file
                updated_keywords = [kw.strip() for kw in form.cleaned_data['keywords'].split(',')]
                with open(file_path, 'w') as file:
                    json.dump(updated_keywords, file)
                self.message_user(request, 'Keywords updated successfully.')
                return HttpResponseRedirect(request.path_info)
        else:
            form = KeywordsForm(keywords=keywords)

        context = self.admin_site.each_context(request)
        context['opts'] = self.model._meta
        context['form'] = form
        return render(request, "admin/keywords_edit.html", context)
```

Figure 235 Backend Function to Modify Keywords

Backend Function to Modify Keywords:

- This function is part of a Django admin class that allows for the modification of keywords through the admin interface.
- It includes a custom method `get_urls` to add an additional URL path for editing keywords.
- The function first loads the current keywords from a JSON file (`keywords_config.json`) located within the config directory relative to the `BASE_DIR` of the Django project.
- If the method of the request is POST, it creates an instance of `KeywordsForm` with the posted data and the currently loaded keywords. If the form is valid, it saves the updated keywords back to the JSON file and informs the user of the successful update.
- It then creates a context with the form and renders the `admin/keywords_edit.html` template to display the form for editing.

8.3.1.2.3 View Saved Projects and Export

The screenshot shows a dark-themed administrative interface for managing saved projects. At the top, there's a header with the text "Select project to change" and three buttons: "EXPORT TO CSV", "EXPORT TO PDF", and "ADD PROJECT +". Below the header, there's a search bar labeled "Action: ----- Go" followed by "0 of 39 selected". The main content area is a table listing several projects:

- Name of the Project:** Food and Nutrition Security Enhancement Project - P164319
Link to the project: [View](#)
Procurement Details: Procurement of improved maize seed.
Link to the Procurement: [View](#)
Date of Publication: March 4, 2024
Organization: WB
- Enhancing human security through local climate actions**
[View Project](#)
- Recovery & Resilience**
[View Project](#)
- Name of the Project:** GCP /GLO/505/ROK: Implementation of Codex standards to support containment and reduction of foodborne AMR
Duration of the Project: 20-Jul-2021 - 19-Jun-2026
Link to the project: [View](#)
Organization: UNFAO
- Name of the Project:** Ministry of Health and Population Implementation Letter - Food Security Program Activities
Duration of the Project: 2010-07-17 - 2013-07-16
Sector: Agriculture
Organization: USAID
- Name of the Project:** Green and Resilient Rural Recovery through Agri-Food System Transformation in the Asia and Pacific Region - Senior Natural Capital Finance Specialist
Link to the project: [View](#)
Project Status: Status: Closed
Project Description: 55113-001; Regional, Bangladesh, Cambodia, India, Kyrgyz Republic, Lao People's Democratic Republic, Maldives, Mongolia, Nauru, Nepal, Pakistan, China, People's Republic of, Sri Lanka, Vanuatu, Viet Nam; Agriculture, natural resources and rural development; Posting date: 27 Jul 2022
Organization: ADB
- Enhancing human security through local climate actions**
[View Project](#)

Figure 236 Admin Panel for Saved Projects

```

##saved projects
try:
    admin.site.unregister(Project)
except admin.sites.NotRegistered:
    pass

class ProjectAdmin(admin.ModelAdmin):
    list_display = ('get_project_details', 'user',)
    readonly_fields = ('get_project_details',)

    def get_project_details(self, obj):
        # Using BeautifulSoup to clean the content
        soup = BeautifulSoup(obj.content, features="html.parser")

        # Remove the 'Save Project' button and any other unwanted elements
        for button in soup.find_all('button'):
            button.decompose()

        # Optionally remove other unwanted elements here

        # Construct the clean HTML
        clean_html = ''.join(str(tag) for tag in soup)

        # Return safe HTML for the admin interface
        return mark_safe(clean_html)

    get_project_details.short_description = 'Project Details'

    # This makes the custom 'get_project_details' method be used on the detail view as well
    def get_readonly_fields(self, request, obj=None):
        if obj: # This is the detail view
            return ('get_project_details',)
        else: # This is the list view
            return self.readonly_fields

```

Figure 237 Backend Function to View Saved Projects

Viewing Saved Projects:

- The ProjectAdmin class is a Django model admin that unregisters the existing Project model from the admin site and then re-registers it with customized options.
- It sets up a list_display to show a method get_project_details which cleans the project's HTML content using BeautifulSoup, and marks the cleaned HTML as safe to render in the admin.

```

##export_to_csv_or_pdf

def get_urls(self):
    urls = super().get_urls()
    my_urls = [
        path('export-csv/', self.export_csv, name='export_csv'),
        path('export-pdf/', self.export_pdf, name='export_pdf'),
    ]
    return my_urls + urls

@admin.action(description='Export selected to CSV')
def export_csv(self, request):
    queryset = self.model.objects.all()

    response = HttpResponse(content_type='text/csv')
    response['Content-Disposition'] = 'attachment; filename=projects.csv'
    writer = csv.writer(response)

    # Write the header.
    writer.writerow(['ID', 'Project Details', 'User'])

    for obj in queryset:
        # Use BeautifulSoup to parse the HTML content
        soup = BeautifulSoup(obj.content, features="html.parser")

        # Initialize an empty list to collect text lines
        details_lines = []

        # Iterate over paragraph tags or any other logical division of content
        for content in soup.find_all(['p', 'h1', 'h2', 'h3', 'div', 'li']): #
            # Extract the text from each content section
            content_text = content.get_text(separator=" ", strip=True)
            # Unescape HTML entities like '&' to '&
            content_text = html.unescape(content_text)
            details_lines.append(content_text)

            # If there's an <a> tag, add its text and href as a new line
            for link in content.find_all('a', href=True):
                link_text = f"{link.text} ({link['href']})"
                details_lines.append(link_text)

        # Join all the details with a newline character
        project_details = "\n".join(details_lines)

        # Get the user associated with the project
        user = obj.user.get_full_name() if obj.user else 'Anonymous'

        # Write row to CSV
        writer.writerow([obj.pk, project_details, user])

    return response

```

Figure 238 Backend function to export to CSV.

Export to CSV:

- The `get_urls` method adds custom paths to the admin for exporting projects to CSV or PDF.
- The `export_csv` admin action creates a CSV response that lists projects' details, including the content cleaned by `get_project_details` and the associated user. It uses BeautifulSoup to parse and extract text from the HTML content of each project.

```

@admin.action(description='Export selected to PDF')
def export_pdf(self, request):
    buffer = BytesIO()
    doc = SimpleDocTemplate(buffer, pagesize=letter)
    styles = getSampleStyleSheet()

    # Define a Paragraph style for hyperlinks
    styles.add(ParagraphStyle(name='Link', textColor=colors.blue, underline=True))

    elements = []

    queryset = self.model.objects.all()
    for obj in queryset:
        soup = BeautifulSoup(str(self.get_project_details(obj)), features="html.parser")

        # Extract text and links from the soup object
        content_pieces = []
        for elem in soup.recursiveChildGenerator():
            if elem.name == 'a':
                # Make sure the link is complete (e.g., starts with http)
                link = elem['href'] if elem.has_attr('href') else ''
                # Only add text and a hyperlink if there's an actual link
                if link:
                    content_pieces.append(Paragraph(f'<a href="{link}" color="blue">{elem.text}</a>', styles['Link']))
            elif elem.name is None: # NavigableString objects have None as their name
                text = str(elem).strip()
                if text:
                    content_pieces.append(Paragraph(text, styles['Normal']))

        # Add each piece of content to the elements list
        for content_piece in content_pieces:
            elements.append(content_piece)
            elements.append(Spacer(1, 0.05 * inch))

        # Add extra space after each project
        elements.append(Spacer(1, 0.2 * inch))

    # Build the PDF
    doc.build(elements)
    pdf = buffer.getvalue()
    buffer.close()

    response = HttpResponse(content_type='application/pdf')
    response[Content-Disposition] = 'attachment; filename="projects.pdf"'
    response.write(pdf)

    return response

```

Figure 239 Backend function to export to PDF.

Export to PDF:

- The export_pdf admin action creates a PDF file of the projects' details.
- It uses the ReportLab library to build a PDF document, which is cleaned by the get_project_details method using BeautifulSoup.
- It generates a response with Content-Type set to application/pdf, prompting the browser to download the file as an attachment.

8.3.2 Sample Code for the Automation Script

8.3.2.1 Scrape Bots

8.3.2.1.1 Notable Modules

```
import csv
from csv import writer
import pandas as pd
import numpy as np
import time
from bs4 import BeautifulSoup as soup
import requests
from urllib.request import urlopen as uReq
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.common.exceptions import NoSuchElementException, ElementNotSelectableException
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.common.action_chains import ActionChains
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.common.desired_capabilities import DesiredCapabilities
from fake_useragent import UserAgent as ua
```

Figure 240 Screenshots of Notable Modules/Packages/Libraries used for Scraping.

8.3.2.1.2 Web Driver Setup

```
options = webdriver.ChromeOptions()
options.add_argument('--ignore-certificate-errors')
options.add_argument('--ignore-ssl-errors')
options.add_argument('--headless=new')
options.add_argument('--start-maximized')
options.add_argument('--disable-web-security')
user_agent = ua.random
options.add_argument(f"--user-agent={user_agent}")
options.add_argument('--disable-gpu')
options.add_experimental_option('excludeSwitches', ['enable-logging'])
options.add_experimental_option('detach', True)
driver = webdriver.Chrome(options=options)
driver.set_page_load_timeout(300)

# Open the webpage
driver.get(url)
```

Figure 241 Setting up WebDriver for Scraping

8.3.2.1.3 Scraping Logics

```

while True:
    try:
        ADBS = []
        page = driver.page_source
        page_soup = soup(page, 'html.parser')

        containers = page_soup.findAll('div', {'class': 'item linked'})

        for container in containers:
            ADB = {}

            title = container.find('div',{'class' : 'item-title'})
            project_titles = title.find('a')
            project_link = title.a['href']
            for project_title in project_titles:
                project_title = (project_titles.text.strip())

            ADB['Name of the Project'] = project_title
            ADB['Link to the project'] = project_link

            project_status = container.find('div', {'class': 'item-meta'})
            stats = project_status.find('div')
            stat = (stats.text.strip())

            ADB['Project Status'] = stat

            project_summary = container.find('div', {'class':'item-summary'})
            summary = (project_summary.text.strip())

            ADB['Project Description'] = summary

            file = open('ADB.csv', 'a', newline= '', encoding='utf-8')
            writer = csv.writer(file)
            headers = ([project_title, project_link ,stat, summary])
            writer.writerow(headers)
            file.close()

            pass

            time.sleep(5)
            page_num += 1
            new_url = f'https://www.adb.org/projects/tenders/country/nepal?page={page_num}'
            driver.get(new_url)
            driver.set_page_load_timeout(60)

            driver.quit()
    except:
        break

```

Figure 242 Sample Scraping Logic for ADB

ADB (Asian Development Bank) Scraping Logic does following:

- Use Selenium to navigate the pages.
- Parse page source with BeautifulSoup.
- Find all project containers.
- For each container, extract the project title, link, status, and summary.
- Write the extracted information into 'ADB.csv'.
- Handle pagination by incrementing the page number and loading new pages until all pages are scraped.

```

while True:
    try:
        JICAS=[]
        page = driver.page_source
        page_soup = soup(page, 'html.parser')
        containers = page_soup.findAll('div', {'class': 'disclist'})

        for container in containers:
            JICA = {}
            project_name = container.find('li')
            name = (project_name.text.strip())

            print(name)
            JICA['Name of the Project']= name

            JICA = {}
            project_link = container.find('a')

            project_links = re.findall(r'href="(.*?)"',str(project_link))
            project_links = [link for link in project_links if not link.startswith('/english/assets')]
            project_links = [f'' if link.startswith\('/nepal/english'\) else link for link in project\_links\]
            project\_links
            print\(project\_links\)
            JICA\['Link to the Project'\] = project\_links

            file = open\('JICA.csv', 'a', newline= '', encoding='utf-8'\)
            writer = csv.writer\(file\)
            headers = \(\[name, project\_links\]\)
            writer.writerow\(headers\)
            file.close\(\)

            driver.close\(\)
    except:
        break

```

Figure 243 Scraping Logic for JICA

JICA (Japan International Cooperation Agency) Scraping Logic does following:

- Retrieve page content with Selenium.
- Parse with BeautifulSoup to find project containers.
- Extract the project name and links using regex and BeautifulSoup.
- Adjust project links to ensure they are complete.
- Write project names and links to 'JICA.csv'.
- Close the driver at the end of the script.

```

page_num = 1
max_page_num = 8

driver.set_page_load_timeout(60)
url = 'https://www.eda.admin.ch/countries/nepal/en/home/international-cooperation/projects.olddesign.olddesign.olddesign.olddesign.par_projectfilter_2b80_page{page_num}.html'
driver.get(url)

# Open the webpage
while True:
    try:
        SDCS = []
        page = driver.page_source
        page_soup = soup(page, 'html.parser')

        containers = page_soup.find_all('article', class_= 'clearfix')

        containers = [article for article in page_soup.find_all('article', class_= 'clearfix') if article.find('h4')]

        for container in containers:
            SDC = {}
            h4_tag = container.find('h4')
            a_tag = h4_tag.find('a') if h4_tag else None
            project_name = a_tag.text.strip() if a_tag else 'N/A'
            SDC['Name of the Project'] = project_name

            project_link = 'https://www.eda.admin.ch' + a_tag['href'] if a_tag else 'N/A'
            SDC['Link to the project'] = project_link

            small_tag = container.find('small')
            project_duration = small_tag.text.strip() if small_tag else 'N/A'
            SDC['Duration of the Project'] = project_duration

            strong_tags = container.find_all('strong')
            project_description = ' '.join([strong.text.strip() for strong in strong_tags]) if strong_tags else 'N/A'
            SDC['Project Details'] = project_description

            file = open('SDC.csv', 'a', newline= '', encoding='utf-8')
            writer = csv.writer(file)
            headers = ([project_name, project_link, project_duration, project_description])
            writer.writerow(headers)
            file.close()

            pass
            time.sleep(5)
            page_num += 1
            if page_num<=max_page_num:
                new_url = 'https://www.eda.admin.ch/countries/nepal/en/home/international-cooperation/projects.olddesign.olddesign.olddesign.olddesign.par_projectfilter_2b80_page{page_num}.html'
                driver.get(new_url)
            else:
                driver.quit()
    except:
        break

```

Figure 244 Scraping Logic for SDC

SDC (Swiss Agency for Development and Cooperation) Scraping Logic does following:

- Use Selenium to interact with web pages.
- Parse pages with BeautifulSoup for project details.
- Extract project name, link, duration, and description.
- Write details to 'SDC.csv'.
- Manage pagination by checking the maximum page number.

```

try:
    # Scroll to the button and click it
    driver.execute_script("arguments[0].scrollIntoView(true);", button)
    driver.execute_script("arguments[0].click();", button)
    time.sleep(1)
except Exception as e:
    print(f"Error clicking a 'Read More' button: {e}")

UKAIDS = []
page = driver.page_source
page_soup = soup(page, 'html.parser')

project_containers = page_soup.find('div', id = 'response-container')
containers = project_containers.find_all('div', class_ = 'app-search-result')

for container in containers:
    UKAID = {}

    project_title_container = container.find('h3', class_ = 'govuk-heading-s app-search-result-title')
    name = project_title_container.text.strip()
    UKAID['Name of the Project'] = name

    link = project_title_container.find('a', class_ = 'govuk-link--no-visited-state')['href']
    full_link = "https://devtracker.fcdo.gov.uk" + link # Ensure the link is complete
    UKAID['Link to the project'] = full_link

    # Finding the start date within the information container
    info_divs = container.find_all('div', class_ = 'app-search-result-info')
    start_date = ""

    for info_div in info_divs:
        # Each 'info_div' contains multiple 'div' elements for different info types
        info_pairs = info_div.find_all('div')
        for pair in info_pairs:
            title = pair.find('span', class_ = 'govuk-body-s app-search-result-info_title').text.strip()
            value = pair.find('span', class_ = 'govuk-body-s app-search-result-info_value').text.strip()
            if title == "Start date":
                start_date = value
                UKAID['Start Date'] = start_date
                break
        if start_date: # Exit the outer loop if start date is found
            break

    # Finding the project description
    description = container.find('p', class_ = 'govuk-body-s description').text.strip()
    UKAID['Project Details'] = description

    # Output for debugging

    file = open('UKAID.csv', 'a', newline= '', encoding='utf-8')
    writer = csv.writer(file)
    headers = ([name,"https://devtracker.fcdo.gov.uk" + link, description, start_date ])
    writer.writerow(headers)
    pass

    next_btn_xpath = f"/html/body/div[5]/main/div[4]/div[2]/div[3]/ul/li[7]/a"
    next_btn = driver.find_element(By.XPATH, next_btn_xpath)

    if next_btn:
        ActionChains(driver) \
            .scroll_to_element(next_btn) \
            .click(next_btn).perform()
        time.sleep(4)
    else:
        driver.quit()

except:
    break

```

Figure 245 Scraping Logic for UKAID

UKAID (United Kingdom's Foreign, Commonwealth & Development Office) Scraping Logic does following:

- Navigate and interact with pages using Selenium.
- Extract project name, link, start date, and description using BeautifulSoup.
- Write the data into 'UKAID.csv'.
- Use Selenium's ActionChains to interact with the 'Next' button for pagination.

```

while True:
    try:
        body = driver.find_element(By.TAG_NAME, 'body')
        body.send_keys(Keys.END)
        body.send_keys(Keys.PAGE_UP)
        time.sleep(5)

        buttons = driver.find_element(By.CLASS_NAME, 'cta-button')
        button = buttons.find_element(By.TAG_NAME, 'button')

        ActionChains(driver)\n            .scroll_to_element(button)\n            .click(button).perform()
        time.sleep(2)

        UNDPs=[]
        page = driver.page_source
        page_soup = soup(page, 'html.parser')

        containers = page_soup.findAll('div', {'class': 'content-card'})

        for container in containers:
            UNDP = {}
            link = container.a['href']
            project_link = 'https://www.undp.org/' + str(link)
            UNDP['project_link'] = project_link

            content_captions = container.find('div', class_='content-caption')

            h5 = content_captions.find('h5')
            project_name = str(h5).replace('<h5>', '').replace('</h5>', '')
            UNDP['project_name'] = project_name

            print(project_name , project_link)
            file = open('UNDP.csv', 'a', newline= '', encoding='utf-8')
            writer = csv.writer(file)
            headers = ([project_name, project_link])
            writer.writerow(headers)
            file.close()

            pass

    except:
        break

```

Figure 246 Scraping Logic for UNDP

- UNDP (United Nations Development Programme) Scraping Logic does following:
- Scroll through the page using Selenium's key presses.
- Extract the project names and links after parsing with BeautifulSoup.
- Save the extracted data into 'UNDP.csv'.

```

while True:
    try:
        UNFAOS=[]
        page = driver.page_source
        page_soup = soup(page, 'html.parser')
        containers = page_soup.findAll('div', {'class': 'rgaccord1-nest'})
        for container in containers:
            UNFAO = {}
            h3 = container.find('h3', {'class': 'rgaccord1-toggle'})
            project_name = (h3.text.strip())
            UNFAO['Name of the Project'] = project_name

            content = container.find('div',{'class':'csc-default'})
            project_desc = (content.text.strip())
            match = re.search(r'Period:(.*)', project_desc)
            period = match.group(0).replace('Period:', '')
            UNFAO['Duration of the Project'] = period

            project_link = 'https://www.fao.org/nepal/programmes-and-projects/project-list/ru/?no_cache=1'
            UNFAO['Link to the project'] = project_link

            print(project_name , period, project_link)

            file = open('UNFAO.csv', 'a', newline= '', encoding='utf-8')
            writer = csv.writer(file)
            headers = ([project_name, period, project_link])
            writer.writerow(headers)
            file.close()

            driver.close()
            pass

    except:
        break

```

Figure 247 Scraping Logic for UNFAO

UNFAO (Food and Agriculture Organization of the United Nations) Scraping Logic does following:

- Retrieve and parse the page content using Selenium and BeautifulSoup.
- Extract the project name, duration, and link.
- Write the extracted data into 'UNFAO.csv'.
- Close the Selenium driver at the end.

```

file = open('USAID.csv', 'w', newline='')

url = 'https://www.foreignassistance.gov/data_query/results.csv?&transaction_type=Obligations&country_id=524'

response = urllib.request.urlopen(url)
csv_content = response.read().decode('utf-8')

csv_file = StringIO(csv_content)

df = pd.read_csv(csv_file)

selected_columns = ['Activity Name', 'International Sector Name', 'Activity Start Date', 'Activity End Date']
subset_df = df[selected_columns]

subset_df = subset_df.drop_duplicates()

subset_df['Name of the Project'] = subset_df['Activity Name']
subset_df['Duration of the Project'] = subset_df['Activity Start Date'] + " - " + subset_df['Activity End Date']
subset_df['Sector'] = subset_df['International Sector Name']

subset_df = subset_df.drop(['Activity Name', 'International Sector Name', 'Activity Start Date', 'Activity End Date'], axis=1)

subset_df.to_csv('USAID.csv', index=False, quoting=csv.QUOTE_NONNUMERIC)

df1 = pd.read_csv('UNFAO.csv')
df1.drop_duplicates()
df1.dropna()
df1.to_csv('UNFAO.csv', index=False)

```

Figure 248 Scraping Logic for USAID

USAID (United States Agency for International Development) Scraping Logic does following:

- Downloads and reads a CSV file from a given URL using urllib and pandas.
- Selects specific columns, renames them, and removes duplicates.
- Writes cleaned data to 'USAID.csv'.

```

while True:
    try:
        WBs = []
        page = driver.page_source
        page_soup = soup(page, 'html.parser')

        containers = page_soup.findAll('tr', {'class': 'ng-star-inserted'})

        for container in containers:
            WB = {}

            procurement = container.find('td')
            procurement_link = procurement.a['href']
            Procurement_descs = procurement.find('a')
            for Procurement_desc in Procurement_descs:
                procurement_desc = (Procurement_descs.text.strip())

            WB['Procurement Details'] = procurement_desc
            WB['Link to the Procurement'] = procurement_link

            project = container.find('td', {'data-th': 'Project Title'})
            project_link = project.a['href']
            Project_descs = project.find('a')
            for Project_desc in Project_descs:
                project_desc = (Project_descs.text.strip())

            WB['Name of the Project'] = project_desc
            WB['Link to the Project'] = project_link

            Dates = container.find('td', {'data-th': 'Published Date'})
            for Date in Dates:
                Published_Date = (Dates.text.strip())

            WB['Date of Publication'] = Published_Date

            file = open('WB.csv', 'a', newline='', encoding='utf-8')
            writer = csv.writer(file)
            headers = ([project_desc, project_link, procurement_desc, procurement_link, Published_Date])
            writer.writerow(headers)
            file.close()

            next_button_xpath = f"/html/body/div[3]/div[2]/div[2]/div[2]/procurement-search/div/div/div/div/div[1]/div/section/tabsset/section/div/projects-tab[1]/table-api/div[1]/div/div[2]/div/u/l/li[13]/a/i"
            next_button = driver.find_element(By.XPATH, next_button_xpath)

            driver.set_page_load_timeout(300)
            ActionChains(driver)\n                .scroll_to_element(next_button)\n                .click(next_button).perform()\n            time.sleep(1)

            pass

    except:
        break

```

Figure 249 Scraping Logic for WB

WB (World Bank) Scraping Logic does following:

- Uses Selenium to navigate and BeautifulSoup to parse HTML.
- Extracts procurement details, project titles, links, and published dates.
- Writes data to CSV in append mode.
- Navigates pages using an XPATH to find the "next" button.

[Back to Report](#)

8.3.2.2 Containerization

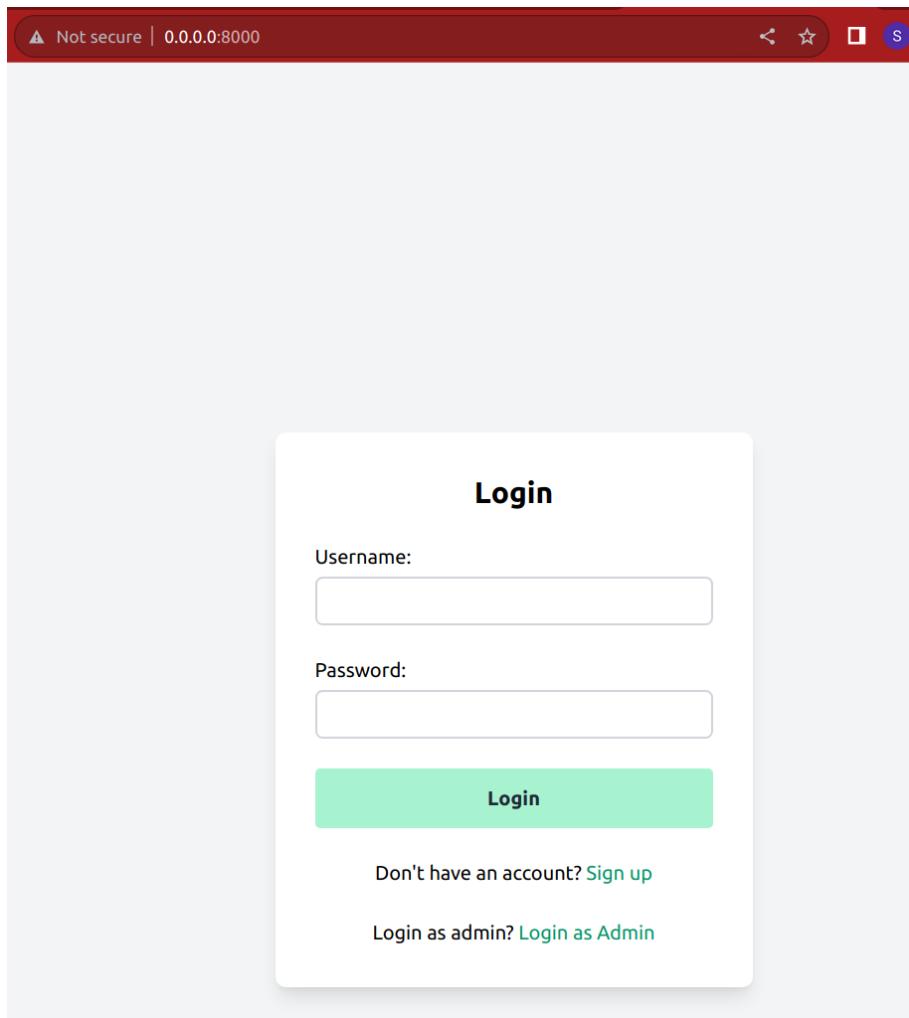
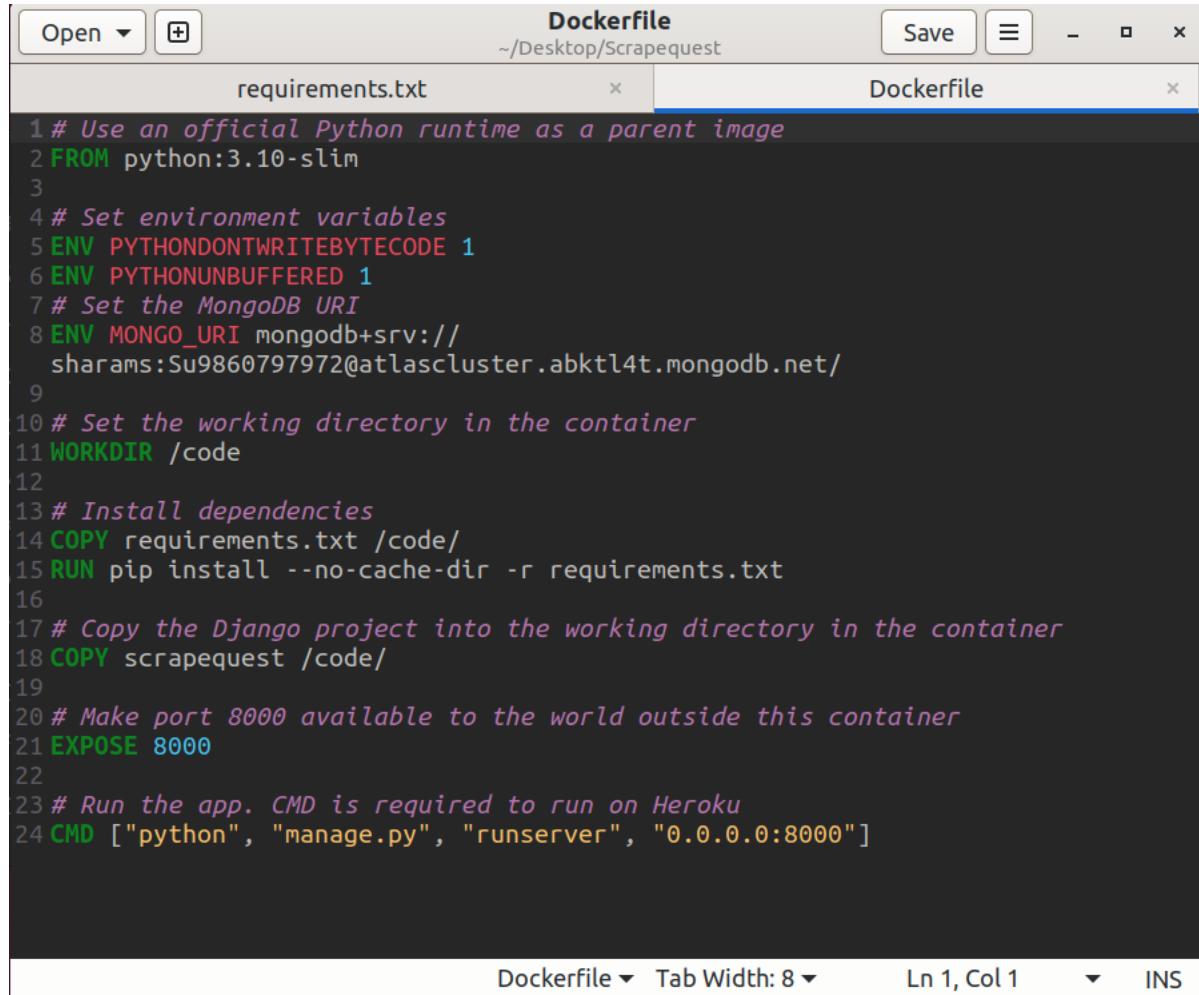


Figure 250 Container in Action

```
sharumss@sharumss-virtual-machine:~/Desktop/Scrapequest$ sudo docker images scrapequest:latest
REPOSITORY      TAG          IMAGE ID      CREATED        SIZE
scrapequest    latest        06023190be01   About an hour ago   363MB
```

Figure 251 Docker Image



```
Dockerfile
~/Desktop/Scrapequest
requirements.txt x Dockerfile x

1 # Use an official Python runtime as a parent image
2 FROM python:3.10-slim
3
4 # Set environment variables
5 ENV PYTHONDONTWRITEBYTECODE 1
6 ENV PYTHONUNBUFFERED 1
7 # Set the MongoDB URI
8 ENV MONGO_URI mongodb+srv://sharams:Su9860797972@atlascluster.abktl4t.mongodb.net/
9
10 # Set the working directory in the container
11 WORKDIR /code
12
13 # Install dependencies
14 COPY requirements.txt /code/
15 RUN pip install --no-cache-dir -r requirements.txt
16
17 # Copy the Django project into the working directory in the container
18 COPY scrapequest /code/
19
20 # Make port 8000 available to the world outside this container
21 EXPOSE 8000
22
23 # Run the app. CMD is required to run on Heroku
24 CMD ["python", "manage.py", "runserver", "0.0.0.0:8000"]
```

Dockerfile ▾ Tab Width: 8 ▾

Ln 1, Col 1 ▾

INS

Figure 252 Dockerfile.

8.4 Appendix D: Designs

8.4.1 Gantt Chart

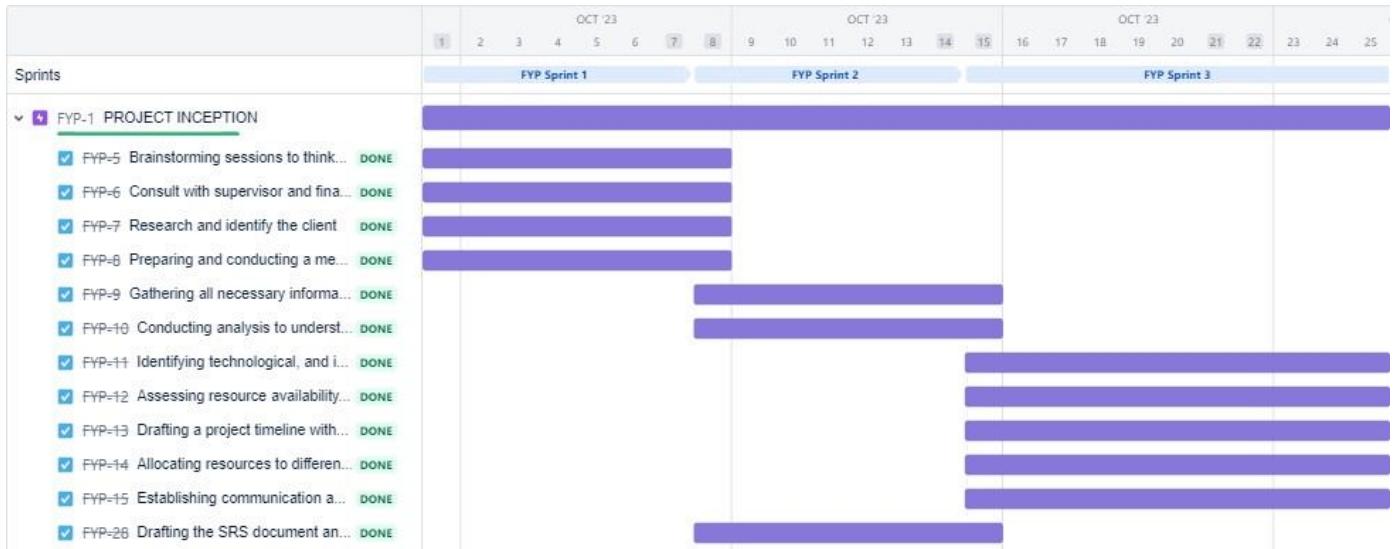


Figure 253 Gantt Chart Timeline of Project Inception

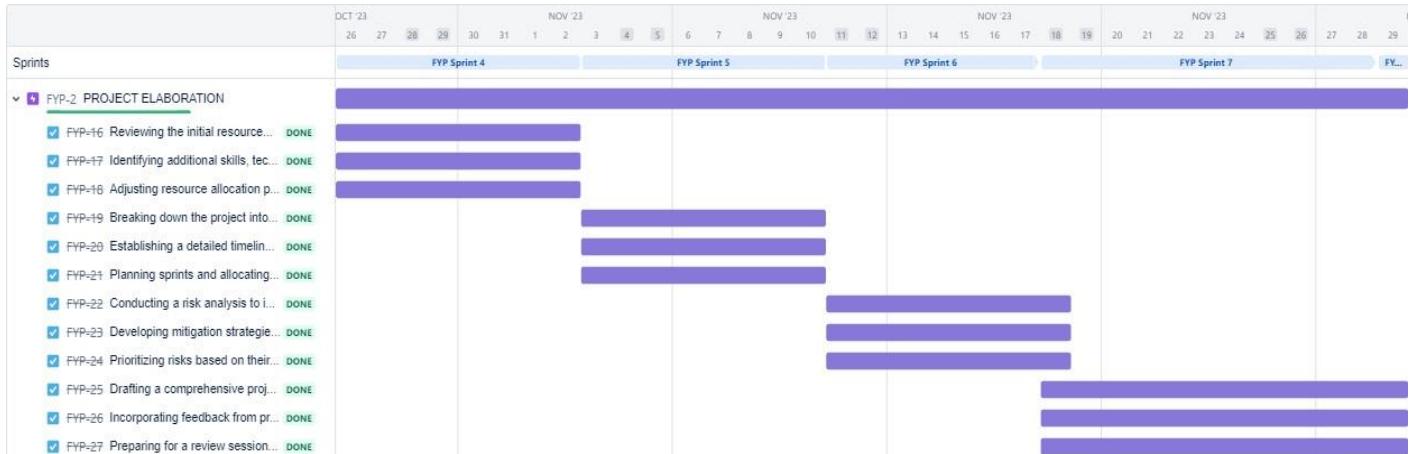


Figure 254 Gantt Chart Timeline of Project Elaboration

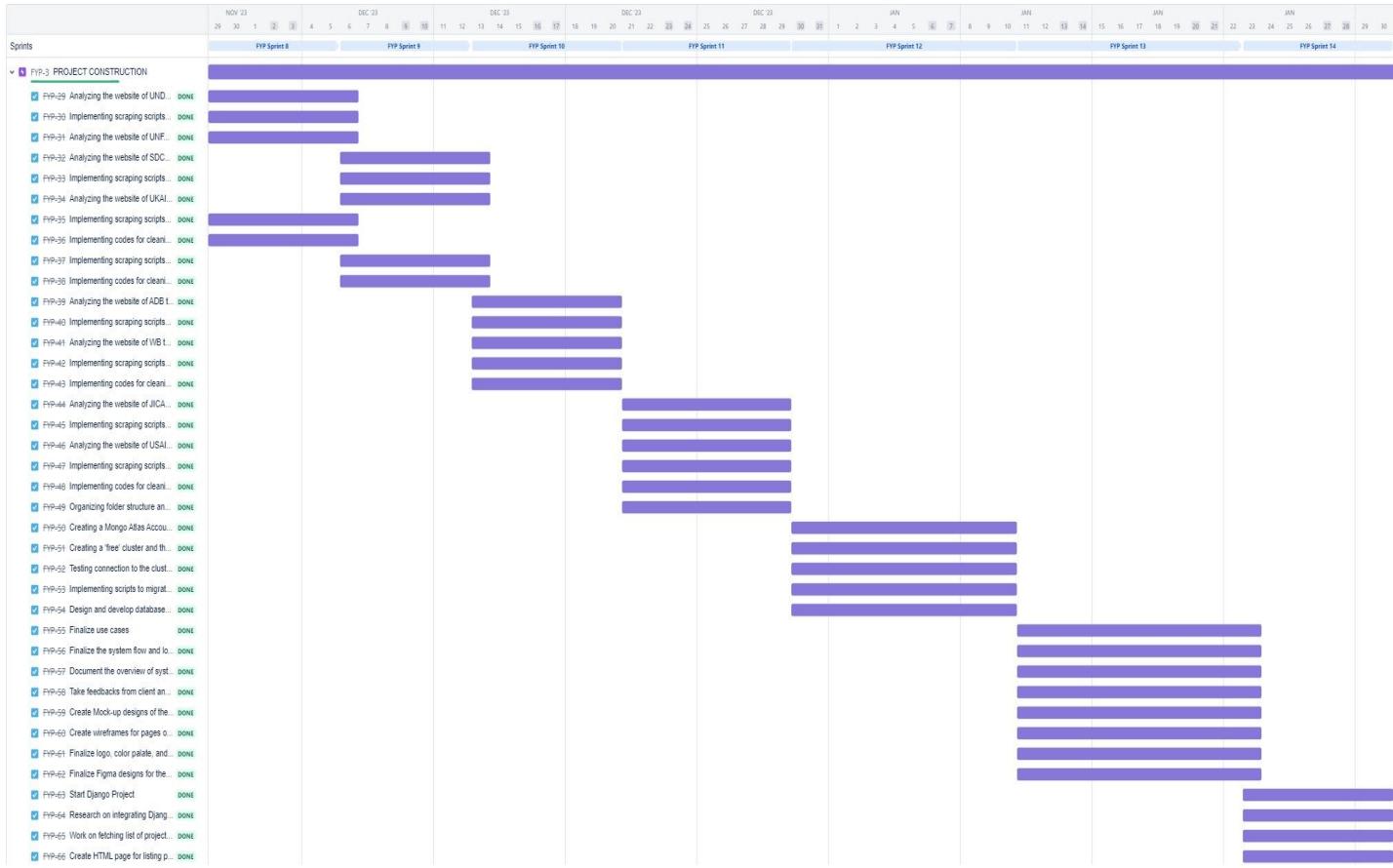


Figure 255 Gantt Chart Timeline of Project Construction (I)

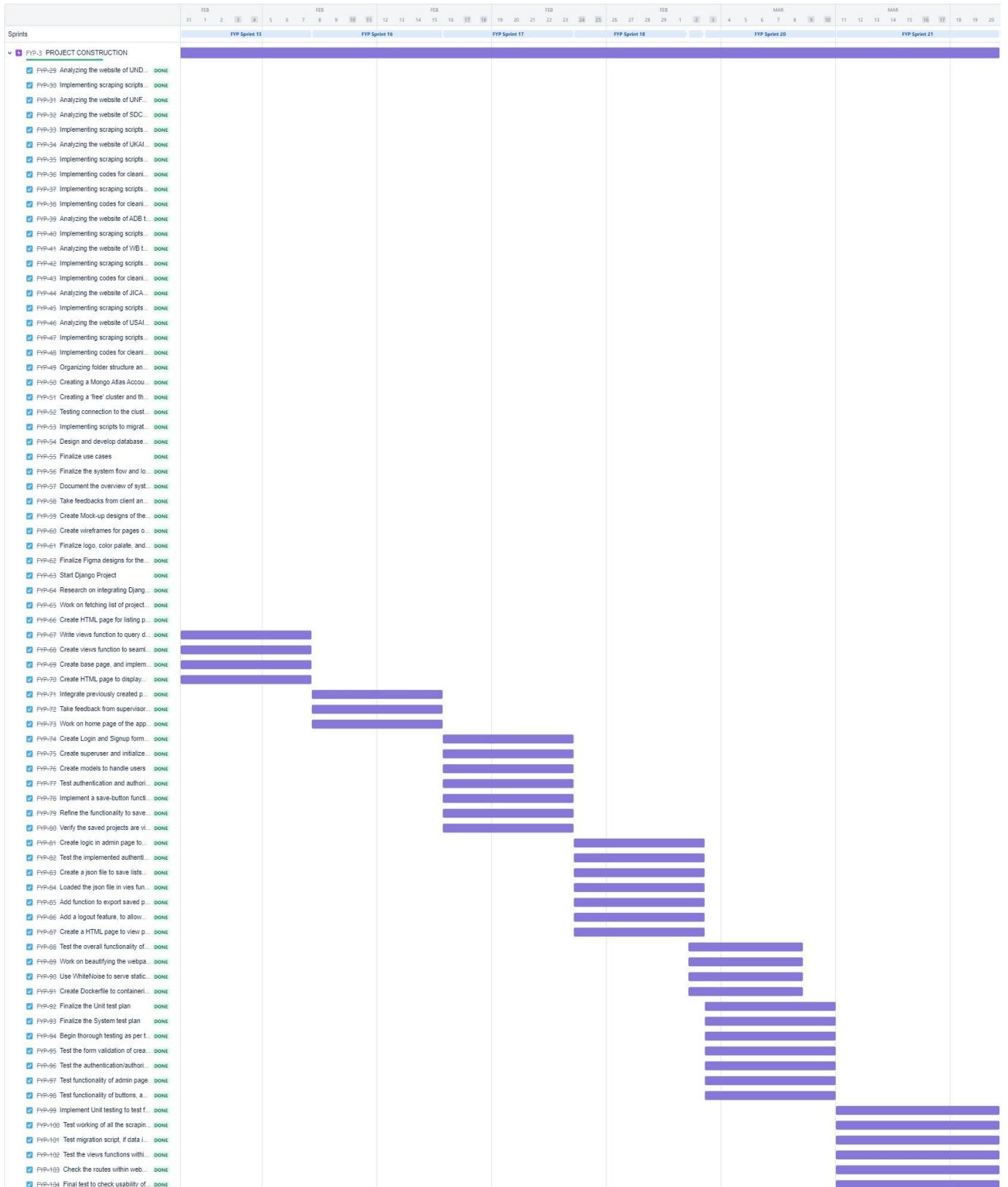


Figure 256 Gantt Chart Timeline of Project Construction (2)

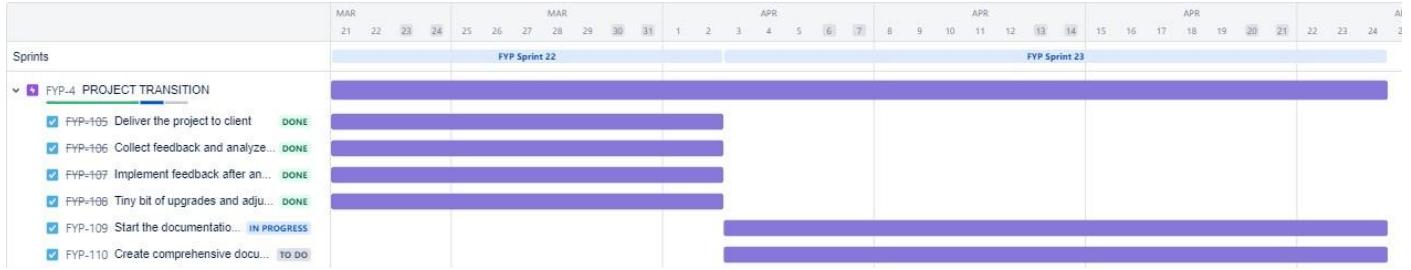


Figure 257 Gantt Chart Timeline of Project Transition

8.4.1.1 Gantt Chart Iterations

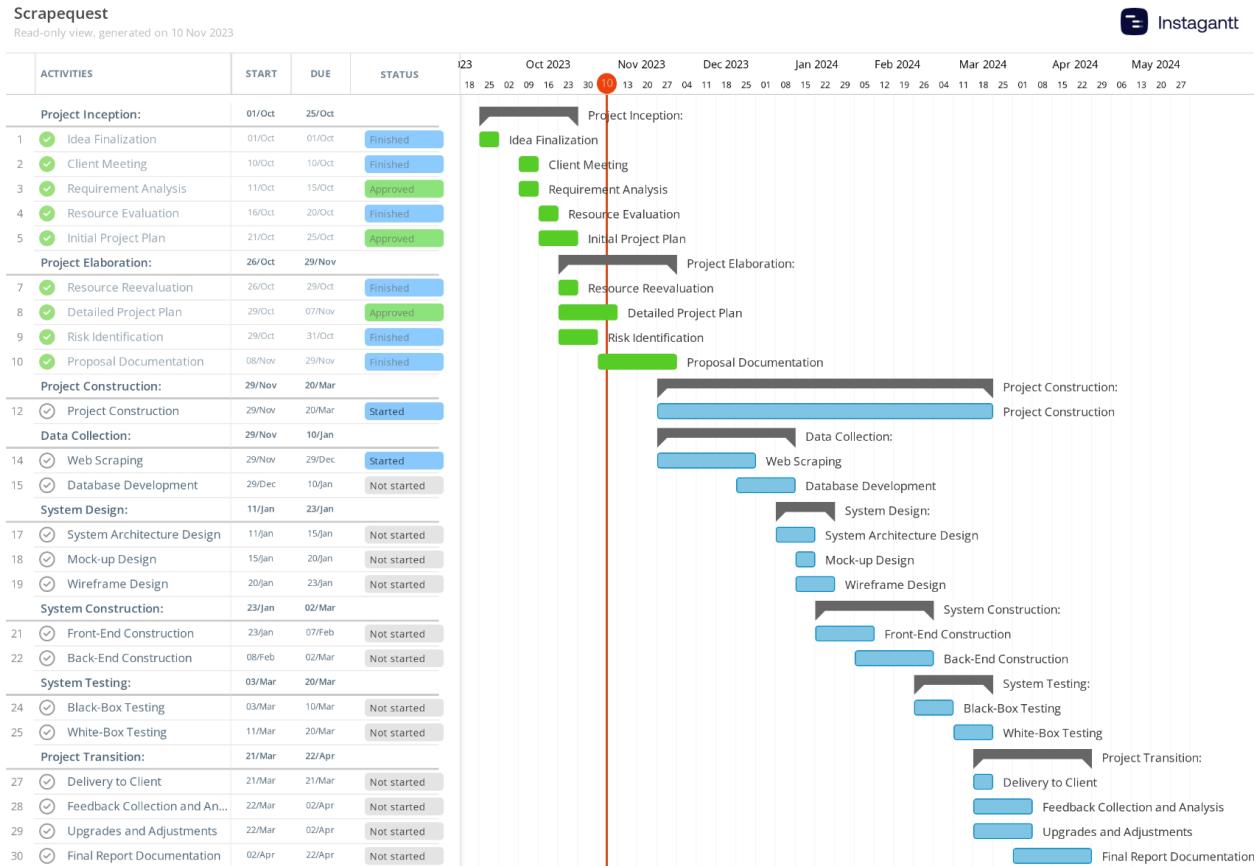


Figure 258 Gantt Chart Iteration 1

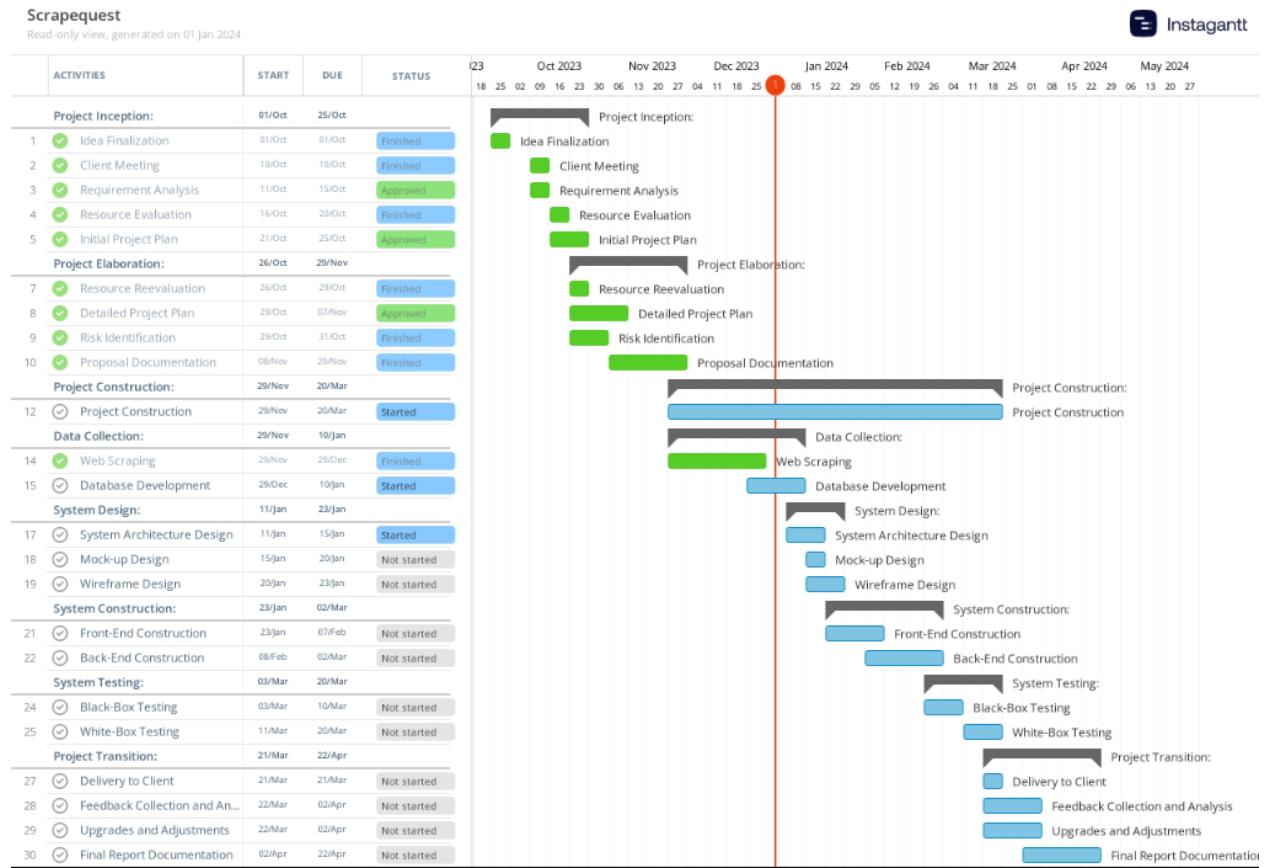


Figure 259 Gantt Chart Iteration 2

8.4.2 Work Breakdown Structure

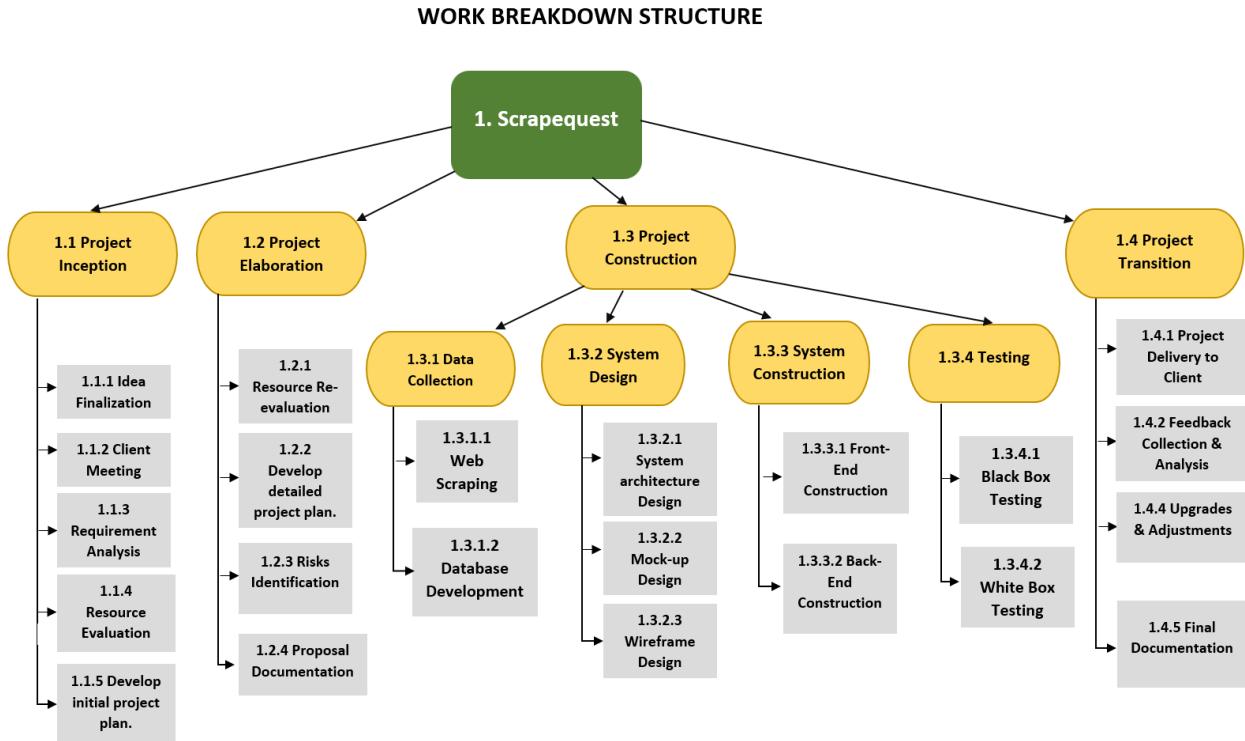


Figure 260 Work Breakdown Structure

8.4.3 Data Flow Diagram

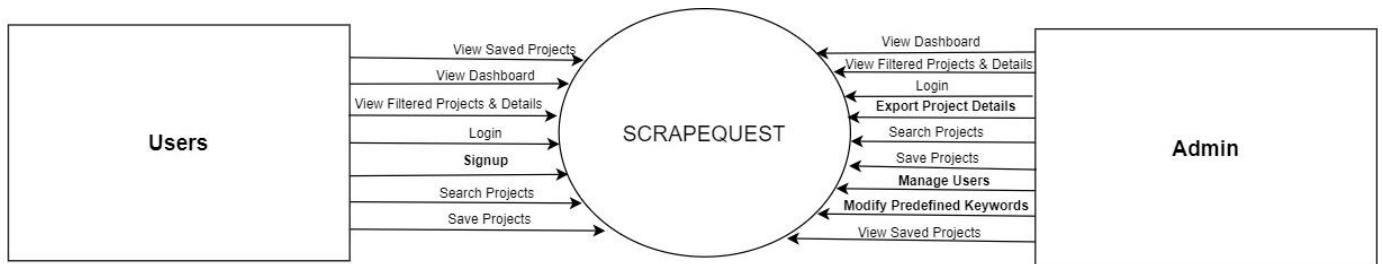


Figure 261 Level 0 DFD for Scrapequest

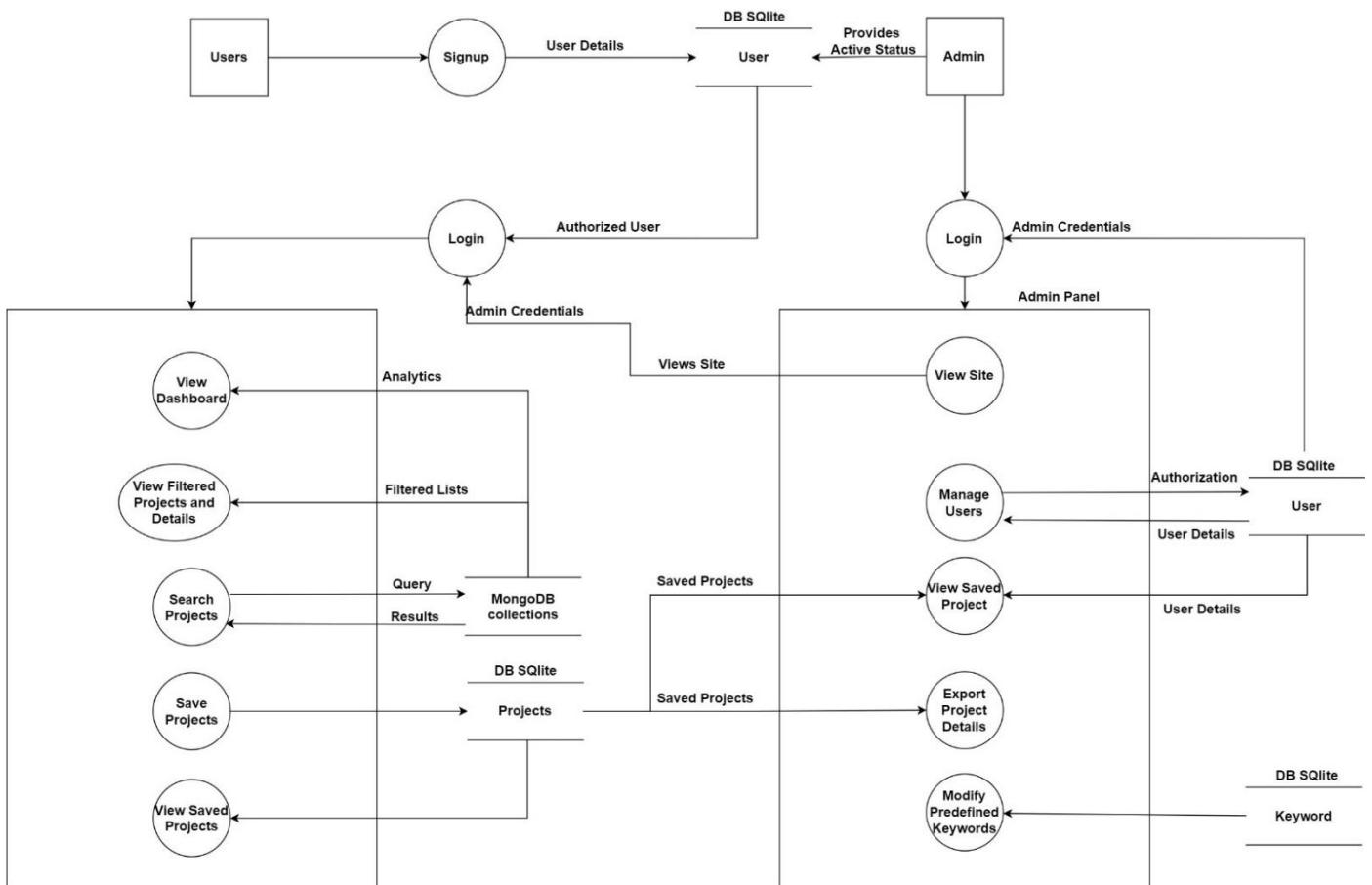


Figure 262 Level 1 DFD for Scrapequest

[Back to Report](#)

8.4.3.1 Data Flow Diagram Iterations



Figure 263 Level 0 DFD for Scrapequest Iteration 1

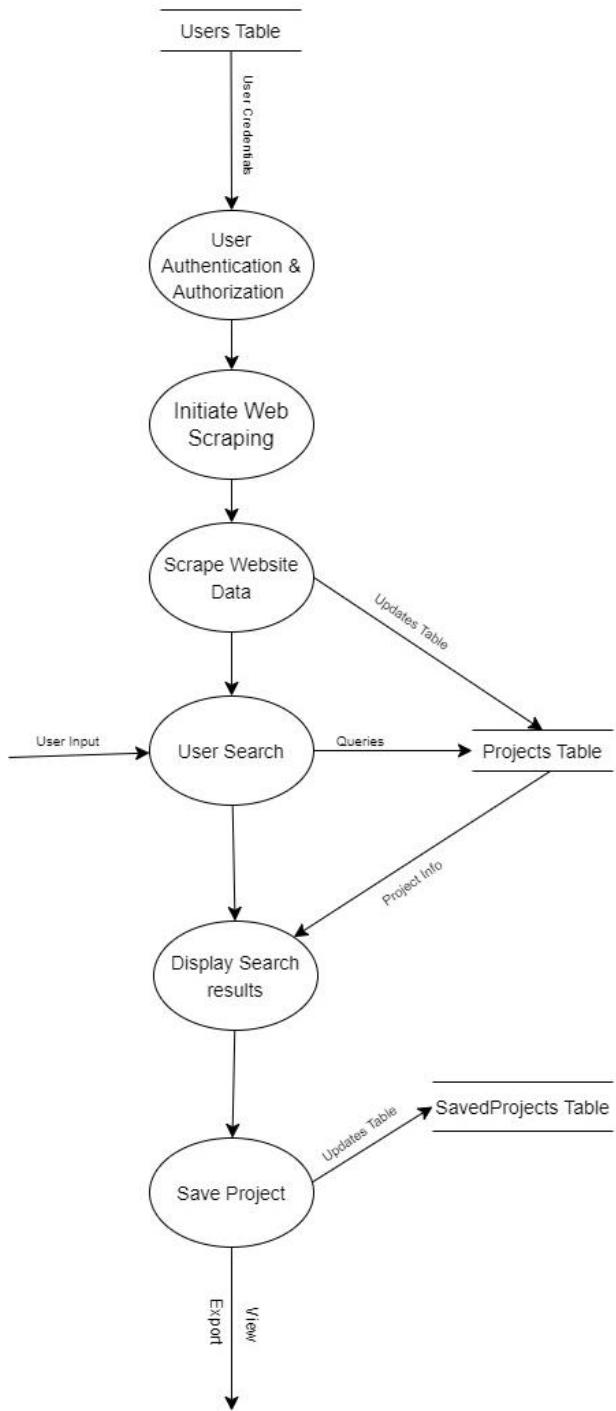


Figure 264 Level 1 DFD (Users) Iteration 1

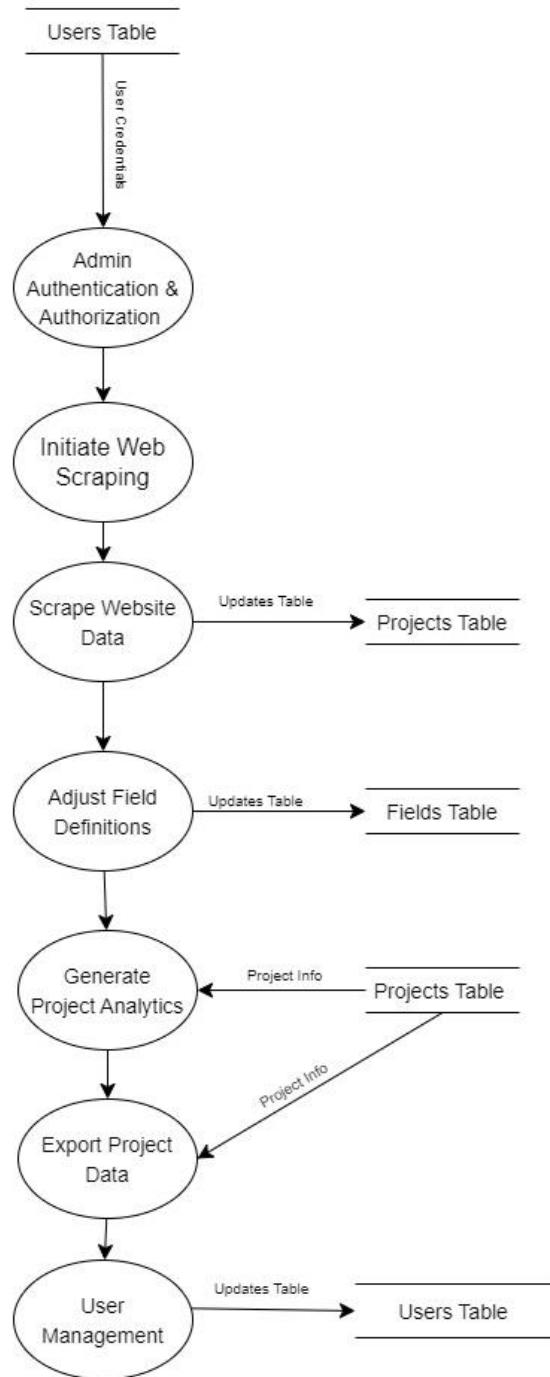


Figure 265 Level 1 DFD (Admin) Iteration 1

8.4.4 Use Case Diagram

Actors of Scrapequest

Actors: User (Employee), Admin (Administrative body)

Table 51 Identification of Actors

Features

Name: Register

Actors: User

Description: Employees will enroll in this application by filling in an enrollment form and need active or staff status for admin, so that, they can login to use the features of Scrapequest .

Table 52 High Level Use Case of Registration

Name: Login

Actors: User, Admin

Description: Users should login into the system using their credentials to use the features of Scrapequest .
--

Table 53 High Level Use Case of Login

Name: Logout

Actors: User

Description: Users can logout of the system.

Table 54 High Level Use Case of Logout

Name: Search Projects

Actors: User, Admin

Description: Users can search for projects after logging into the system and view project information.

Table 55 High Level Use Case of Searching Projects

Name: Save Projects

Actors: User, Admin

Description: Users can save projects of their choice for future reference or to shortlist them.
--

Table 56 High Level Use Case of Saving Projects

Name: View Saved Projects
Actors: User, Admin
Description: Users can revisit or retrieve their saved projects.

Table 57 High Level Use Case of Viewing Saved Projects

Name: View Project Details
Actors: User, Admin
Description: Users can view details of project, filtered via organizations or predefined keywords.

Table 58 High Level Use Case Diagram of Viewing Project Details

Name: View Analytics
Actors: Admin, User
Description: Admin/User can view data and analytics to gain further insights and help them make strategic decision.

Table 59 High Level Use Case of Viewing Analytics

Name: Modify Predefined Fields/Keywords
Actors: Admin
Description: Admin can save modify fields/keywords, according to their liking in case of changes in organizational vision/goals.

Table 60 High Level Use Case of Modifying Predefined Keywords

Name: Manage User
Actors: Admin
Description: Admin can manage, create, edit, or remove users who are allowed to login into the system.

Table 61 High Level Use Case of Managing Users

Name: View and Export Saved Project Details
Actors: Admin
Description: Admin can view, and export saved project details according to their liking in various formats.

Table 62 High Level Use Case of Viewing and Exporting Saved Project Details[Back to Report](#)

8.4.4.1 Use Case Diagram Iterations

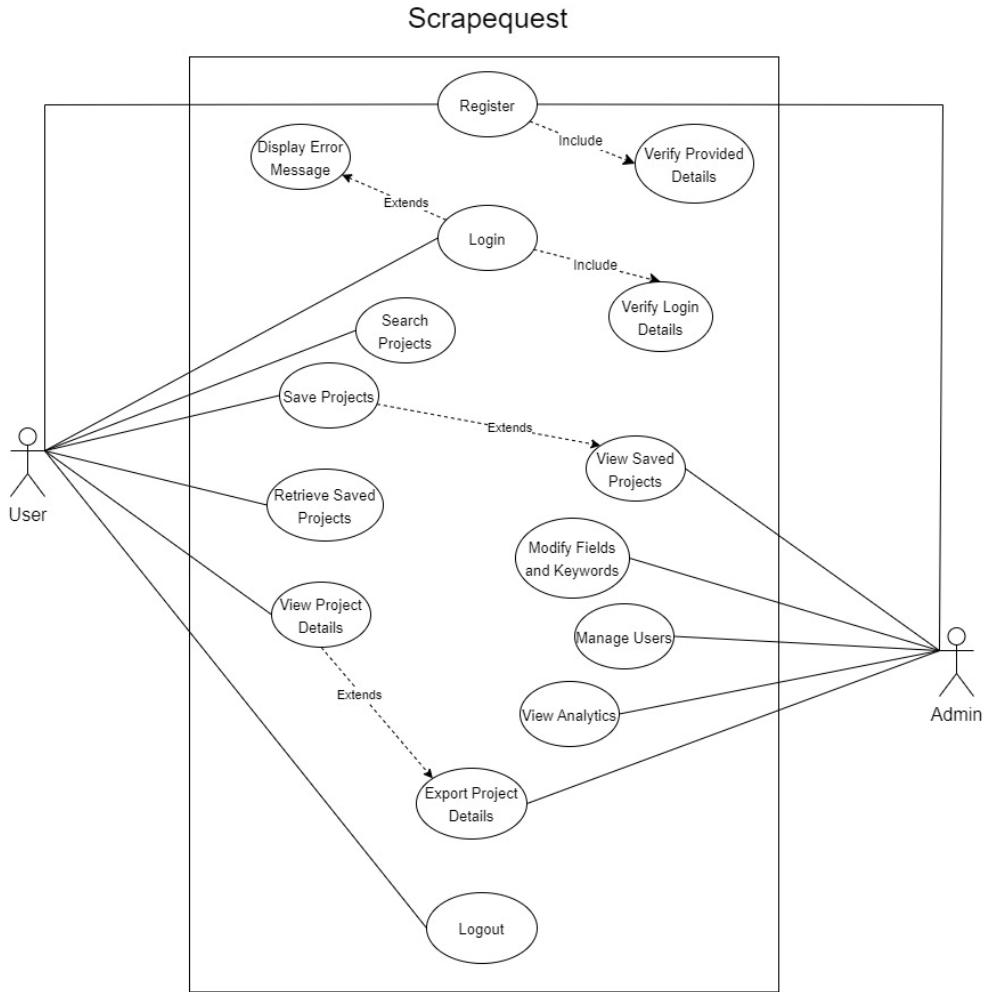
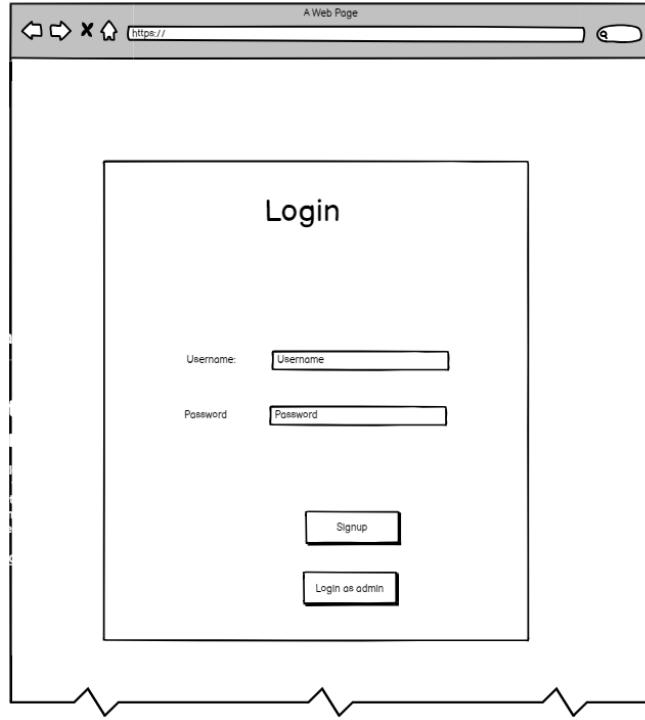


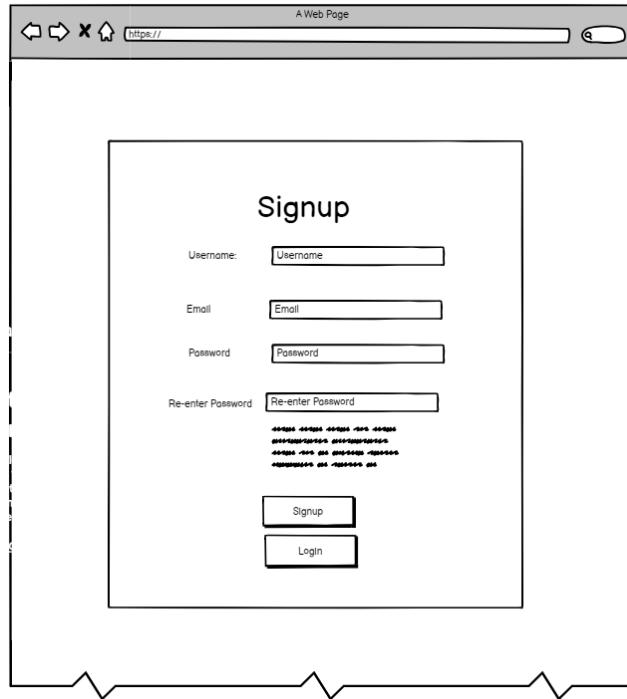
Figure 266 Overall Use Case Diagram Iteration 1

8.4.5 Wireframes



A wireframe diagram of a web browser window titled "A Web Page". The address bar shows "https://". The main content area is a login form titled "Login". It contains two text input fields: "Username:" and "Password". Below these are two buttons: "Signup" and "Login as admin".

Figure 267 Wireframe for Login Page



A wireframe diagram of a web browser window titled "A Web Page". The address bar shows "https://". The main content area is a signup form titled "Signup". It contains four text input fields: "Username", "Email", "Password", and "Re-enter Password". Below the "Re-enter Password" field is a password strength meter consisting of a series of small squares. At the bottom are two buttons: "Signup" and "Login".

Figure 268 Wireframe for Signup Page

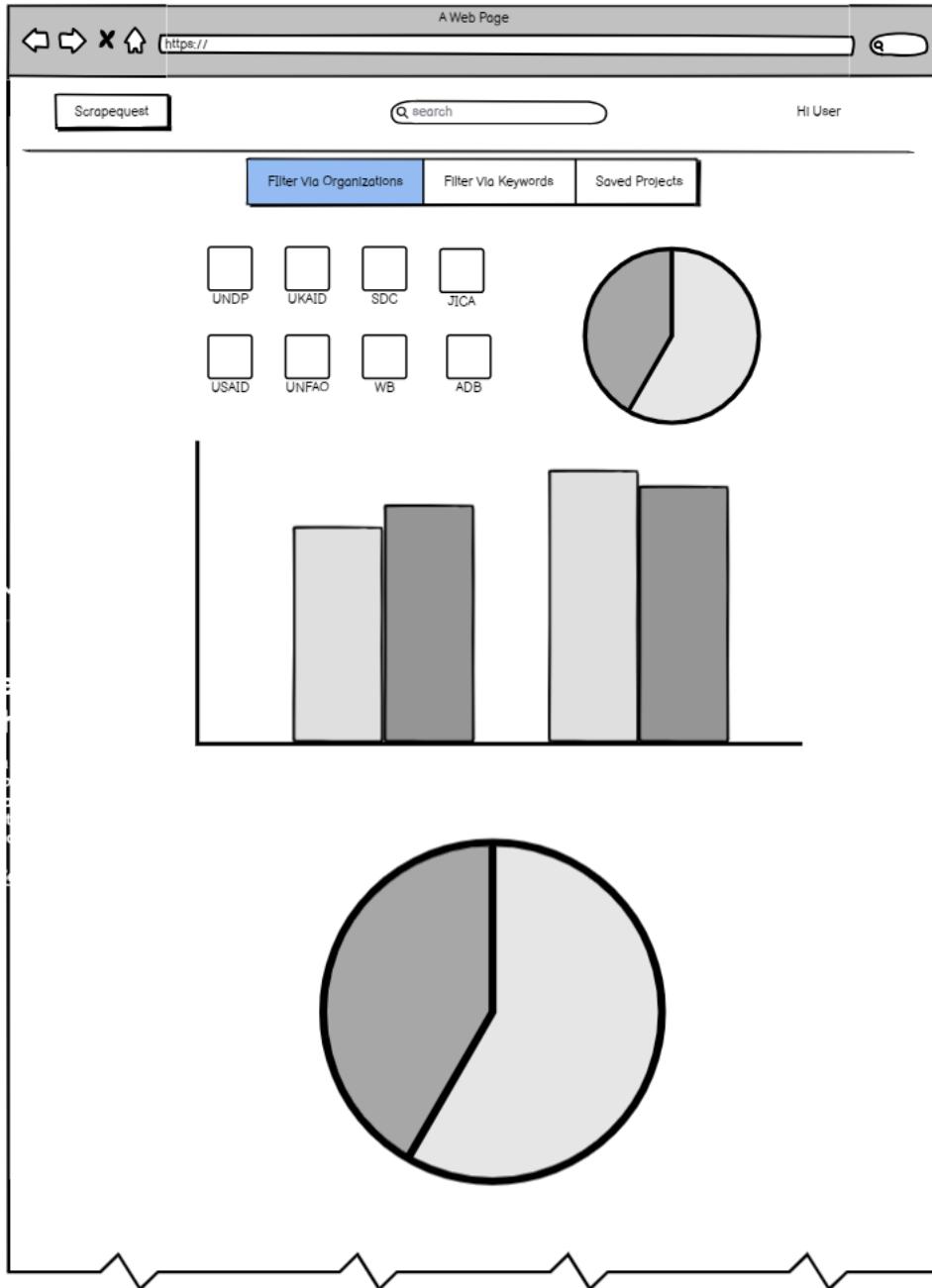


Figure 269 Wireframe for Homepage

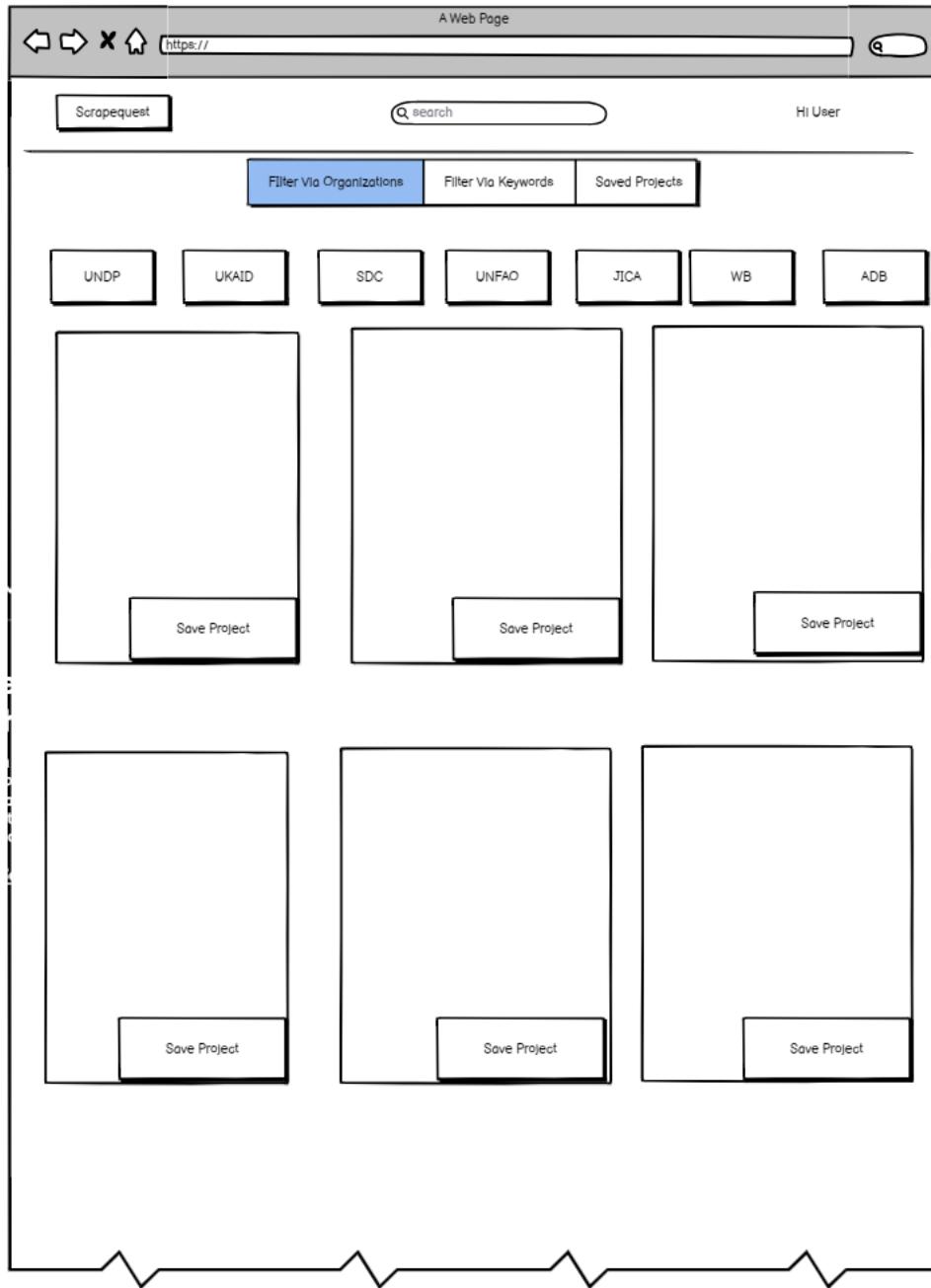


Figure 270 Wireframe for Filter via Organizations Page

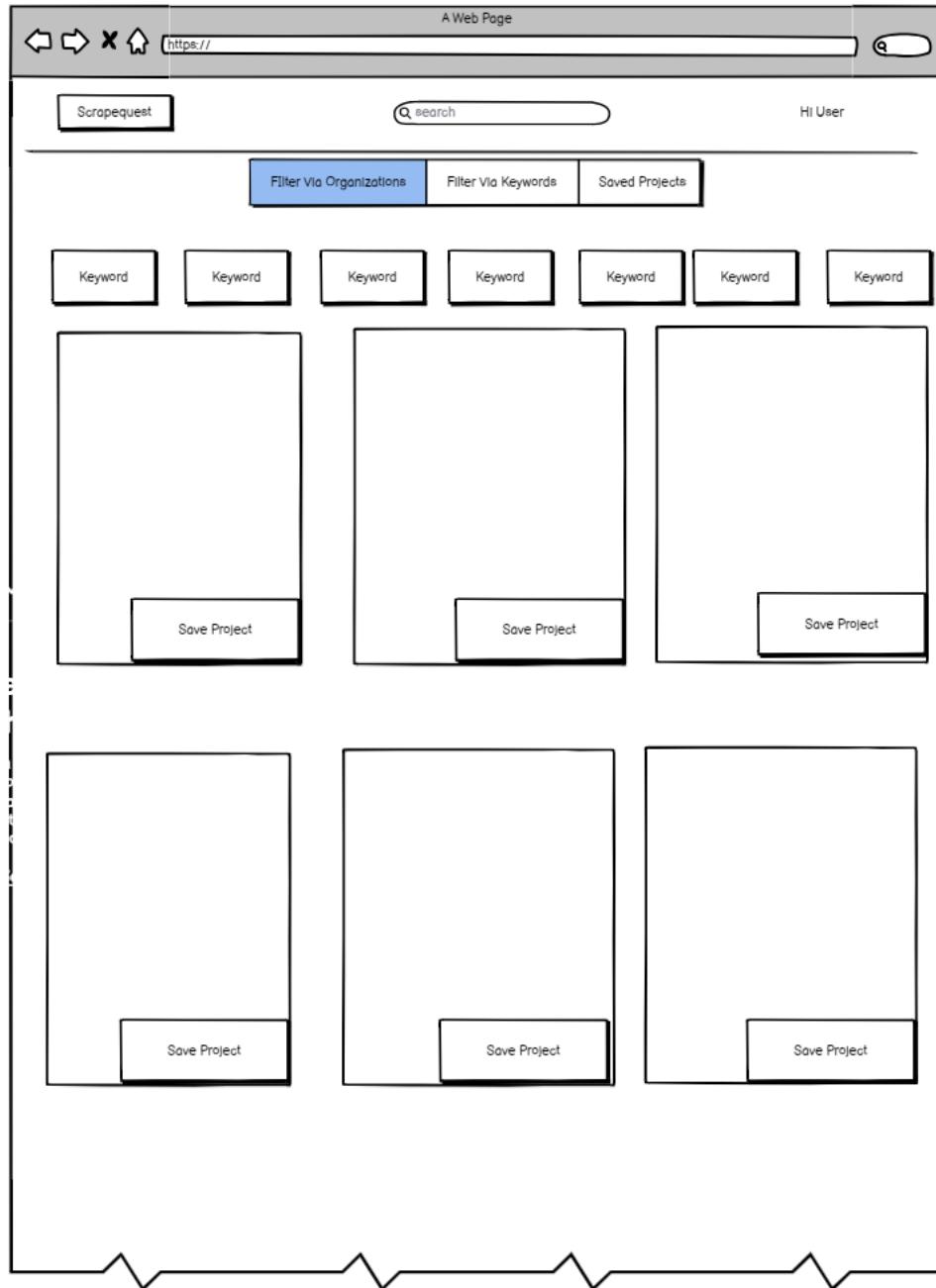


Figure 271 Wireframe for Filter via Keyword Page

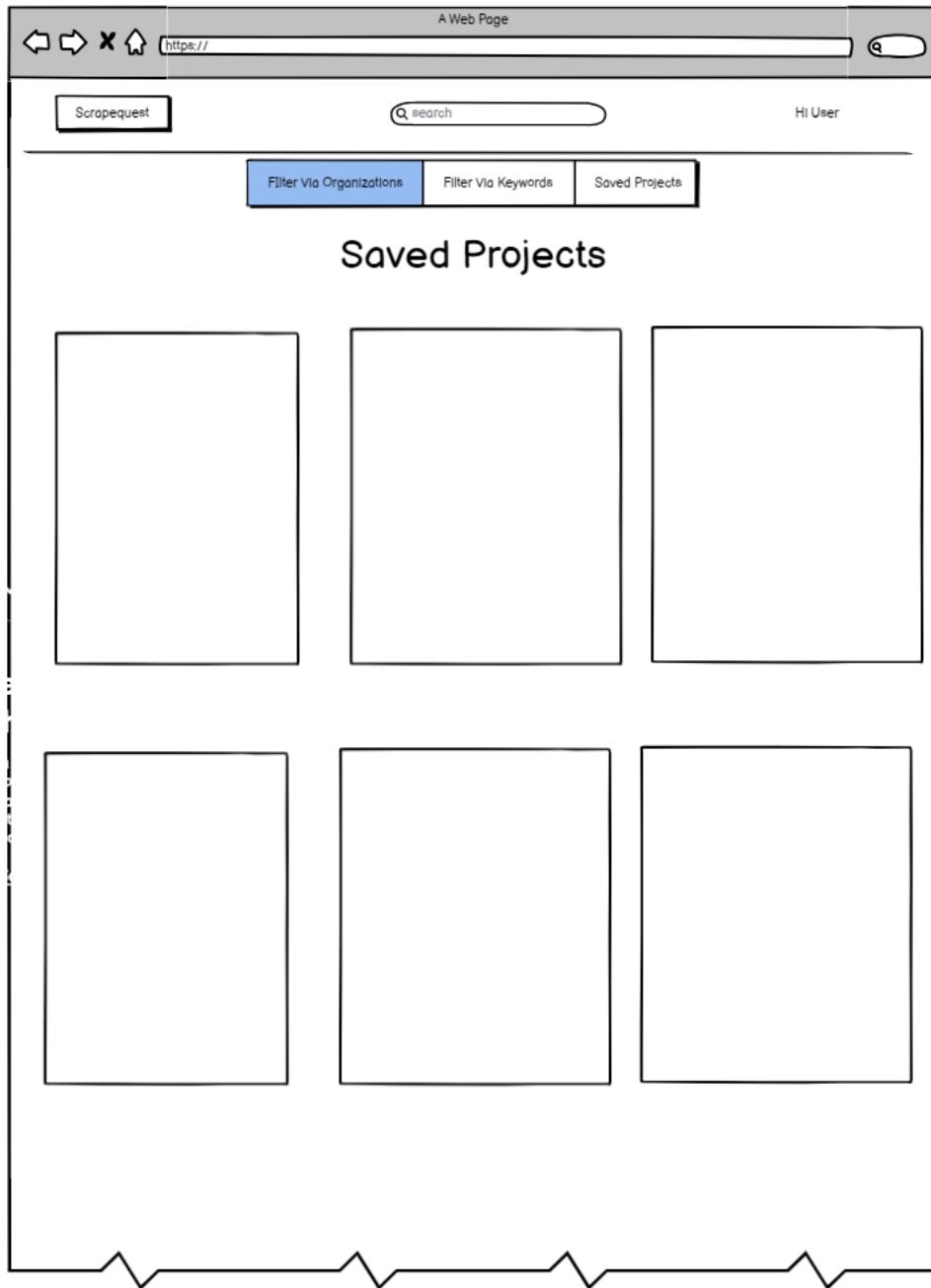


Figure 272 Wireframe for Saved Projects Page

8.4.6 Milestone Chart

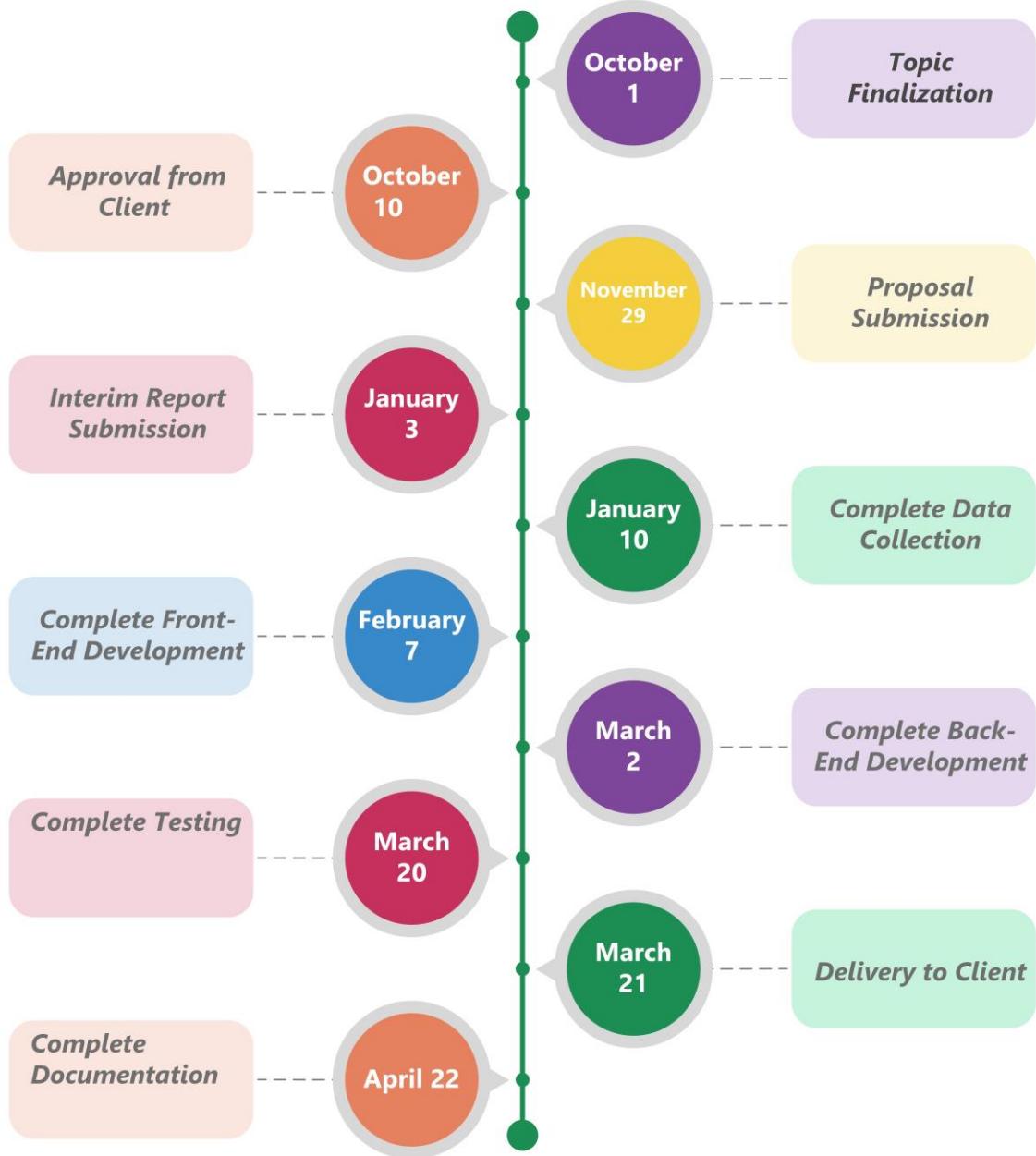


Figure 273 Milestone Chart

8.5 Appendix E: Screenshots of the System

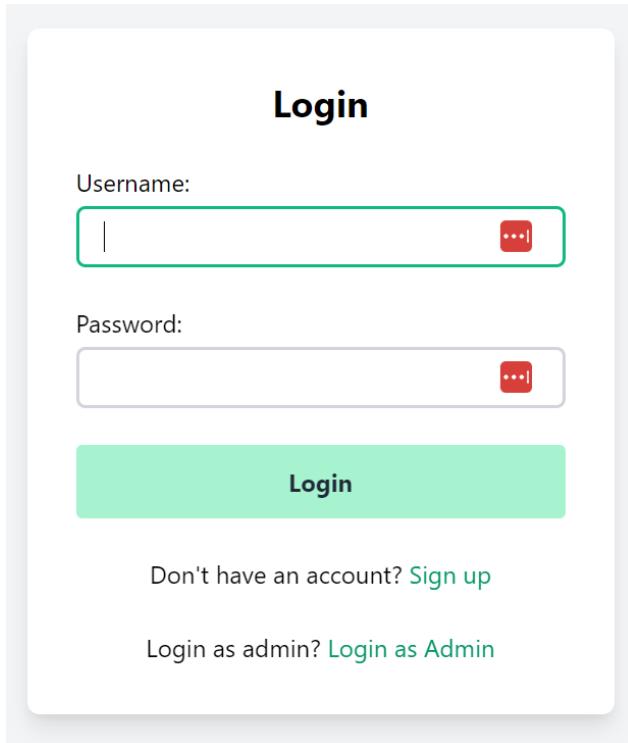


Figure 274 Login UI

Sign Up

Username:

Required. 150 characters or fewer. Letters, digits and @./+/-/_ only.

Email:

Required. Inform a valid email address.

Password:

Your password can't be too similar to your other personal information.
Your password must contain at least 8 characters.
Your password can't be a commonly used password.
Your password can't be entirely numeric.

Password confirmation:

Enter the same password as before, for verification.

Sign up

Already have an account? [Login](#)

Figure 275 Signup Page

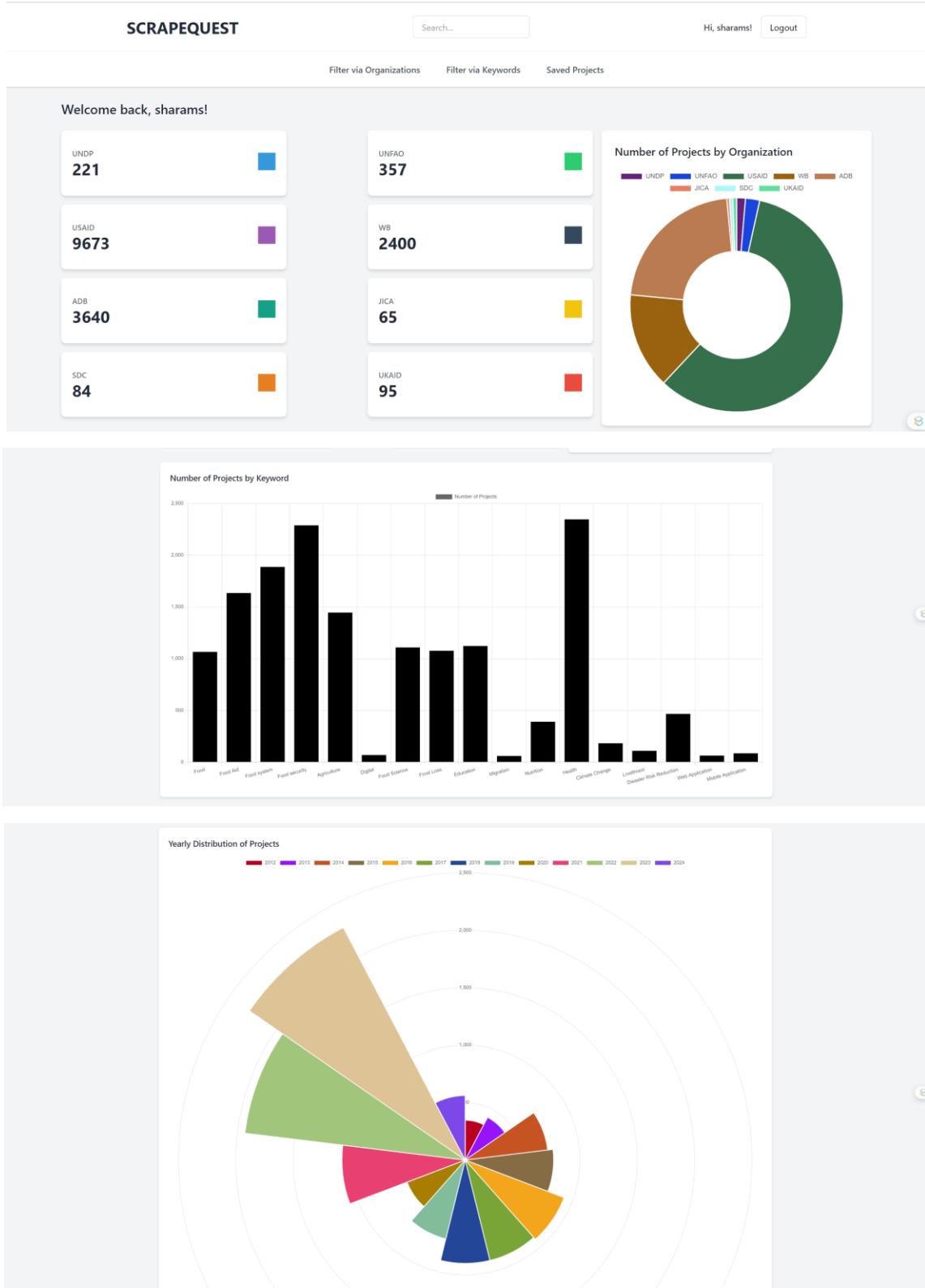


Figure 276 Homepage.

The screenshot shows the SCRAPEQUEST platform interface. At the top, there's a navigation bar with the title "SCRAPEQUEST", a search bar, and user account information ("Hi, sharams! Logout"). Below the navigation is a toolbar with several buttons: "Filter via Organizations" (which is highlighted with a red box), "Filter via Keywords", and "Saved Projects". Underneath the toolbar, there are tabs for different organizations: UNDP, UNFAO, USAID, JICA, ADB, WB, UKAID, and SDC. The main content area is titled "UNDP Projects" and displays a grid of project cards. Each card contains the project name, a "View Project" link, and a "Save Project" button. The projects listed are:

- Recovery & Resilience
- Enhancing human security through local climate actions
- European Union Support to Inclusive Federalism (EUSIF)
- Sambodhan: Temporary Basic Income
- Renewable Energy for Rural Livelihood
- Value Chain Development of Fruit and Vegetables Project (VCDP)
- Leaving No One Behind in Nepal's Green, Resilient, and Inclusive COVID-19 Recovery
- Strengthening Urban Preparedness, Earthquake Preparedness and Response in Western Regions of Nepal
- Support to Knowledge and Lifelong Learning Skills (SKILLS) Programme
- Sustainable Tourism for Livelihood Recovery
- Developing Climate Resilient Livelihoods in the Vulnerable Watershed in Nepal
- View Project

Figure 277 Page to Display Projects Filtered Via Organizations

The screenshot shows the SCRAPEQUEST interface. At the top, there is a search bar and a user greeting "Hi, sharams! Logout". Below the header are several filter options: "Filter via Organizations", "Filter via Keywords" (which is highlighted with a red box), and "Saved Projects". A horizontal row of buttons includes "Food" (highlighted with a red box), "Food Aid", "Food system", "Food security", "Agriculture", "Digital", "Food Science", "Food Loss", "Education", "Migration", "Nutrition", "Health", "Climate Change", "Livelihood", and "Disaster Risk Reduction". Below these are two application type buttons: "Web Application" and "Mobile Application". The main content area features a heading "Showing Results For Food" (highlighted with a red box). Three project cards are displayed:

- Project 1:** Name of the Project: Food and Nutrition Security Enhancement Project - P164319. Link to the project: [View](#). Procurement Details: Procurement of improved maize seed. Link to the Procurement: [View](#). Date of Publication: March 4, 2024. Organization: WB. Save Project button.
- Project 2:** Name of the Project: Food and Nutrition Security Enhancement Project - P164319. Link to the project: [View](#). Procurement Details: Reconstruction and Maintenance of office rooms & Program hall. Link to the Procurement: [View](#). Date of Publication: February 28, 2024. Organization: WB. Save Project button.
- Project 3:** Name of the Project: Food and Nutrition Security Enhancement Project - P164319. Link to the project: [View](#). Procurement Details: Procurement of Seasonable Summer Vegetables Seed (PCU Gorkha). Link to the Procurement: [View](#). Date of Publication: February 28, 2024. Organization: WB. Save Project button. A small circular icon with a question mark is next to this card.

Below this section is a heading "Related Results For Food". A single related project card is shown:

- Related Project:** Name of the Project: GCP /GLO/505/ROK: Implementation of Codex standards to support containment and reduction of foodborne AMR. Duration of the Project: 20-Jul-2021 - 19-Jun-2026. Link to the project: [View](#). Organization: UNFAO. Save Project button.

Figure 278 Page to Display Projects Filtered Via Keywords

SCRAPEQUEST

Search...

Hi, sharams! Logout

Filter via Organizations Filter via Keywords **Saved Projects**

Saved Projects

<p>Recovery & Resilience View Project</p> <p>Project Details: Remittances sent home by over 4 million migrant workers have significantly contributed to the economic development of Nepal. The proposed project will enhance the capacities of the Government of Nepal and strengthen mechanisms to better protect the rights of workers abroad. There will be a special focus on women's need and the adverse effects of climate change exacerbating the vulnerability of migrants. The project capitalises on Switzerland's longstanding engagement on labour migration in Nepal.</p> <p>Duration: 01.07.2023 - 31.12.2026</p>	<p>MCC Transmission Lines Activity Duration: Sector: Energy</p>	<p>Name of the Project: UNJP/NEP/084/UNJ Assessing the impact of the global crisis on the agriculture and food security situation in Nepal Duration of the Project: 20-Jun-2022 - 31-Mar-2023 Link to the project: View Organization: UNFAO</p>
<p>MiRiDew - Migrant Rights and Decent Work View Project</p> <p>Project Details: The goal is to support the Government of Nepal to conduct the 2021 census in line with international standards to obtain high quality data, reflecting Nepal's social, ethnic, religious and linguistic diversity and which can be used to better respond to the needs of the different population groups. Given its track record and emphasis on federal state building, Switzerland has a key interest that the census is adapted to the federal context and gender equality and social inclusion principles are followed.</p> <p>Duration: 01.07.2021 - 31.12.2024</p>	<p>2021 Population and Housing Census of Nepal View Project</p> <p>Project Details: The Fleming Fund helps low- and middle-income countries to fight antimicrobial resistance. A management agent has been appointed to deliver: country grants 24 low- and middle-income countries, regional grants in West Africa, East and Southern Africa, South Asia and South East Asia, and a global fellowships programme. These initiatives aim to improve laboratory capacity and diagnosis as well as data and surveillance of antimicrobial resistance (AMR).</p> <p>Start Date: 2016-10-10</p>	

Figure 279 Saved Projects Page.

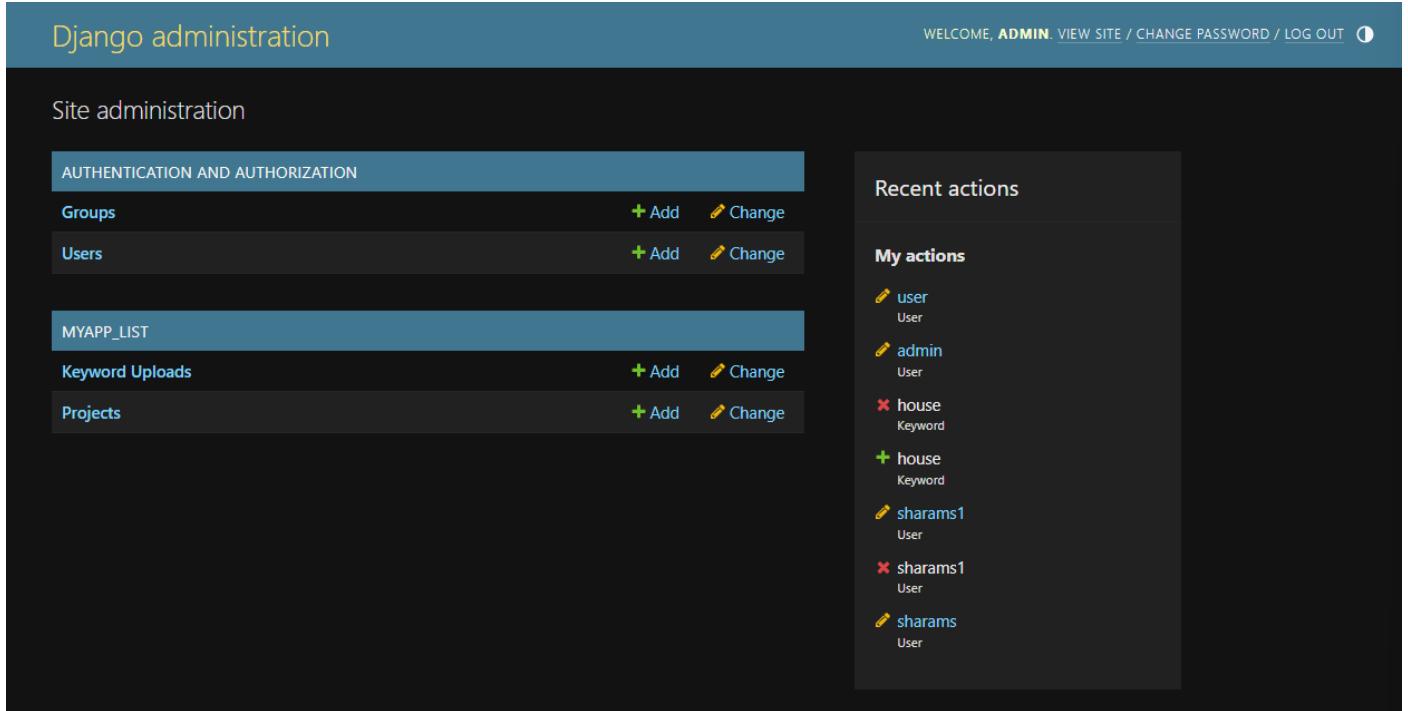


Figure 280 Admin Panel

8.6 Appendix E: User Feedback

8.6.1 User Feedback Form

User Feedback Form - Scrapequest

B I U ↲ ✖

This survey is intended to collect your feedback after using the Scrapequest tool to better understand its performance and impact on the project discovery process.

Tailored Solution for Your Needs



How satisfied are you with Scrapequest? *

- Very satisfied
- Satisfied
- Neutral
- Unsatisfied
- Very unsatisfied

How easy was it to learn to use Scrapequest? *

- Very easy
 - Easy
 - Neither easy nor difficult
 - Difficult
 - Very difficult
-

How would you rate the user interface and design of Scrapequest?

- Excellent
 - Good
 - Fair
 - Poor
 - Very poor
-

How effectively does Scrapequest help you identify relevant projects? *

- Extremely effectively
- Very effectively
- Moderately effectively
- Slightly effectively
- Not at all effectively

Which features of Scrapequest do you find most valuable? *

- Real-time updates on new projects
- Data filtering to match our strategic goals
- Automated verification of data reliability
- Cost-effectiveness
- Ease of integration with our current systems
- Other (please specify): _____

Have you experienced any technical issues while using Scrapequest? *

Please Specify

- No
- Other...

7. Has Scrapequest affected the amount of time you spend on project discovery? *

- Greatly reduced time
- Somewhat reduced time
- No change
- Somewhat increased time
- Greatly increased time

How has Scrapequest impacted your productivity? *

- Significantly improved
- Moderately improved
- Slightly improved
- No impact
- Slightly worsened
- Significantly worsened

How satisfied are you with the support and training materials provided for Scrapequest? *

- Very satisfied
- Satisfied
- Neutral
- Unsatisfied
- Very unsatisfied

Would you recommend Scrapequest to colleagues or other companies? *

- Definitely would recommend
- Probably would recommend
- Might or might not recommend
- Probably would not recommend
- Definitely would not recommend

What improvements would you like to see in future updates of Scrapequest?

Long-answer text

Any other comments or feedback you'd like to provide about Scrapequest?

Long-answer text

Figure 281 User Feedback Form

8.6.2 Sample of Filled User Feedback Forms

Responses cannot be edited

User Feedback Form - Scrapequest

This survey is intended to collect your feedback after using the Scrapequest tool to better understand its performance and impact on the project discovery process.

* Indicates required question

Tailored Solution for Your Needs

 SCRAPEQUEST

How satisfied are you with Scrapequest? *

Satisfied

Very satisfied

Neutral

Unsatisfied

Very unsatisfied

How easy was it to learn to use Scrapequest? *

- Very easy
- Easy
- Neither easy nor difficult
- Difficult
- Very difficult

How would you rate the user interface and design of Scrapequest?

- Excellent
- Good
- Fair
- Poor
- Very poor

How effectively does Scrapequest help you identify relevant projects? *

- Extremely effectively
 - Very effectively
 - Moderately effectively
 - Slightly effectively
 - Not at all effectively
-

Which features of Scrapequest do you find most valuable? *

- Real-time updates on new projects
- Data filtering to match our strategic goals
- Automated verification of data reliability
- Cost-effectiveness
- Ease of integration with our current systems
- Other (please specify): _____

Have you experienced any technical issues while using Scrapequest? *

Please Specify

No

Other: _____

7. Has Scrapequest affected the amount of time you spend on project discovery? *

Greatly reduced time

Somewhat reduced time

No change

Somewhat increased time

Greatly increased time

How has Scrapequest impacted your productivity? *

- Significantly improved
- Moderately improved
- Slightly improved
- No impact
- Slightly worsened
- Significantly worsened

How satisfied are you with the support and training materials provided for Scrapequest? *

- Very satisfied
- Satisfied
- Neutral
- Unsatisfied
- Very unsatisfied

Would you recommend Scrapequest to colleagues or other companies? *

- Definitely would recommend
 - Probably would recommend
 - Might or might not recommend
 - Probably would not recommend
 - Definitely would not recommend
-

What improvements would you like to see in future updates of Scrapequest?

Any other comments or feedback you'd like to provide about Scrapequest?

Figure 282 Sample of Filled User Feedback Form

8.6.3 User Feedback Results

User Feedback Form - Scrapequest

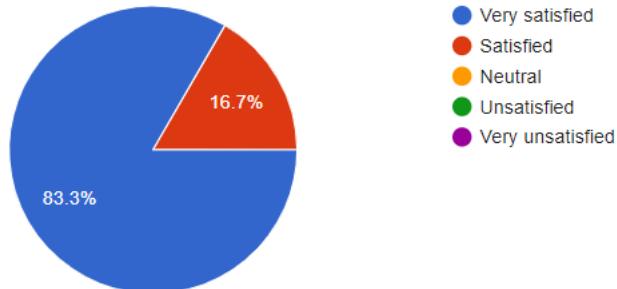
6 responses

[Publish analytics](#)

 Copy

How satisfied are you with Scrapequest?

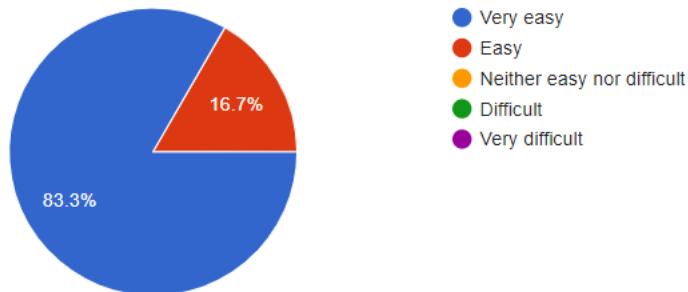
6 responses



How easy was it to learn to use Scrapequest?

6 responses

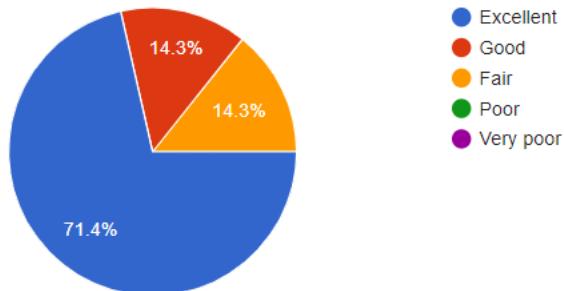
 Copy



[Copy](#)

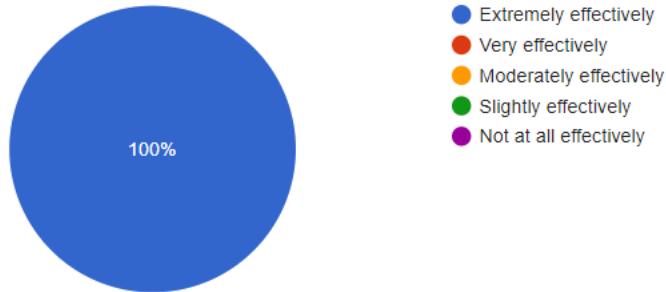
How would you rate the user interface and design of Scrapequest?

6 responses

[Copy](#)

How effectively does Scrapequest help you identify relevant projects?

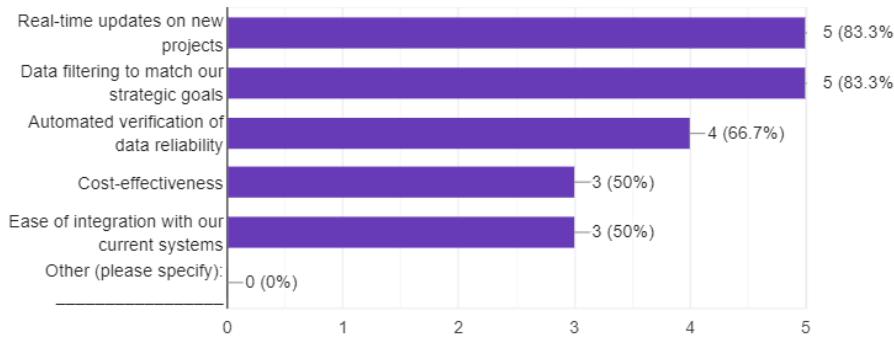
6 responses



[Copy](#)

Which features of Scrapequest do you find most valuable?

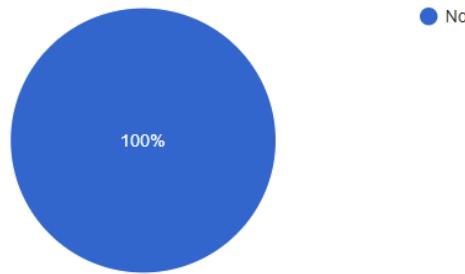
6 responses



Have you experienced any technical issues while using Scrapequest?

[Copy](#)

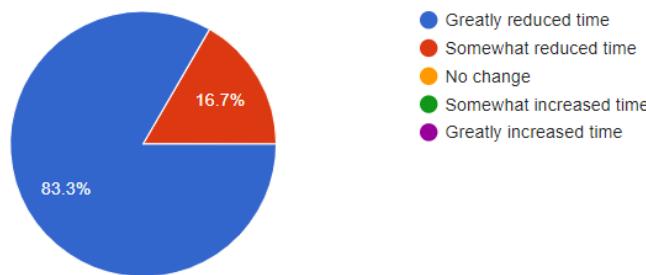
6 responses



7. Has Scrapequest affected the amount of time you spend on project discovery?

[Copy](#)

6 responses



How has Scrapequest impacted your productivity?

 Copy

6 responses



- Significantly improved
- Moderately improved
- Slightly improved
- No impact
- Slightly worsened
- Significantly worsened

How satisfied are you with the support and training materials provided for Scrapequest?

 Copy

6 responses

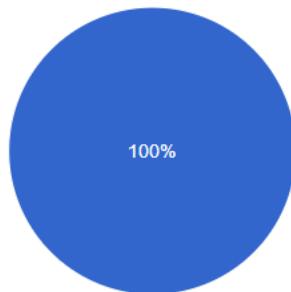


- Very satisfied
- Satisfied
- Neutral
- Unsatisfied
- Very unsatisfied

Would you recommend Scrapequest to colleagues or other companies?

 Copy

6 responses



- Definitely would recommend
- Probably would recommend
- Might or might not recommend
- Probably would not recommend
- Definitely would not recommend

What improvements would you like to see in future updates of Scrapequest?

3 responses

no improvements

no

no

Any other comments or feedback you'd like to provide about Scrapequest?

3 responses

no

nope

Figure 283 User Feedback Form Results

8.6.3.1 Summary of Results

The results of the survey conducted to gauge user experience with the Scrapequest tool reveal several insights regarding its effectiveness and user satisfaction. The majority of respondents (83.3%) reported being satisfied or very satisfied with the tool, indicating a strong positive reception. In terms of usability, the same proportion found it easy or very easy to learn how to use Scrapequest.

When evaluating the user interface and design, 71.4% of the users rated it as fair, suggesting some room for improvement in this area. However, the tool's functionality in helping identify relevant projects was highly praised, with all respondents affirming that it does so effectively.

Users highlighted several features as most valuable, with real-time updates on new projects, data filtering aligned with strategic goals, and automated verification of data reliability each receiving high appreciation from 83.3% of respondents. Notably, no technical issues were reported by any of the users.

The impact on productivity was unanimously positive, with all participants noting an improvement. Similarly, all respondents were satisfied with the support and training materials provided and would definitely recommend Scrapequest to colleagues or other companies.

Overall, the survey underscores the significant positive impact of Scrapequest on the project discovery process, with high levels of user satisfaction and effectiveness in enhancing productivity. Despite the overall positive feedback, the response time of the app received mixed reviews, suggesting an area for future enhancements.

8.7 Appendix G: Future Work

8.7.1 Readings for Future Work

8.7.1.1 Cloud-Based Auto-Scaling

- What is Auto-Scaling?
 - Available from:

<https://www.ibm.com/topics/autoscaling>
- An Efficient Approach for Resource Auto-Scaling in Cloud Environments
 - DOI:10.11591/ijece.v6i5.10639
 - Available from:

https://www.researchgate.net/publication/305180276_An_Efficient_Approach_for_Resource_Auto-Scaling_in_Cloud_Environments
- A Novel Approach for Optimization Auto-Scaling in Cloud Computing Environment
 - DOI: 10.5815/ijcnis.2015.11.05
 - Available from:

https://www.researchgate.net/publication/307851611_A_Novel_Approach_for_Optimization_Auto-Scaling_in_Cloud_Computing_Environment
- Automated Setup of Multi-Cloud Environments for Microservices Applications
 - Available from: <https://inria.hal.science/hal-01312606/document>

8.7.1.2 Automated Compliance Checking System

- NLP-based Automated Compliance Checking of Data Processing Agreements against GDPR
 - Available from: <https://arxiv.org/pdf/2209.09722.pdf>
- Semantic NLP-based Information Extraction from Construction Regulatory Documents for Automated Compliance Checking
 - Available from:

https://polytechnic.purdue.edu/sites/default/files/files/Zhang_and_El-Gohary_IE_Draft.pdf

8.7.1.3 Multilingual Data Handling Capability

- Harnessing NLP Techniques in the Processes of Multilingual Content Management
 - Available from: <https://aclanthology.org/E12-2002.pdf>
- Lifelong Learning Natural Language Processing Approach for Multilingual Data Classification
 - Available from:
https://www.researchgate.net/publication/361500367_Lifelong_Learning_Natural_Language_Processing_Approach_for_Multilingual_Data_Classification

8.7.1.4 Resource Optimization Framework

- Research on Performance Optimization of Web Application System based on JAVA EE
 - Available from:
https://www.researchgate.net/publication/338424087_Research_on_Performance_Optimization_of_Web_Application_System_based_on_JAVA_EE
- Performance Gains from Web Performance Optimization
 - Available from: <https://www.scitepress.org/papers/2016/59036/59036.pdf>
- Progressive Web Apps (PWA)
 - Available from: <https://web.dev/explore/progressive-web-apps>

8.7.1.5 Adaptive Scraping System for Dynamic Websites

- Intelligent and Adaptive Web Data Extraction System Using Convolutional and Long Short-Term Memory Deep Learning Networks
 - Available from: <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9523501>
- Web Scraping using Machine Learning
 - Available from:
<https://www.diva-portal.org/smash/get/diva2:1468583/FULLTEXT01.pdf>

8.8 Appendix H: Requirements from Client

8.8.1 Client requirements

Functional Requirements:

- The platform must support secure user authentication and authorization mechanisms.
- The system should automate scraping project information from organizations like the United Nations and World Bank, and other agencies, including project descriptions, keywords, and metadata.
- Users can easily discover projects by filtering them based on predefined keywords like food science, agriculture, and climate change using a robust search engine.
- Create a user-friendly web interface for easy project navigation and exploration, incorporating visualizations and data representations to enhance user experience.
- Adhere to data privacy laws and regulations.
- Provide comprehensive technical documentation and create user guides to facilitate effective platform usage.

Non-Functional Requirements:

- Design the system architecture to scale efficiently.
- Ensure availability and reliability.
- Ensure compatibility.

8.8.2 Client Approval Letter

Raman Pradhananga
Vertex Special Technologies
Sanepa, Lalitpur
October 6, 2023

Sharams Kunwar
Scrapequest
Islington College
Kamalpokhari
Kathmandu

Subject: Permission Granted for the role of Client.

Dear Sharams Kunwar:

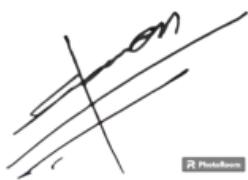
I have reviewed your request for the role of client, and I am pleased to inform you that I have agreed to be your client for the project of the application. Taking reference from our recent verbal meeting I am willing to provide the requirements and the necessary information that I need in the Scrapequest application.

If you have any questions, or if I can be of further service to you, please call me or arrange a meeting for which we can discuss the issues and present the progress of the application development till date.

Sincerely,

Raman Pradhananga

Sharams Kunwar
Scrapequest

A handwritten signature in black ink, appearing to read "Sharams Kunwar". It is written in a cursive style with some loops and variations in thickness.

Signature

[Back to Report](#)

8.9 Appendix I: Considered Methodology

8.9.1 Waterfall Methodology

The Waterfall model represents a linear and sequential approach to project management. In the initial stages of software development, requirements are collected from stakeholders and customers. Subsequently, a sequential project plan is formulated to fulfill these requirements. The phases of the Waterfall model are executed in a strict sequential order, necessitating the completion of each preceding phase before progressing to the next (Devaraj, 2023). However, a notable limitation of this model is its inability to adapt to changing requirements, having high amount of risk and uncertainty (tryqa, 2023).

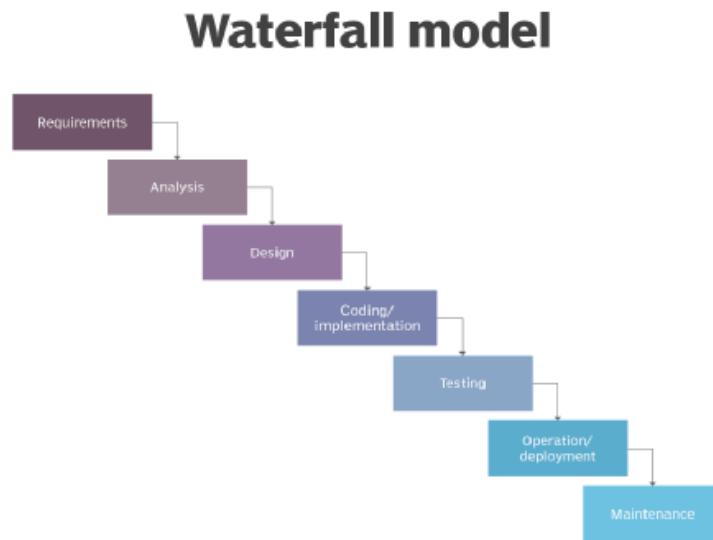


Figure 284 Waterfall Methodology (Lutkevich, 2023)

The waterfall model is not suitable for the academic project because it requires us to follow each stage of development one after the other, making it difficult to adapt to changes in requirements from the client. Since planning is done upfront, any future changes are not easily accommodated. Additionally, the short timeframe of the project may lead to insufficient time for thorough testing, as testing is only conducted at the end. This increases the risk of not catching bugs early in the development process.

8.9.2 Prototype Methodology

The Prototype methodology is centered around client satisfaction, emphasizing the creation of a prototype based on client requirements. The project is deemed finished only when the client expresses contentment with the prototype. This approach proves beneficial when project requirements are uncertain, enabling the selection of various features to construct the desired product. The development team constructs a prototype, undergoes evaluation with the client, and repeats this cycle until the client approves the final product (Martin, 2023).

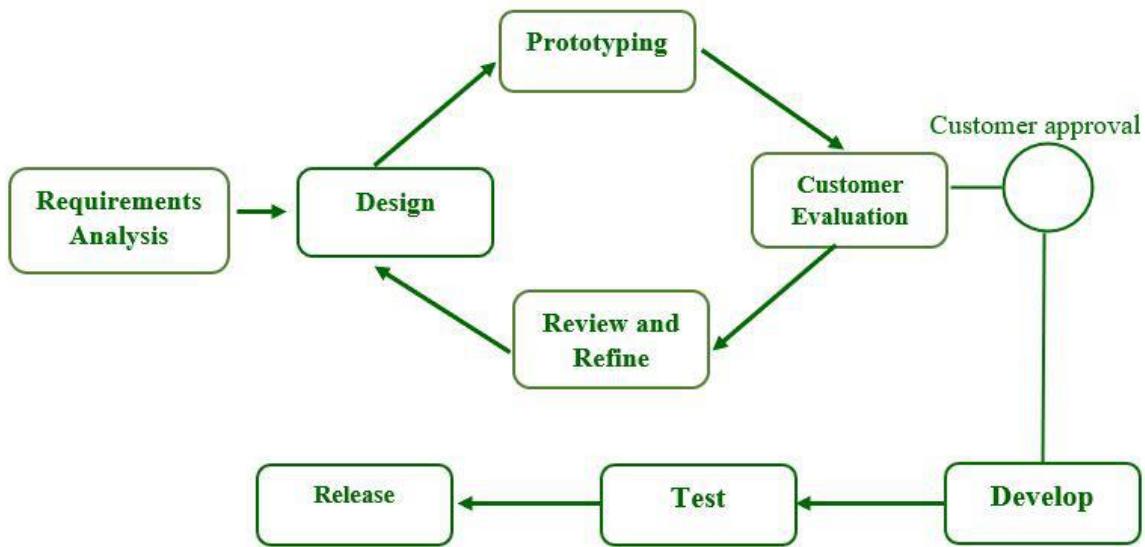


Figure 285 Prototyping Model (geeksforgeeks, 2023)

The prototype methodology also mayn't be the best fit for the academic project due to several reasons. First, managing multiple prototypes can be challenging, making it difficult to follow a systematic schedule. This can lead to confusion and hinder the development process. Moreover, the prototype approach often requires more resources, and in a single-person project, these resources may not be readily available. This lack of resources can impede progress and limit the effectiveness of the prototype methodology for our specific project requirements.

8.9.3 Spiral Model

The Spiral Model is an iterative approach to software development designed to mitigate project risks. In its structure, it shares similarities with both the Waterfall model and the Prototype model but follows a spiral iteration (geeksforgeeks, 2023). The development team initiates the process by constructing a scaled-down version of the product, evaluating associated risks, and documenting strategies to address them. The decision to advance to the subsequent step is then made. This cycle persists until the final product is deemed ready. The Spiral Model is recognized for its complexity, and in theory, it has the potential to continue indefinitely (Gurendo, 2023).

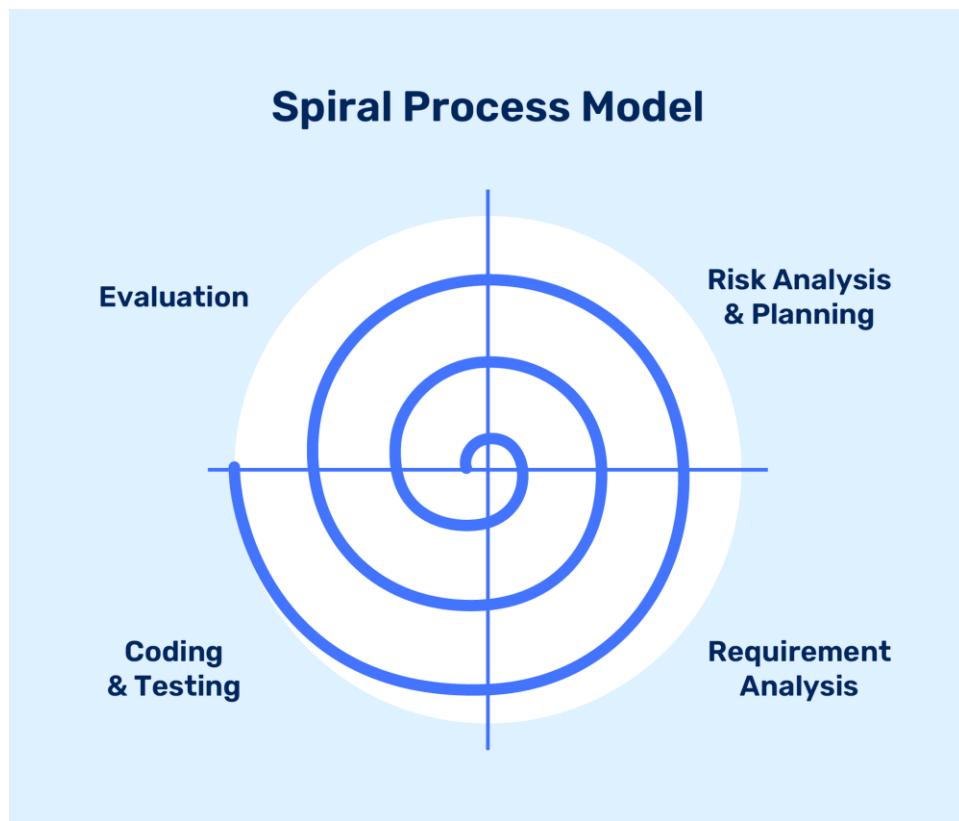


Figure 286 Spiral Process Model (Gladun, 2023)

The spiral model is also not well-suited for this academic project, primarily because of resource constraints in a single-person development scenario. This model involves iterative cycles, and due to limited resources, it becomes challenging to allocate time and effort for each spiral effectively. Additionally, scheduling becomes a significant issue, as the project might

unintentionally spiral into an infinite loop without clear timelines or milestones. The need for constant reassessment and refinement in each cycle can be time-consuming and may not align with the constraints of a single-person project with limited resources.

8.9.4 Unified Process

The Unified Process is a software development process framework characterized by its iterative, use case-driven, architecture-centric, and incremental approach. It offers the flexibility to be tailored to specific projects or organizations (Osis & Donins, 2017). Within the Unified Process, Unified Modeling Language (UML) is employed for documentation and design purposes. Examples of implementations of the Unified Process include Rational Unified Process (RUP), Open Unified Process (OUP), and Agile Unified Process (AUP) (bartleby, 2023).

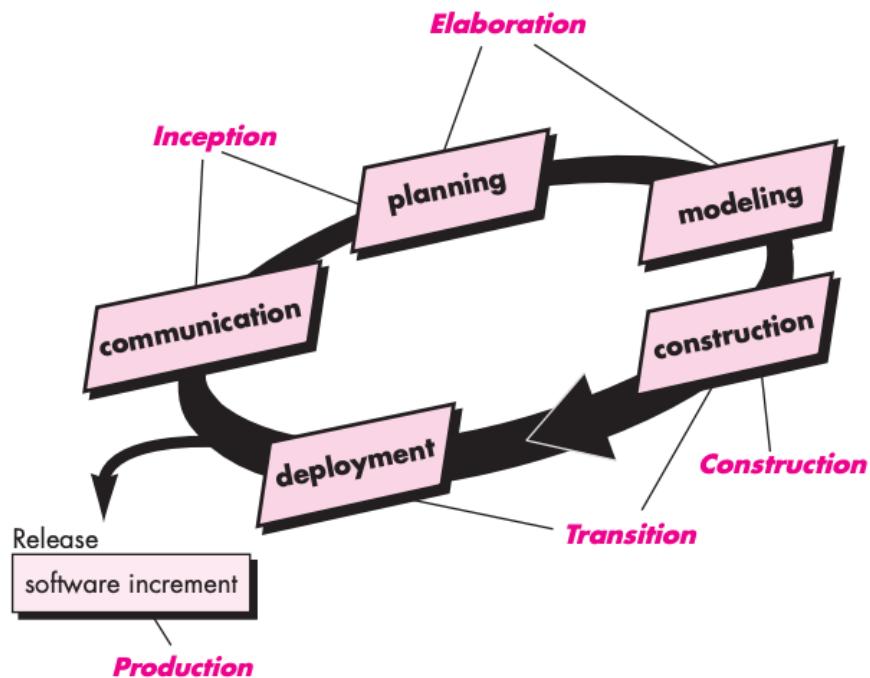


Figure 287 Unified Process (Abaroa, 2019)

[Back to Report](#)

8.10 Appendix J: SRS Document

8.10.1 Project Title: Scrapequest

8.10.2 Category: Web Application

8.10.3 Introduction

8.10.3.1 Purpose

This Software Requirements Specification (SRS) outlines the requirements for the development of Scrapequest, a comprehensive project discovery platform. It aims to identify and explore international development projects in fields such as food science, digitalization, education, etc.

8.10.3.2 Intended Audiences and Reading Suggestions

This document is intended for developers, project managers, system administrators, and other stakeholders involved in the development and deployment of Scrapequest. It is recommended to start with the overview sections and then proceed to the sections relevant to each reader type.

8.10.3.3 Project Scope

Scrapequest is designed to address challenges related to project discovery in the international development sector. The platform focuses on streamlining the process of identifying and exploring projects from various donor organizations, such as the United Nations, World Bank, and others. The scope of Scrapequest includes providing a comprehensive solution to the following issues faced by users, including employees and administrators:

For Users (Employees):

- Difficulty in identifying relevant projects aligned with their goals.
- Inefficiency in project discovery and exploration.
- Limited tools for tracking and revisiting potential opportunities.

Web Application Features:

- Project Search and Discovery.
- Filtering Projects using predefined keywords based on organizational goals.
- Save and Retrieve Project.
- View and Export Project Details.

For Administrative Body:

- Challenges in managing fields, user accounts, and system configurations.
- Limited insights into user interactions, saved projects, and system analytics.

Web Application Features:

- Ability to add, update and delete/ manage user accounts.
- Update and manage predefined fields.
- View saved/ shortlisted projects.
- View and Export Project Details.

Overall Project Scope:

Scrapequest provides a unified platform for efficient project discovery and management. It encompasses a user-friendly Android application for project seekers and a robust web application for system administrators. The aim is to enhance the user experience, improve project discovery efficiency, and provide valuable insights through analytics. The features listed above contribute to a comprehensive solution that addresses the challenges faced by users and administrators in the international development project discovery domain.

8.10.3.4 Existing System

The current project discovery system relies on manual processes, involving:

- Extensive paperwork and human efforts for searching and collating potential projects.
- Time-consuming practices that divert valuable resources from core business functions.
- Drawbacks in user-friendliness and efficiency, limiting the system's effectiveness.
- The risk of inaccuracies in data collection, missed opportunities, and misalignment with organizational objectives.
- Substantial investments in time and personnel for manual project data management.
- Challenges in navigating the overwhelming volume of project-related information available online.
- Lack of adequate security measures, posing a risk of data loss due to mismanagement.

To address these issues and stay competitive, a transition to automated tools for project discovery is crucial. This shift can reduce costs, enhance accuracy, and improve overall competitiveness in the dynamic business landscape.

8.10.3.5 Proposed System

The main aim of the proposed system is to replace the existing manual project discovery tasks to attain efficiency and gain competitive advantage by reducing costs and achieve accuracy. It can be very fruitful including advantages as such below:

- Reduces the need for extensive manual efforts, optimizing human resources for more strategic tasks.
- Enables the use of data analytics to extract valuable insights, aiding in more informed decision-making for project selection.
- Positions the organization ahead of competitors by enabling faster, more accurate, and data-driven project discovery and decision-making processes.
- Facilitates better collaboration among team members by providing a centralized platform for sharing and discussing potential projects.
- Adaptable to the changing needs and scale of the organization, ensuring continued efficiency as the business grows.
- Allows for customization based on specific criteria, tailoring the project discovery process to align with organizational goals and preferences.
- Provides real-time information on potential projects, enabling quick decision-making and responsiveness to market changes.
- Automated tools accelerate the project discovery process, saving time compared to manual searches.

8.10.4 Overall Description

8.10.4.1 System Features

Some of the features of the system are listed below:

8.10.4.1.1 Login/ Register

8.10.4.1.1.1 Authentication and Authorization

Description and Priority:

Secure access to the application with user authentication and authorization.

(Priority: **High**)

Stimulus/Response Sequences

- Users/Admin register with valid credentials.
- Users/Admin log in with registered credentials.

Functional Requirements:

- **REQ-1:** Users/Admin can register with a valid username and password and log in with registered credentials.

8.10.4.1.2 Project Discovery (User)

8.10.4.1.2.1 Project Search and Discovery

Description and Priority:

Enable users to search for projects aligned with their goals, ensuring efficient project discovery.

(Priority: **High**)

Stimulus/Response Sequences

- Users enter search criteria.
- System retrieves relevant projects.
- Users view project details.

Functional Requirements:

- **REQ-2:** Users can search for projects using predefined keywords.
- **REQ-3:** Efficient filtering based on predefined keywords.
- **REQ-4:** Intuitive user interface for a seamless search experience.

8.10.4.1.2 Personalized Project Management

Description and Priority:

Provide users with tools for managing their discovered projects efficiently. (Priority: **High**)

Stimulus/Response Sequences

- Users view Project Details.
- Users can save projects for future references or shortlist them.
- System retrieves saved projects.

Functional Requirements:

- **REQ-5:** Ability to view Project Details.
- **REQ-6:** Ability to save Projects.
- **REQ-7:** Ability to revisit saved Projects.

8.10.4.1.3 Admin Management (Admin)

8.10.4.1.3.1 User Management

Description and Priority:

Enable users to search for projects aligned with their goals, ensuring efficient project discovery.
(Priority: **High**)

Stimulus/Response Sequences

- Admin adds, updates, or deletes users.

Functional Requirements:

- **REQ-8:** Admin can add, update, and delete users.

8.10.4.1.3.2 Predefined Field Configuration

Description and Priority:

Enable administrators to update and manage predefined fields.

(Priority: **High**)

Stimulus/Response Sequences

- Admin updates and manages predefined fields.

Functional Requirements:

- **REQ-9:** Admin can update and manage predefined fields.
- **REQ-10:** Admin can edit predefined fields.

8.10.4.1.3.3 Analytics

Description and Priority:

Provide administrators with insights into user interactions, saved projects, and system analytics.

(Priority: **Medium**)

Stimulus/Response Sequences

- User and Admin monitors trends related to specific fields and organizations.
- Admin tracks saved projects and export project details.

Functional Requirements:

- **REQ-11:** User/Admin can monitor trends related to specific fields and organizations.
- **REQ-12:** Admin can track saved projects and export project details.

8.10.4.2 User Class and Characteristics

Scrapequest web application will be used by two users, i.e., Normal Users (Employee) and Admin (Administrative Body). According to the users their roles are listed below:

UC1: Normal Users (Employees)**UC1.1**

Can view project details.

UC1.2

Can navigate projects.

UC1.3

Can search for projects.

UC1.4

Can save/shortlist projects for future references.

UC1.5

Can revisit saved projects.

UC1.6

Can monitor trends related to specific fields and organization.

UC2: Admin (Administrative Body)**UC 2.1**

Can add, update, or delete users and manage user roles.

UC2.2

Can add, delete, or edit predefined fields.

UC2.3

Can modify the list of donor organizations.

UC2.4

Monitor trends related to specific fields and organizations.

UC2.5

Can track saved projects and export project details.

8.10.4.3 Operating Environment

In the present context, Scrapequest is focused on a Web platform for both the user and administrative body of the client. Further planning is to make this run on different platforms like IOS and Android or integrate this tool into the existing project management system used by the client.

8.10.4.4 Assumptions and Dependencies

AS1: The main audience of this application is the client, Vertex Special Technologies.

AS2: Reports and documents of this system are accessible to developers and the researchers interested in developing similar kinds of system.

8.10.4.5 Functional Requirements

8.10.4.5.1 Authentication and Authorization

Req. ID	Requirement Description	
REQ-1		Users/Admin can register with a valid username and password and log in with registered credentials.
Sys. Req. ID	System Requirement	
SR.01	The system should offer a user registration process that requires unique usernames and strong password criteria for secure login.	
SR.02	Users must log in using their registered credentials, authenticated, and authorized according to the stored credentials in the Users table of the database.	
SR.03	System should allow admins to log in using secure, unique admin credentials for accessing administrative functionalities.	
SR.04	System should display appropriate error messages for unsuccessful login attempts or registration failures.	
SR.05	System should define roles with specific access permissions, with administrators having administrative functionalities and regular users having project-related features.	
SR.06	System should allow users to manage their profiles, including updating passwords or other relevant information.	

Table 63 Functional Requirement Description of Authentication and Authorization

8.10.4.5.2 Search Projects

Req. ID	Requirement Description	
	Sys. Req. ID	System Requirement
REQ-2	Efficient filtering based on predefined keywords.	
	SR.07	System must have secure user authentication and authorization, to ensure only authorized users can access search functionality.
	SR.08	System should offer an admin interface for managing and configuring predefined keywords related to project fields, allowing for the addition, removal, or modification of these keywords.
	SR.09	Users can search for projects by entering predefined keywords, which the system must validate and process to match against a predefined list.
	SR.10	System should provide robust search engine that efficiently filters projects using selected keywords, considering multiple options, and providing relevant results.
	SR.11	System should provide a user-friendly web interface that simplifies keyword search input and submission.
	SR.12	The search results should be clear, organized, and provide relevant project information, with options for users to view detailed information.

Table 64 Functional Requirement Description of Searching Projects

8.10.4.5.3 Filter Projects

Req. ID	Requirement Description	
	Sys. Req. ID	System Requirement
REQ-3	Users can search for projects using predefined keywords.	
	SR.13	System should offer a user-friendly interface with a search feature, allowing users to input keywords for project filtering and display predefined keywords related to their work fields.
	SR.14	The system shall implement a robust search engine capable of efficiently processing keyword-based queries.
	SR.15	The system must align keywords with predefined fields of work, allowing admins to associate keywords with specific fields and update these associations as needed.
	SR.16	System should allow users to filter projects by multiple keywords simultaneously and sort results by relevance, date, or other relevant criteria using the system.
	SR.17	The system should dynamically update project listings based on applied filters.

Table 65 Functional Requirement Description of Filtering Projects

8.10.4.5.4 Seamless User Interaction

Req. ID	Requirement Description	
	Sys. Req. ID	System Requirement
	SR.18	System should offer a user-friendly search engine for projects based on predefined keywords, offering an intuitive interface with filtering options.
	SR.19	System should provide a user-friendly interface for project navigation and exploration, incorporating features like project categorization, sorting options, and clear display of project details like title, description, and organization.
	SR.20	System should enhance user experience by incorporating visualizations and data representations, such as charts and graphs, to present project data trends, keyword distributions, and other relevant insights.
	SR.21	System should implement a responsive design that adapts to various screen sizes and devices.
	SR.22	System should ensure compatibility with major web browsers to guarantee a consistent experience for users using different browsers.

Table 66 Functional Requirement Description of Seamless User Interface

8.10.4.5.5 Project Details

Req. ID	Requirement Description	
	Sys.	System Requirement
	Req. ID	
REQ-5	Ability to view Project Details.	
	SR.23	The system should enable users to access detailed project information such as Title, Link, Description, Duration, Field of Work, and Organization.
	SR.24	The system should provide a clear, accessible, and comprehensible way to view project details, allowing users to return to the previous page after viewing.
	SR.25	The system should implement secure authentication mechanisms to restrict access to project details to only authorized users.
	SR.26	System should retrieve project details efficiently by optimizing database queries to handle a large number of projects.

Table 67 Functional Requirement Description of Project Details

8.10.4.5.6 Save Projects

Req. ID	Requirement Description	
	Sys.	System Requirement
	Req. ID	
REQ-6	Ability to view save Projects for future reference.	
	SR.27	The system must ensure user authentication before allowing them to save projects, and they should have appropriate authorization based on their roles.
	SR.28	The system should provide a user-friendly and intuitive web interface with a "Save Project" button easily accessible on the webpage.
	SR.29	The system should store saved projects in the SavedProjects table, ensuring each user has a unique list of these projects.
	SR.30	The system should send a confirmation message to the user after saving a project and display appropriate error messages to assist the user in resolving any issues.
	SR.31	System should allow users to manage their saved projects in a dedicated section of their profile, offering options to view, edit, and remove them.

Table 68 Functional Requirement Description of Saving Projects

8.10.4.5.7 Retrieve Saved Projects

Req. ID	Requirement Description	
Sys. Req. ID	System Requirement	
REQ-7	Ability to revisit saved Projects.	
SR.32		
SR.33		
SR.34		
SR.35		
SR.36		

Table 69 Functional Requirement Description of Retrieving Saved Projects

8.10.4.5.8 Admin User Management

Req. ID	Requirement Description	
	Sys. Req. ID	System Requirement
	SR.37	The system should offer an admin interface for user management, allowing users to add, update, and delete users with necessary details.
	SR.38	The system should guarantee secure authentication and authorization mechanisms for admin access, restricting user management actions to authorized personnel.
	SR.39	System should offer user management interface which is user-friendly and intuitive, featuring clear and simple forms for user addition, update, and deletion.
	SR.40	The system shall ensure the database's Users table securely stores user data, including usernames and passwords, and handle deleted records appropriately to maintain data integrity.
	SR.41	The system should record all user management actions by the admin, including the action type, timestamp, and the user performing the action.

Table 70 Functional Requirement Description of Admin User Management

8.10.4.5.9 Modify Predefined Fields

Req. ID	Requirement Description	
REQ-9	Admin can update/manage/ edit predefined fields.	System Requirement
	Sys. Req. ID	
SR.42		The system should offer a user-friendly interface for admins to manage predefined fields related to project categories and keywords, allowing for easy addition, removal, or update.
SR.43		System should ensure that the field management functionality should only be accessible to authorized admins, ensuring secure admin authentication and compliance with the system's security requirements.
SR.44		The admin should be able to add new fields to the system, each with a unique associated keywords.
SR.45		The admin should have the ability to remove existing fields from the system, ensuring that the removal does not impact any existing projects associated with that field.
SR.46		The admin should be able to update existing field names and keywords, ensuring consistent system reflection of these changes.
SR.47		The system should conduct validation checks to ensure that the added or updated fields comply with the specified constraints and guidelines.
SR.48		The system should ensure that changes to predefined fields are reflected seamlessly in the associated projects, and handling removed fields appropriately to maintain data integrity.

Table 71 Functional Requirement Description of Modifying Predefined Fields

8.10.4.5.10 Monitor Trends

Req. ID	Requirement Description	
	Sys. Req. ID	System Requirement
REQ-10		Admin/Users can monitor trends related to specific fields and organizations.
	SR.49	The system shall provide an Analytics Dashboard accessible to the Admin and Users.
	SR.50	The Analytics Dashboard shall display trends related to specific fields and organizations.
	SR.51	Admin/Users shall be able to view the frequency of projects in each field over time.
	SR.52	Admin/Users shall be able to see the popularity of organizations based on the number of projects associated with them.
	SR.53	The Analytics Dashboard shall support graphical representations, such as charts and graphs, for better visualization of trends.
	SR.54	The system shall generate trend reports that can be exported in various formats for further analysis.
	SR.55	The Analytics Dashboard should be user-friendly, easy to navigate for admins.
	SR.56	The Analytics Dashboard must provide real-time or near-real-time trends updates and be capable of handling a large volume of data.

Table 72 Functional Requirement Description of Monitoring Trends

8.10.4.5.11 Track Saved Projects and Export Details

Req. ID	Requirement Description	
REQ-11	Sys. Req. ID	System Requirement
	SR.57	System should provide the admin with ability to view a list of saved projects by employees, including project details.
SR.58		System should allow admin to export project details in various formats like CSV and Excel for further analysis.

Table 73 Functional Requirement Description of Tracking Saved Projects and Exporting Details

8.10.4.6 External Interfaces Requirements

8.10.4.6.1 User Interface (UI)

The User Interface (UI) of the web application will have a user-friendly interface making it easier for the users to navigate the web pages, helping them to efficiently identify relevant projects. According to functionality, the color combinations will be declared for the elements in the pages.

8.10.4.6.2 Hardware

The application is developed for the client, i.e., Vertex Special Technologies. So, the client must have their own desktop/laptop computers running Windows or Ubuntu as an Operating System with a stable internet connection to run this application. Besides that, no other hardware components are required.

8.10.4.6.3 Software

To develop the web application, Django has been used as a Web Framework. Similarly, Python libraries like Selenium and BeautifulSoup are used for scraping websites of donor organizations. All the coding is done using Virtual Studio (VS) Code as an IDE. Similarly, MongoDB database is used to store the scraped data and SQLite is used to store other data like user credentials required for the application.

8.10.4.7 Other Non-Functional Requirements

8.10.4.7.1 Performance Requirements

Req.ID	Requirement Description	Priority
PR.01	The web scraping process should extract project information from donor organization websites with minimal latency, ensuring real-time or near-real-time data availability.	Should
PR.02	The platform's search engine should offer swift and user-friendly project discovery, with a maximum acceptable response time defined to enhance user satisfaction.	Should
PR.03	Optimize project details and user-specific information retrieval from the database for a seamless user experience by setting acceptable time limits for queries and retrievals.	Should
PR.04	The web interface should be responsive, ensuring a seamless user experience with fast page and search results loading times, meeting predefined standards.	Should
PR.05	The analytics dashboard should efficiently process and visualize data, enabling administrators to derive insights without delays, and set acceptable time frames for generating and displaying reports.	Should

Table 74 Non-functional requirement description of Performance Requirements

8.10.4.7.2 Safety Requirements

Req.ID	Requirement Description	Priority
SaR.01	Regularly backup the database to prevent data loss in system failures or unexpected events to restore the system to a consistent state.	Should
SaR.02	Implement industry-standard encryption algorithms and enforce password policies to ensure strong and secure user credentials.	Should
SaR.03	Define and enforce access control policies to restrict unauthorized access to sensitive functionalities.	Could

Table 75 Description of Safety Requirements

8.10.4.7.3 Other Software Quality Attributes

Req.ID	Requirement Description	Priority
SQA.01	The system architecture should be designed to accommodate an increasing number of users and projects without compromising performance.	Should
SQA.02	The platform should be designed for ease of use and minimal training, and usability testing should be conducted to ensure it caters to users of varying technical expertise.	Should
SQA.03	The system should be highly reliable, minimizing downtime and ensuring consistent platform access, while also implementing error-handling mechanisms to gracefully handle unexpected errors and exceptions.	Should
SQA.04	The system's technical documentation, including installation guides, user manuals, and system architecture, should be comprehensive and up to date to aid users and administrators in effectively using the platform.	Should

Table 76 Description of Other Software Quality Attributes

[Back to SRS Document](#)

8.11 Appendix K: Legal, Social and Ethical Issues (Detailed)

8.11.1 Legal Issues

8.11.1.1 Global Context

8.11.1.1.1 Data Privacy and Protection

Issue: Many countries have stringent data protection laws (like the GDPR in the European Union) that govern the collection, use, and storage of personal data. Scraping personal data without consent can lead to violations of these laws (Daruwalla, 2018).

Mitigation: "Scrapequest" ensures compliance by focusing solely on publicly available data and avoiding personal data. It also incorporates security features to anonymize and secure any personal data, it beholds to further safeguard against legal issues.

8.11.1.1.2 Copyright and Intellectual Property

Issue: Web scraping can potentially infringe on copyright if the data scraped is copyrighted and used commercially without permission.

Mitigation: "Scrapequest" checks to determine the copyright status of data before scraping. It also uses data for analysis rather than reproduction, and in the future, it could implement citing sources to help further against copyright infringement issues.

8.11.1.1.3 Robots.txt

Issues:

- If a scraper accesses parts of a website against the explicit instructions of the robots.txt file or the site's ToS, it could be considered a breach of contract. In jurisdictions like the United States, courts have sometimes viewed this as unauthorized access under laws like the Computer Fraud and Abuse Act (CFAA) (IGleads, 2024).
- In some legal frameworks, excessively scraping a website to the point where it impacts the website's performance could be viewed as a "trespass to chattels," which refers to interfering with the owner's use of their personal property (in this case, their server resources) (Riley, 2019).

Mitigation:

- Scrapequest only accesses areas of the website that are allowed, respecting the site administrator's guidelines, and avoiding sections marked as disallowed.
- Scrapequest does not negatively impact the performance of the websites it scrapes. It implements rate-limiting and human-like interactions to reduce the risk of legal issues related to server load and enhance compliance with various legal standards.
- Users of Scrapequest are informed of the legal implications of scraping data and are instructed to use the tool only in ways that comply with applicable laws and website policies. This includes training or guidelines on how to set up and use scraping tools responsibly.

8.11.1.2 Context of Nepal

Issue: Nepal is in the process of strengthening its data protection laws. The current legal framework under the Electronic Transactions Act is somewhat broad and does not specifically address many modern data privacy concerns directly related to web scraping. In the future, Nepal's regulations may evolve to address digital data practices more specifically, impacting how web scraping should be conducted.

Mitigation: Staying informed about changes in local laws and adapting the scraping practices accordingly is crucial. Collaboration with legal experts to routinely assess compliance can prevent potential legal issues.

Furthermore, the software used in the project includes only student versions provided by London Metropolitan University, trial versions, or software that comes with the Windows operating system. No unauthorized or illegally modified software was used. The system complies with all relevant laws, ensuring it meets both global standards and those specific to Nepal. In the development, testing, and implementation of the system, all 80 sections of the Electronic Transactions Act were carefully considered (TEPC, 2006). The system does not contain any harmful software, and steps have been taken to protect user privacy and system security. Additionally, no illegal websites or resources were accessed during the research and documentation phases of the project.

[Back to Report](#)

8.11.2 Social Issues

8.11.2.1 Privacy Concerns

Web scraping can sometimes inadvertently collect personal data without consent, leading to privacy violations. Scrapequest only targets publicly available, non-personal data. It implements data filters and security measures to further protect individual privacy.

8.11.2.2 Data Accuracy and Misrepresentation

Incorrectly scraped data can lead to misinformation or misinterpretation. Such inaccuracies in data become problematic, when such data might be used to make significant economic or policy decisions. Scrapequest incorporates validation algorithms to verify the accuracy of scraped data and ensures reliability of the data. Furthermore, it allows regular updates and checks to ensure data remains current and accurate.

8.11.2.3 Impact on Website Operations

Intensive scraping activities can lead to high server loads, potentially degrading service for other users and incurring extra costs for the website owners. Scrapequest is programmed to imitate human-like interactions while scraping and respect rate limiting guidelines to prevent server overload. Also, users are instructed to run the scraping bots less frequently. This approach respects the operational capacity of target websites.

Besides that, the project is free and open-source (FOSS) and has been made to impact society positively by automating the project discovery processes replacing the arduous manual tasks implemented traditionally. All the possible social issues that could be caused by the system have been eliminated. The system or report does not contain any religious or political contents, it does not hamper self-esteem or self-respect of any individual of the society. The report or the system does violate any social aspect that could result in the social breach.

[Back to Report](#)

8.11.3 Ethical Issues

8.11.3.1 Layoffs

The automation of data collection can lead to job displacement, especially in roles that traditionally involved manual data gathering and analysis. Scrapequest addresses these concerns by making the tool complement human workers rather than replacing them. This involves features that augment human decision-making and utilizing the tool to open up new opportunities for value creation and innovative business practices.

8.11.3.2 Impact on Website Operations

Intensive scraping activities can hamper the operation of targeted websites, reducing their responsiveness or leading to additional costs due to increased bandwidth and server maintenance. Scrapequest implements ethical scraping practices that include rate limiting, respecting robots.txt directives, and avoiding frequent scraping to minimize the operational impact on the websites being scraped.

8.11.3.3 Dependency and Over-reliance

There is a risk that organizations might become overly dependent on automated tools like Scrapequest for decision-making, potentially overlooking the nuanced understanding that human analysts provide. Scrapequest is designed to function as a decision-support tool rather than a decision-maker. It integrates features requiring human verification or interpretation of the data, which can enhance human decision-making processes rather than replace them. Training and guidelines are also provided to encourage users to use Scrapequest as part of a broader analytical process that includes human judgment.

8.11.3.4 Market Distortion

The use of Scrapequest might lead to market distortion if only a few entities have access to this powerful tool. This can potentially create an uneven playing field where larger corporations have a distinct advantage over smaller competitors. Ensuring broader access to Scrapequest and similar technologies can help mitigate this risk. By democratizing access to advanced data tools, small and medium enterprises can also leverage these technologies to compete more effectively. Additionally, Scrapequest could be enabled to offer scaled solutions tailored to different sizes of organizations to ensure that its benefits are accessible to a wider array of businesses.

Besides that, the codes in the system are free from malware, and no harmful codes have been added. The system was developed in a fair environment for legitimate purposes. It ensures that all data rights belong to the client, with access by others only possible through client permission. The system does not incorporate any code from external sources without appropriate attribution, and it represents an original concept by its creator. The system's report includes accurate references and citations, eliminating any ambiguity. The content of the report has been thoroughly reviewed and revised to ensure it is free from plagiarism. Both the report and system adhere to the rules and regulations set by London Metropolitan University and under the guidance of supervisors. The system respects client data and does not infringe on the intellectual property rights of the client. Furthermore, the system and report comply with all ethical standards, thus presenting no ethical issues.

[Back to Report](#)

8.12 Appendix L: Advantages of the System (Detailed)

Economic Growth

- Scrapequest automates project discovery, especially involving projects from international development sector. It reduces the chances of missed opportunities which in turn, aids in landing more international projects which in turn can benefit the nation's economy by welcoming revenue to the country via foreign investments.

Enhanced Efficiency

- Automation of Tedium Tasks: Scrapequest automates the labor-intensive and repetitive task of searching for project information across multiple websites and databases, which traditionally consumes considerable time and resources.
- Rapid Data Collection: The tool can gather vast amounts of data in a fraction of the time it would take human researchers, allowing organizations to respond more quickly to emerging opportunities or changes in the market.

Improved Accuracy and Relevance

- Accurate Data Extraction: Scrapequest is programmed to identify and extract relevant data with high precision, reducing human error that often comes with manual data entry and extraction.
- Real-Time Data Updates: By leveraging web scraping, the tool provides the most current information available, which is crucial for making timely decisions based on the latest data.

Cost Reduction

- Decreased Labor Costs: Automating the process of project discovery reduces the need for extensive manpower dedicated to these tasks, thereby lowering labor costs.
- Minimized Opportunity Costs: Faster data collection and processing means that opportunities can be identified and acted upon more quickly, reducing the costs associated with delays and missed opportunities.

Competitive Advantage

- Strategic Project Alignment: Scrapequest helps organizations identify projects that closely align with their strategic goals and capabilities, facilitating better project selection and portfolio management.
- Market Analysis: The tool can analyse market trends and competitor activities, providing insights that can inform strategic decisions and competitive positioning.

Scalability and Flexibility

- Scalable Solutions: Scrapequest can be scaled to handle increasing amounts of data as the organization grows, without a corresponding increase in resource allocation.
- Customizable Features: The system can be tailored to meet specific organizational needs, whether it's adjusting the scope of search parameters or integrating with existing IT infrastructures.

Data-Driven Decision Making

- Enhanced Data Visualization and Reporting: Scrapequest can integrate with data visualization tools to present complex data in an easily understandable format, aiding in better decision-making.
- Comprehensive Data Analysis: The system can perform complex analyses on the scraped data, uncovering patterns and insights that might not be apparent through manual analysis.

Ethical and Responsible Data Use

- Compliance with Legal Standards: Scrapequest is designed to comply with international and local data privacy laws, ensuring that data collection practices are both legal and ethical.
- Respect for Website Integrity: By adhering to robots.txt files and incorporating rate limiting, the system minimizes its impact on the websites from which it scrapes data, maintaining a responsible footprint.

User Experience and Accessibility

- User-Friendly Interface: Scrapequest features a clean and intuitive interface that makes it accessible to users with varying levels of technical expertise.
- Support and Training: The provision of comprehensive user guides, support, and training materials ensures that users can maximize the tool's potential.

[Back to Report](#)

8.13 Appendix M: Limitations (upon scaling)

- Scalability Concerns:**

As the volume of data increases, there are potential scalability issues that might affect Scrapequest's performance. Ensuring that the system can handle larger amounts of data and more simultaneous users without degrading performance is essential for maintaining efficiency.

- Rate Limiting and IP Blocking:**

Scrapequest could face access issues, such as IP blocking, from websites that have strict controls on data access upon scaling. This is a significant limitation as it could prevent Scrapequest from consistently gathering data.

- Legal and Compliance Challenges:**

There may be potential legal challenges related to violating terms of service or data privacy laws, especially in regions with stringent regulations like Europe's GDPR. Navigating these legal landscapes is crucial to avoid legal repercussions and ensure compliance.

- Data Accuracy and Timeliness:**

There is a risk of collecting data that is not current or accurate. Relying on such sources that do not update frequently may lead to decisions based on outdated or incorrect information.

- Resource Intensity:**

High resource consumption is needed for effective scraping, particularly at a larger scale. This can lead to increased operational costs due to the need for substantial computational power and bandwidth.

- Complexity in Data Normalization:**

Inconsistencies in data formats from different sources pose a challenge. Normalizing this data into a uniform format for analysis and storage can be complex and resource intensive.

- **Language and Localization Challenges:**

Handling data from multilingual sources can be challenging. Additional processing is required to manage various languages and regional settings, which can complicate the scraping process.

Addressing these limitations will require a combination of technological advancements, strategic planning, and continuous improvement to ensure Scrapequest remains a valuable tool in the rapidly evolving digital landscape.

[Back to Report](#)

8.14 Appendix N: User Guide

Step 1: Navigate to the directory and load the docker image of scrapequest onto your system.

Command: “docker load < scrapequest.tar”

```
sharumss@sharumss-virtual-machine:~/Desktop/docker$ sudo docker load < scrapequest.tar
1f00ff201478: Loading layer [=====] 77.83MB/77.83MB
bfc9081d1eb2: Loading layer [=====] 9.552MB/9.552MB
d9ac86592bf9: Loading layer [=====] 32.92MB/32.92MB
55b572ec4bd1: Loading layer [=====] 5.12kB/5.12kB
438781090b57: Loading layer [=====] 12.88MB/12.88MB
1bc01cff5374: Loading layer [=====] 1.536kB/1.536kB
fdf24c222fad: Loading layer [=====] 2.56kB/2.56kB
6040bdf6a7a4: Loading layer [=====] 240.5MB/240.5MB
33eee93cf9d3: Loading layer [=====] 6.341MB/6.341MB
Loaded image: scrapequest:latest
sharumss@sharumss-virtual-machine:~/Desktop/docker$ sudo docker image ls
REPOSITORY          TAG      IMAGE ID   CREATED    SIZE
scrapequest         latest   06023190be01  2 days ago  363MB
```

Figure 288 Loading Docker Image

Step 2: Run the loaded docker image.

Command: “docker run -p 8080:8080 scrapequest: latest”

```
sharumss@sharumss-virtual-machine:~/Desktop/Scrapequest$ sudo docker run scrapequest:latest
Performing system checks...

System check identified some issues:

WARNINGS:
?: (urls.W005) URL namespace 'admin' isn't unique. You may not be able to reverse all URLs in this namespace

System check identified 1 issue (0 silenced).
April 20, 2024 - 11:23:48
Django version 5.0.4, using settings 'Scrapequest.settings'
Starting development server at http://0.0.0.0:8000/
Quit the server with CONTROL-C.
```

Figure 289 Running Docker Image.

Step 3: Open the server, you will be taken to login page. If you haven't already, signup by clicking on the prompt, and register yourself. Be sure to follow instructions.

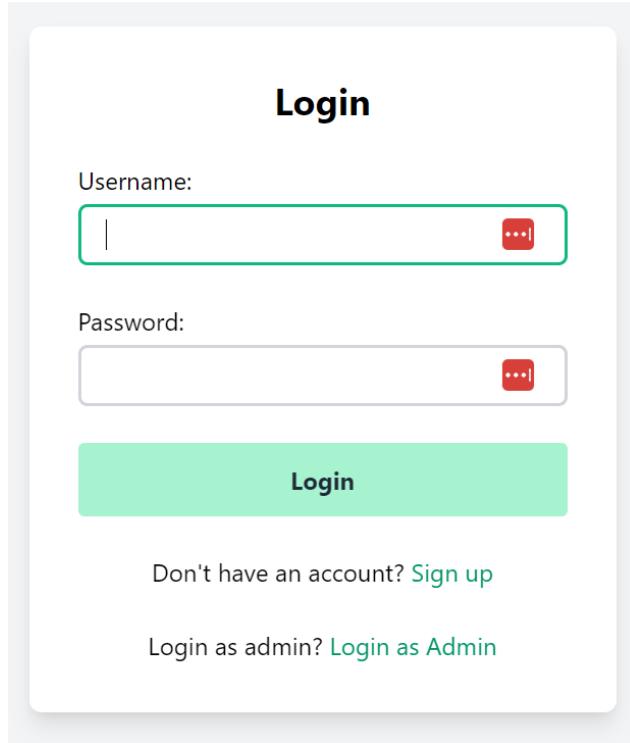


Figure 290 Landing Page

Sign Up

Username:

Required. 150 characters or fewer. Letters, digits and @./+/-/_ only.

Email:

Required. Inform a valid email address.

Password:

Your password can't be too similar to your other personal information.
Your password must contain at least 8 characters.
Your password can't be a commonly used password.
Your password can't be entirely numeric.

Password confirmation:

Enter the same password as before, for verification.

Sign up

Already have an account? [Login](#)

Figure 291 Registration Page

Step 4: Ask admin to provide you “Active Status”. If you’re an admin, go to login page for admin via link in login page, and provide “active status” manually, to register user.

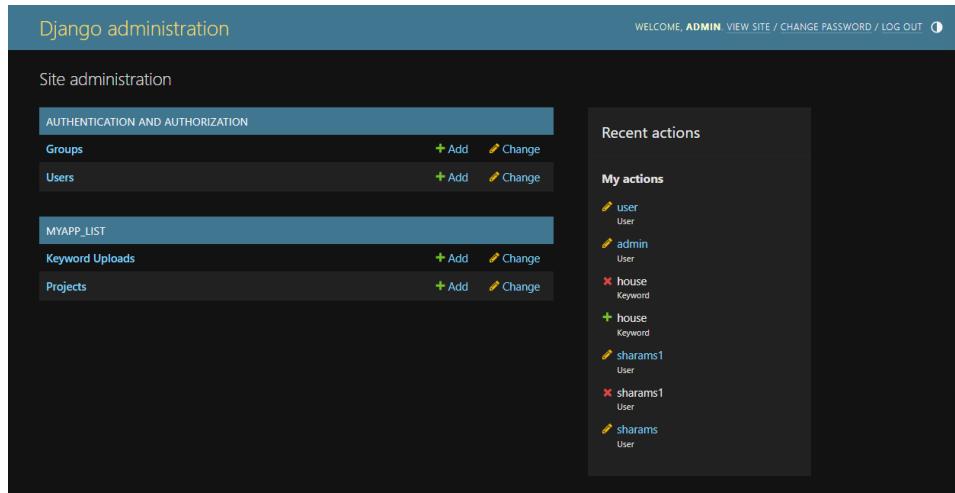


Figure 292 Admin Panel

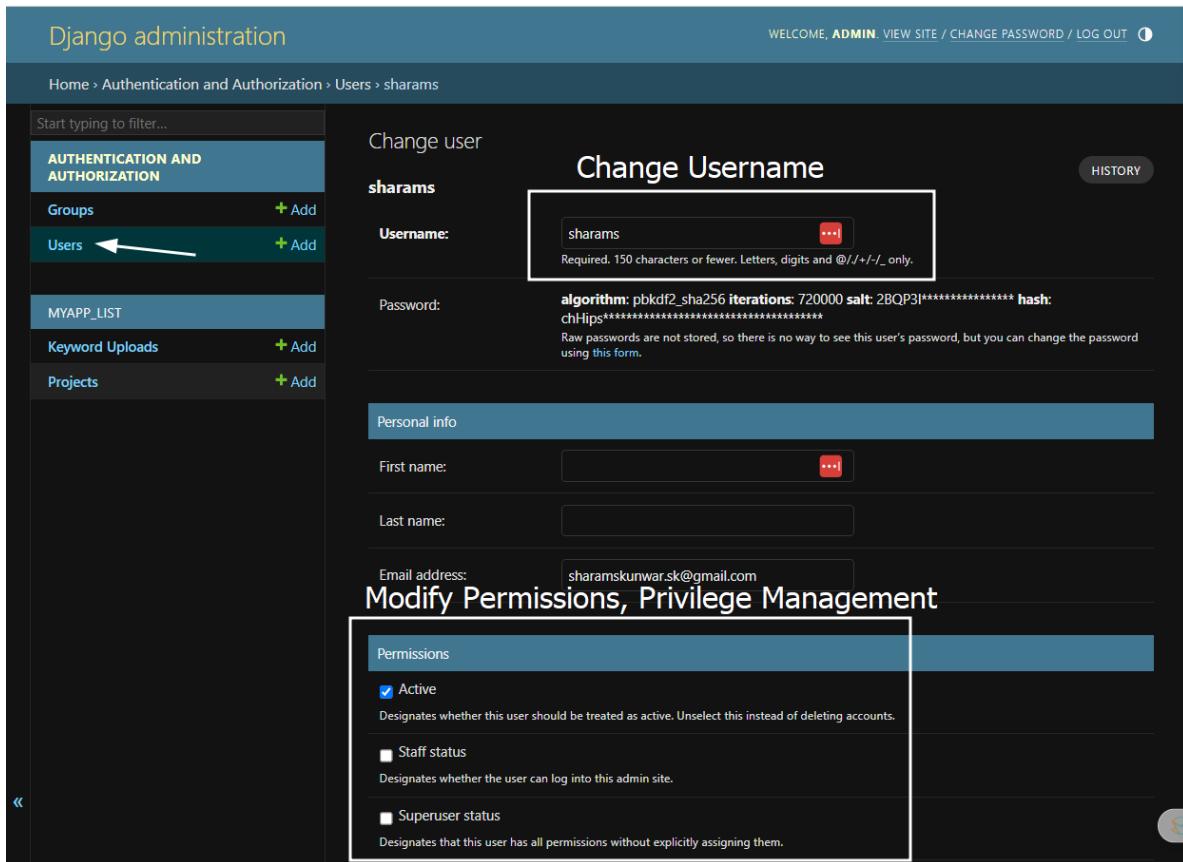


Figure 293 Guide to manage User.

Step 5: Once registered, login using valid credentials, you will be taken to homepage, where you can view dashboard with visualizations which will provide you extra insights and you will be able to search projects of your liking as well. You can also navigate to other pages, which shows projects filtered via different algorithms. Figures below, guides you through each page.

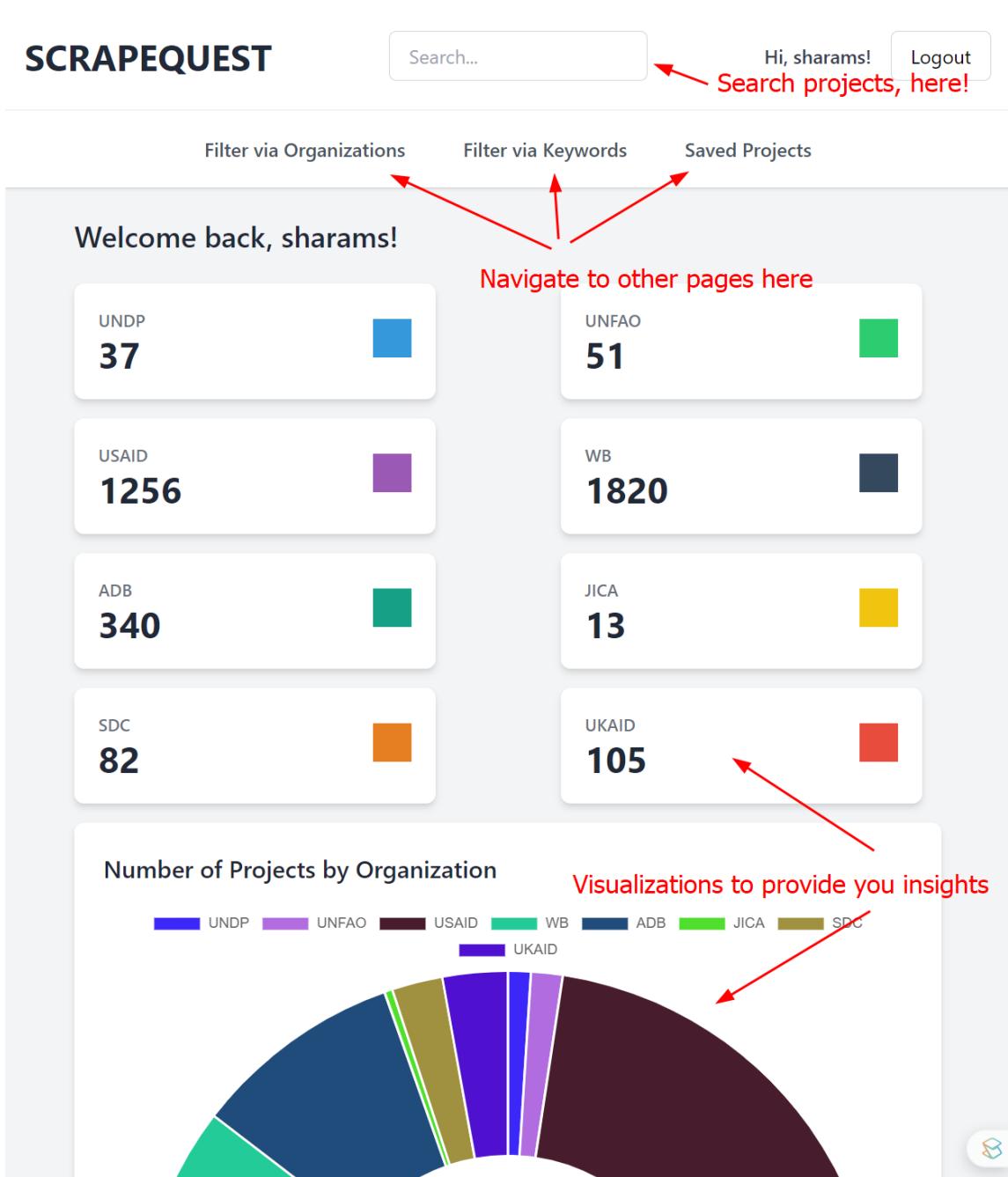


Figure 294 Homepage Guide

SCRAPEQUEST

Search... Hi, sharams! Logout

Filter via Organizations Filter via Keywords Saved Projects

UNDP UNFAO USAID JICA ADB WB UKAID SDC

Switch organizations, using these buttons

WB Projects

Dedicated Grant Mechanism for Indigenous Peoples and Local Communities in Nepal - P171720

[View Project](#)

Desktop computer-4(2 provincial and 2 PMU office) and Printer

[Procurement Details](#)

Published: April 17, 2024

Save Project

Nepal Livestock Sector Innovation Project - P156797

[View Project](#)

Construction of; Slice 1: Store Room, Record Room and Compound Wall at Bull Mother shed and Slice 2; Construction of Quarantine Shed at NLBO, Pokhara

[Procurement Details](#)

Published: April 10, 2024

Save Project

If the project interests you, you may save it using this button for future reference.

Nepal Livestock Sector Innovation Project - P156797

[View Project](#)

Construction of; Slice 1: Store Room, Record Room and Compound Wall at Bull Mother shed and Slice 2; Construction of Quarantine Shed at NLBO, Pokhara

[Procurement Details](#)

Published: April 10, 2024

Nepal Livestock Sector Innovation Project - P156797

[View Project](#)

Procurement of UV-VIS Spectrophotometer for National Animal Feed and Livestock Quality Management Laboratory (NAFLQML) PMU

[Procurement Details](#)

Published: April 10, 2024

Figure 295 Filter Via Organizations Page Guide

SCRAPEQUEST

These buttons are predefined keywords set by your admin, which show you projects related to each keyword upon clicking.

Search... Hi, sharams! Logout

Filter via Organizations Filter via Keywords Saved Projects

Food Food Aid Food system Food security Agriculture Digital Food Science

Food Loss Education Migration Nutrition Health Climate Change Livelihood

Disaster Risk Reduction Web Application Mobile Application Home House

Showing Results For Food system

Name of the Project: Green and Resilient Rural Recovery through Agri-Food System Transformation in the Asia and Pacific Region - Knowledge Management and Communications Expert

Link to the project: [View](#)

Project Status: Status: Closed

Project Description: 55113-001; Regional, Bangladesh, Cambodia, India, Kyrgyz Republic, Lao People's Democratic Republic, Maldives, Mongolia, Nauru, Nepal, Pakistan, China, People's Republic of, Sri Lanka, Vanuatu, Viet Nam; Agriculture, natural resources and rural development; Posting date: 10 Aug 2022

Organization: ADB

You can save these as well.

Name of the Project: Green and Resilient Rural Recovery through Agri-Food System Transformation in the Asia and Pacific Region - Multimedia Expert

Link to the project: [View](#)

Project Status: Status: Closed

Project Description: 55113-001; Regional, Bangladesh, Cambodia, India, Kyrgyz Republic, Lao People's Democratic Republic, Maldives, Mongolia, Nauru, Nepal, Pakistan, China, People's Republic of, Sri Lanka, Vanuatu, Viet Nam; Agriculture, natural resources and rural development; Posting date: 10 Aug 2022

Organization: ADB

Save Project

Figure 296 Filter Via Keywords Page Guide

The screenshot shows the SCRAPEQUEST website interface. At the top, there is a navigation bar with the title "SCRAPEQUEST", a search bar containing "Search...", and user account links "Hi, sharams!" and "Logout". Below the navigation bar are three filter options: "Filter via Organizations", "Filter via Keywords", and "Saved Projects". A red annotation highlights the "Saved Projects" link with the text "these are the projects you saved for future references earlier. admin is able to see your saved projects." An arrow points from this text to the "Saved Projects" link in the navigation bar. The main content area displays two saved project cards. The first card is for "Recovery & Resilience" and the second is for "MCC Transmission Lines Activity". Both cards show project details like names, durations, and links to view more information. A large red rectangular box surrounds both project cards.

these are the projects you saved for future references earlier. admin is able to see your saved projects. \ Saved Projects

Recovery & Resilience
View Project

MCC Transmission Lines Activity
Duration:
Sector: Energy

Name of the Project: UNJP/NEP/084/UNJ
Assessing the impact of the global crisis on the agriculture and food security situation in Nepal
Duration of the Project: 20-Jun-2022 - 31-Mar-2023
Link to the project: View
Organization: UNFAO

MiRiDew - Migrant Rights and Decent Work
View Project
Project Details: Remittances sent home by over 4 million migrant workers have significantly contributed to the economic development of Nepal. The proposed project will enhance the capacities of the Government of Nepal and strengthen mechanisms to better protect the rights of workers abroad. There will be a special focus on women's need and the adverse effects of climate change exacerbating the vulnerability of migrants. The project capitalises on Switzerland's longstanding engagement on labour migration in Nepal.
Duration: 01.07.2023 - 31.12.2026

Figure 297 Saved Projects Page Guide

Step 6: After you're done, you may terminate your session using 'Logout' button in the navbar.

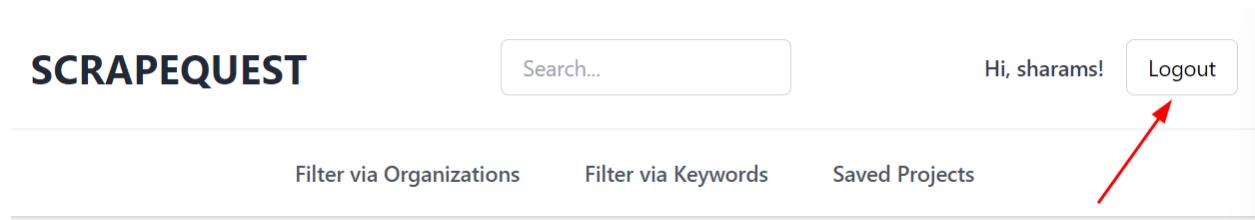


Figure 298 Logging out of the system.

Step 7: If you're an admin, there are set of exclusive features for you. Below is the comprehensive guide.

The screenshot shows a file explorer interface with several Python scripts listed. A red box highlights three specific files: 'cleaned_data.py', 'MongoAtlasconnection.py', and 'mongoscript.py'. Another red box highlights a list of required modules: bs4, reportlab, pymongo, django, selenium, whitenoise, pandas, numpy, selenium, fake_useragent, and requests. A third red box highlights a list of project data files: ADB.py, JICA.py, SDC.py, UKaid.py, UNDP.py, UNFAO.py, USAID.py, and WB.py. Red arrows point from the annotations to their respective targets.

```

1  bs4
2  reportlab
3  pymongo
4  django
5  selenium
6  whitenoise
7  pandas
8  numpy
9  selenium
10  fake_useragent
11  requests
12

```

these are required modules.
clean the scraped data using this script.
send the scraped data to the MongoAtlas executing this script.
you can run scraping my executing these scripts.

Figure 299 Scraping Guide

[Note: **Don't execute scraping scripts frequently.** It may cause compliance issues.]

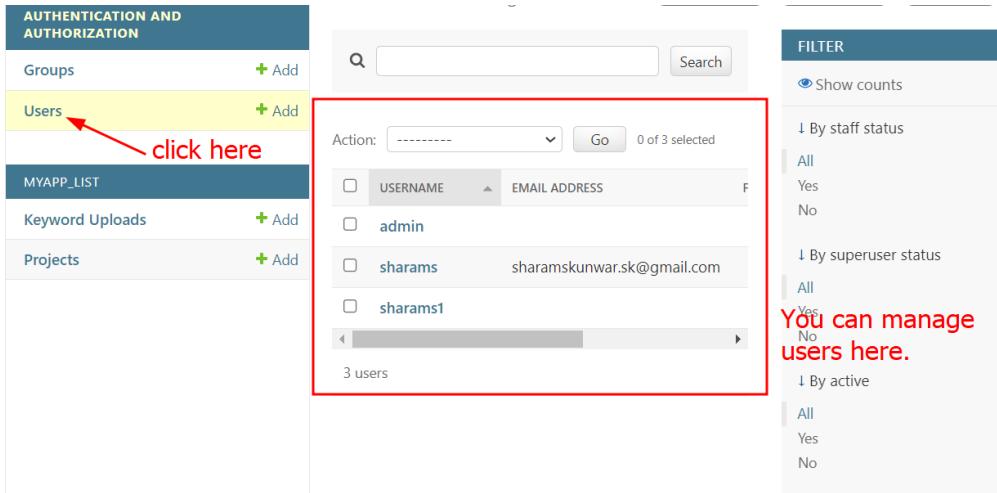


Figure 300 Manage Users Guide

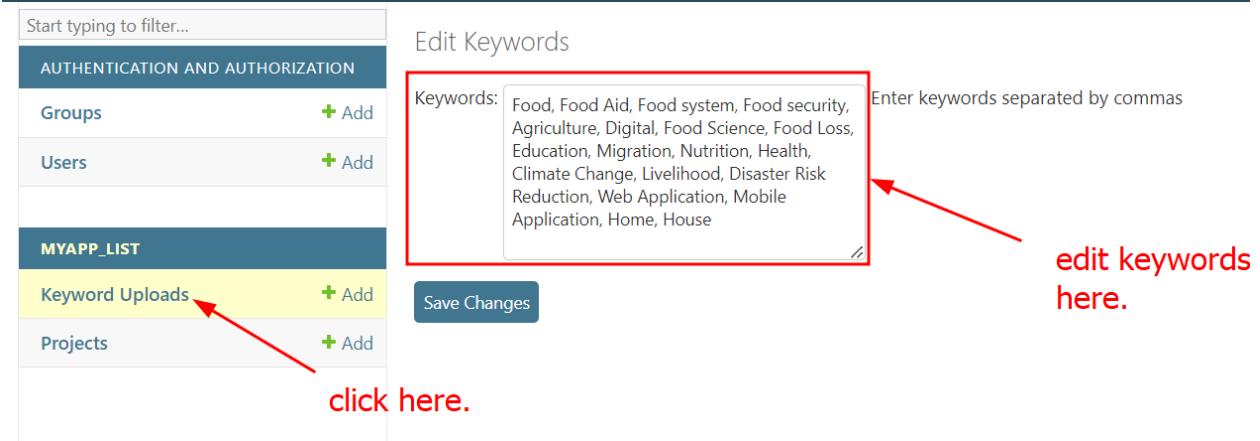


Figure 301 Modify Keywords Guide

Start typing to filter...

AUTHENTICATION AND AUTHORIZATION

Groups + Add
Users + Add

MYAPP_LIST

Keyword Uploads + Add
Projects + Add

Select project to change

Action: ----- Go 0 of 44 selected

PROJECT DETAILS	USER
Promoting Green Recovery Project View Project	sharams
OSRO/NEP/401/USA - Immediate technical assistance to strengthen emergency preparedness for highly pathogenic avian influenza Duration: 2014 - 2015 (project completed) View Project	sharams
Recovery & Resilience View Project	sharams
Recovery & Resilience View Project	sharams
Japan's Official Development Assistance (ODA): Rolling Plan for Nepal (PDF) View Project	sharams
Name of the Project: Food and Nutrition Security Enhancement Project - P164319 Link to the project: View Procurement Details: Procurement of improved maize seed. Link to the Procurement: View Date of Publication: March 4, 2024 Organization: WB	sharams
Enhancing human security through local climate actions View Project	sharams
Recovery & Resilience View Project	sharams

EXPORT TO CSV **EXPORT TO PDF** **ADD PROJECT +**

click here

these are saved projects.

this shows user who saved it.

use buttons here to export details.

Figure 302 Guide to View Saved Projects and Export Details

[Back to Report](#)

8.15 Appendix O: Context Survey

8.15.1 Context Survey Form

Context Survey Form - Scrapequest

B I U ↲ ↳

This survey form aims to capture insights into the current state of project discovery among Nepali Companies, the issues they encounter, and the potential for an automated solution like Scrapequest to bring significant improvements.

Tailored Solution for Your Needs



SCRAPEQUEST

Select the name of your company: *

1. Vertex Special Technologies
2. IME Automotives
3. ISUZU
4. Jagadamba Cement
5. JCB
6. Bhageshwor Builders
7. Bhagawati Nirman Sewa
8. Bharat Wire Ropes Limited
9. International Science Trading Concern
10. Global IME Bank Limited
11. Kewanch Nepal
12. Cypher Technology
13. D and P Construction
14. Syarona Construction
15. Infrastructure Engineering and Research

16. Annapurna Global Impex
17. Unique International Traders
18. Shaurya Enterprises
19. Bhatta and Associates
20. NBSM And Associates Chartered Accountants
21. Eagle Eye It Service
22. Debugsoft Pvt. Ltd.
23. Neology Next
24. Info Developers
25. Max International
26. Paragon IT Solution
27. Leads Innovation
28. Saipal Technologies
29. Dryice Solutions
30. Om Networks
31. Nextgen Solutions
32. Cryptogen Nepal
33. Mero Network
34. The Wave Group Trading
35. Eminence Ways
36. Astral Computers Nepal

What are the sectors of your interest? *

- Human Resources
- Vehicles & Mechanical Equipment
- Miscellaneous
- IT, Printing, Advertising & Stationery
- Medical
- Construction Materials
- Power Generation and Electrical Items
- Consulting Services/ Legal & Audit Services
- Auction & Sale

How do you currently identify potential projects for your organization?

- Through manual searches on various websites and databases.
- Using a subscription to a commercial project discovery database.
- Networking with industry contacts and attending conferences.
- We don't have a structured method for project discovery.
- Other...

How much time does your team typically spend each week on project discovery? *

- Less than 5 hours.
- 5 to 10 hours.
- 10 to 20 hours.
- More than 20 hours.
- Other...

What percentage of discovered projects align closely with your company's strategic goals and expertise? *

- Less than 25%
- 25% to 50%
- 50% to 75%
- More than 75%
- Other...

What are the biggest challenges you face during the project discovery phase? (Select all that apply) *

- Time-consuming manual searches.
 - Overwhelming amount of data to sift through.
 - Difficulty in verifying the accuracy and relevance of found data.
 - High costs associated with project discovery tools and databases.
 - Non-response or incomplete information from potential project leads.
 - Other...
-

How often do you encounter missed opportunities due to delays in the project discovery phase? *

- Very frequently.
- Occasionally.
- Rarely.
- Never.

Which feature would you value the most in a project discovery tool? *

- Real-time updates and alerts on new projects.
- Comprehensive data filtering to match projects with our strategic goals.
- Automatic data verification to ensure reliability.
- Cost-effectiveness compared to current methods.
- Ease of integration with existing IT systems.

How do you currently manage the overload of information in project discovery? *

- We have a dedicated team that handles all data.
- We use specialized software to help filter and organize data.
- We often overlook a lot of data due to resource constraints.
- We do not have a specific method; it's quite chaotic.

What impact does the current project discovery process have on team morale?

- Significantly decreases morale due to repetitive and mundane tasks.
- Somewhat decreases morale but is manageable.
- Does not affect morale.
- Actually increases morale.

How critical is cost efficiency in your project discovery process?

- Extremely critical; we need to minimize costs as much as possible.
- Important but not the only factor we consider.
- Somewhat important; we are willing to invest in quality tools.
- Not important; we focus more on the effectiveness of the process.

Figure 303 Context Survey Form

8.15.2 Sample of Filled Context Survey Form

Responses cannot be edited

Context Survey Form - Scrapequest

This survey form aims to capture insights into the current state of project discovery among Nepali Companies, the issues they encounter, and the potential for an automated solution like Scrapequest to bring significant improvements.

* Indicates required question

Tailored Solution for Your Needs



SCRAPEQUEST

Select the name of your company: *

Vertex Special Technologies



What are the sectors of your interest? *

- Human Resources
- Vehicles & Mechanical Equipment
- Miscellaneous
- IT, Printing, Advertising & Stationery
- Medical
- Construction Materials
- Power Generation and Electrical Items
- Consulting Services/ Legal & Audit Services
- Auction & Sale

How do you currently identify potential projects for your organization?

- Through manual searches on various websites and databases.
- Using a subscription to a commercial project discovery database.
- Networking with industry contacts and attending conferences.
- We don't have a structured method for project discovery.
- Other: _____

How much time does your team typically spend each week on project discovery? *

- Less than 5 hours.
- 5 to 10 hours.
- 10 to 20 hours.
- More than 20 hours.
- Other: _____

What percentage of discovered projects align closely with your company's strategic goals and expertise? *

- Less than 25%
- 25% to 50%
- 50% to 75%
- More than 75%
- Other: _____

What are the biggest challenges you face during the project discovery phase? (Select all that apply) *

- Time-consuming manual searches.
- Overwhelming amount of data to sift through.
- Difficulty in verifying the accuracy and relevance of found data.
- High costs associated with project discovery tools and databases.
- Non-response or incomplete information from potential project leads.
- Other: _____

How often do you encounter missed opportunities due to delays in the project discovery phase? *

- Very frequently.
- Occasionally.
- Rarely.
- Never.

Which feature would you value the most in a project discovery tool? *

- Real-time updates and alerts on new projects.
- Comprehensive data filtering to match projects with our strategic goals.
- Automatic data verification to ensure reliability.
- Cost-effectiveness compared to current methods.
- Ease of integration with existing IT systems.

How do you currently manage the overload of information in project discovery? *

- We have a dedicated team that handles all data.
- We use specialized software to help filter and organize data.
- We often overlook a lot of data due to resource constraints.
- We do not have a specific method; it's quite chaotic.

What impact does the current project discovery process have on team morale?

- Significantly decreases morale due to repetitive and mundane tasks.
- Somewhat decreases morale but is manageable.
- Does not affect morale.
- Actually increases morale.

How critical is cost efficiency in your project discovery process?

- Extremely critical; we need to minimize costs as much as possible.
- Important but not the only factor we consider.
- Somewhat important; we are willing to invest in quality tools.
- Not important; we focus more on the effectiveness of the process.

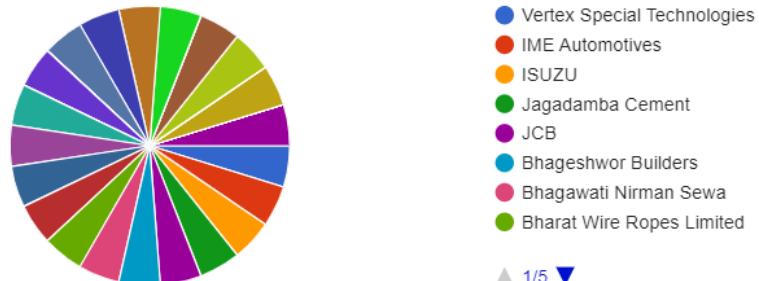
Figure 304 Sample of Filled Context Survey Form

8.15.3 Context Survey Form Results

Select the name of your company:

 Copy

21 responses

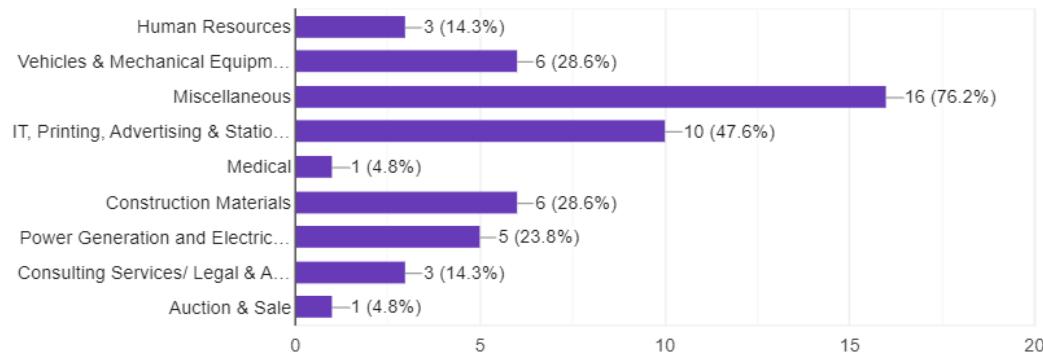


▲ 1/5 ▼

What are the sectors of your interest?

 Copy

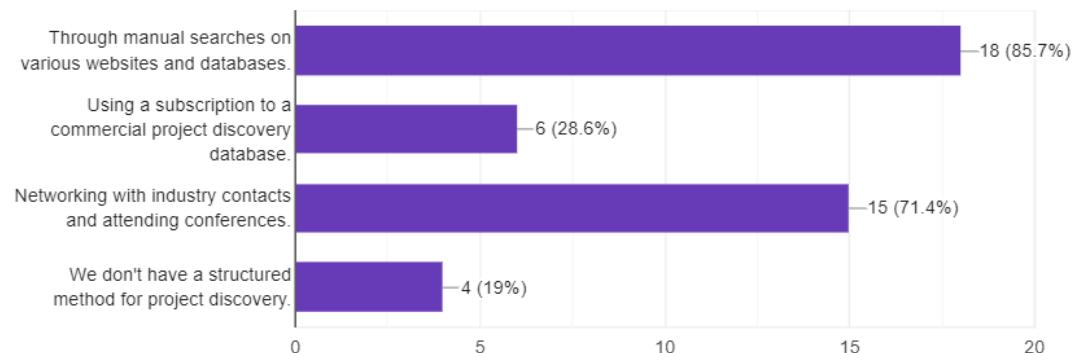
21 responses



How do you currently identify potential projects for your organization?

 Copy

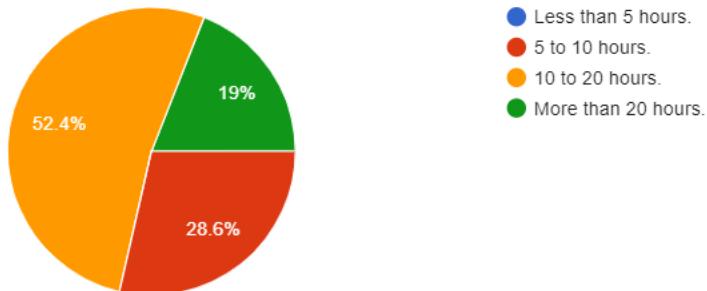
21 responses



Copy

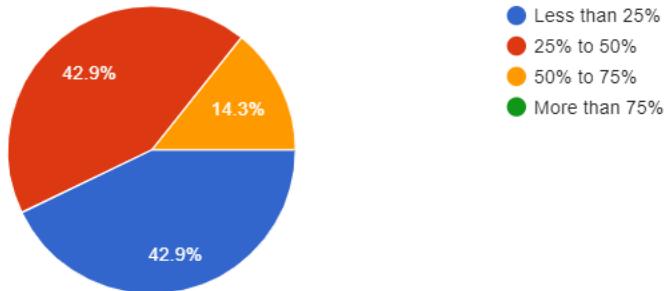
How much time does your team typically spend each week on project discovery?

21 responses


Copy

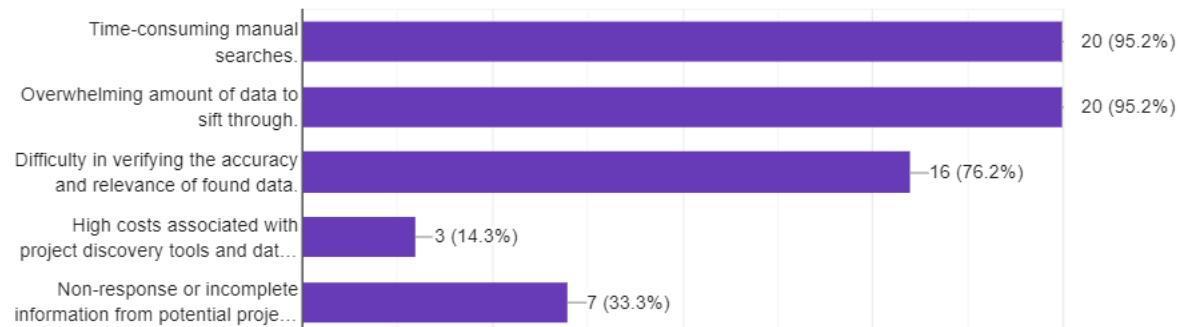
What percentage of discovered projects align closely with your company's strategic goals and expertise?

21 responses


Copy

What are the biggest challenges you face during the project discovery phase? (Select all that apply)

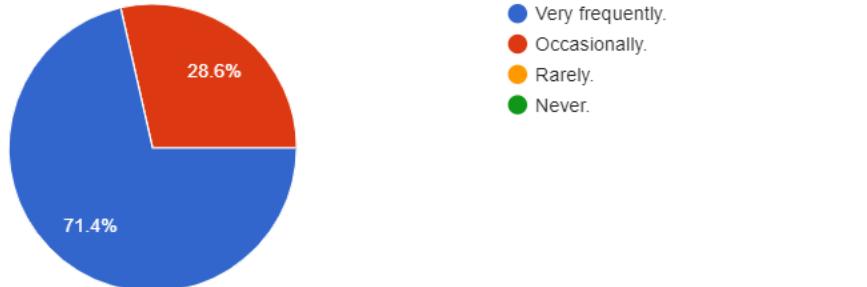
21 responses



How often do you encounter missed opportunities due to delays in the project discovery phase?

 Copy

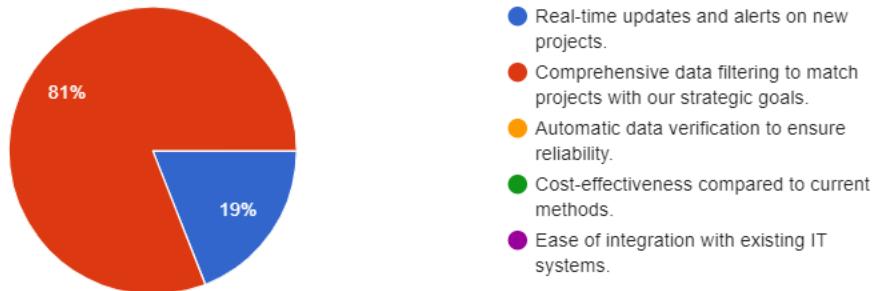
21 responses



Which feature would you value the most in a project discovery tool?

 Copy

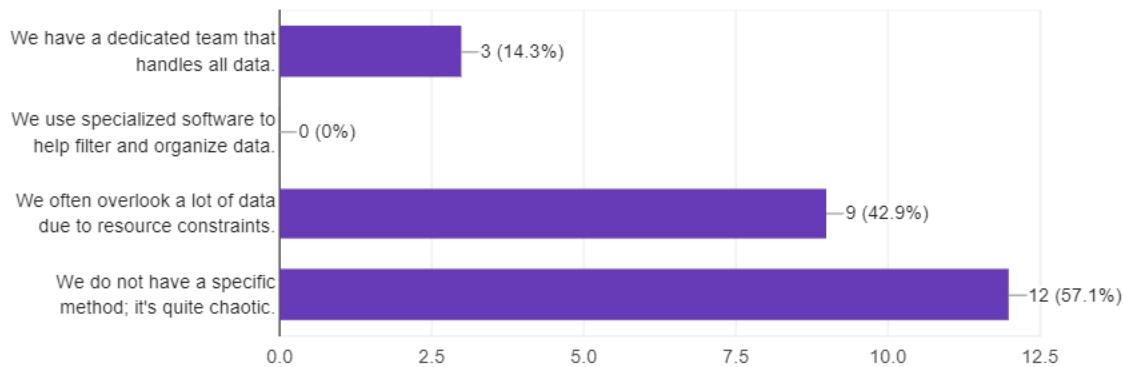
21 responses



How do you currently manage the overload of information in project discovery?

 Copy

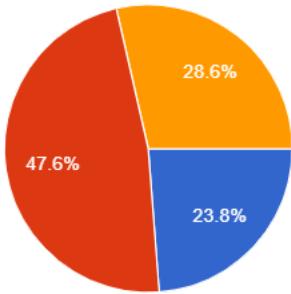
21 responses



What impact does the current project discovery process have on team morale?

 Copy

21 responses

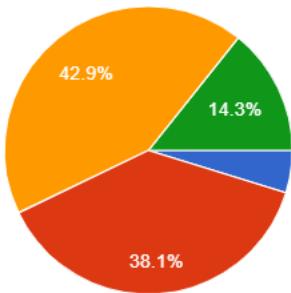


- Significantly decreases morale due to repetitive and mundane tasks.
- Somewhat decreases morale but is manageable.
- Does not affect morale.
- Actually increases morale.

How critical is cost efficiency in your project discovery process?

 Copy

21 responses



- Extremely critical; we need to minimize costs as much as possible.
- Important but not the only factor we consider.
- Somewhat important; we are willing to invest in quality tools.
- Not important; we focus more on the effectiveness of the process.

Figure 305 Results of Context Survey Form

8.15.3.1 Summary of Results

The survey of Nepali companies highlights the current practices and challenges faced during project discovery phases and assesses the potential impact of an automated solution like Scrapequest. A significant majority of respondents (85.7%) rely primarily on manual searches across various platforms to identify potential projects, which suggests a widespread need for more streamlined methods. Networking and industry contacts also play a crucial role for many (71.4%), indicating that relationship-building remains essential.

However, the process is time-consuming for most companies, with over half (52.4%) spending between 5 to 10 hours weekly on project discovery. This significant time investment could potentially be reduced with more efficient tools. The alignment of discovered projects with company strategies is suboptimal, as a substantial portion of companies (42.9%) find only 25% to 50% of projects fitting their goals closely.

The survey identifies several critical challenges: nearly all respondents (95.2%) find the process overly time-consuming and overwhelming due to the large volume of data to sift through. There is also a notable difficulty in verifying the accuracy and relevance of information, coupled with high costs associated with existing discovery tools.

Most companies (71.4%) occasionally miss opportunities due to delays in discovery, underscoring the need for real-time updates and alerts, which 81% of respondents prioritize as a valuable feature in a project discovery tool. Cost efficiency emerges as another significant concern, with over 80% of participants rating it as critical or important, reflecting a general sensitivity to the financial implications of project discovery methods.

In conclusion, the data from the survey strongly indicates a fertile ground for solutions like Scrapequest, which could address the key issues of efficiency, cost, and data management, thereby improving the project discovery landscape for Nepali companies.

[Back to Report](#)

8.16 Appendix P: Progress Table

SN	Tasks	Status	Progress (%)
1.	Idea Finalization	Completed	100%
2.	Client Meeting	Completed	100%
3.	Requirement Analysis	Completed	100%
4.	Resource Evaluation	Completed	100%
5.	Initial Project Plan	Completed	100%
6.	Resource Reevaluation	Completed	100%
7.	Detailed Project Plan	Completed	100%
8.	Risk Identification	Completed	100%
9.	Proposal Documentation	Completed	100%
10.	Web Scraping	Completed	100%
11.	Database Development	Completed	100%
12.	System Architecture Design	Completed	100%
13.	Mock-up Design	Completed	100%
14.	Wireframe Design	Completed	100%
15.	Front-End Construction	Completed	100%
16.	Back-End Construction	Completed	100%
17.	Black-Box Testing	Completed	100%
18.	White-Box Testing	Completed	100%
19.	Delivery to Client	Completed	100%
20.	Feedback Collection and Analysis	Completed	100%
21.	Upgrades and Adjustments	Completed	100%
22.	Final Report Documentation	Completed	100%

Table 77 Progress Table