

Final Report of Online Examination System

Report by: Sayush Khadka

Abstract:

An online examination system is a digital platform designed to make it easier to test and assess students and employees. With the use of this web-based tool, test creators can remotely create, schedule, and manage exams, and test takers can access the system from any location with an internet connection. The system has options including quick grading, timed exams, question banks, and random question generation. The adaptability, usefulness, and affordability of online exam systems have contributed to their rise in popularity over the past few years. They have revolutionized the traditional testing process by enabling remote learning and eliminating the need for physical exams. A system for taking exams online offers many benefits over traditional exam procedures. Because they can take tests at any time and from any location, it allows teachers and students more freedom. This function is especially beneficial during the pandemic, where remote learning has become essential. The strategy is also more efficient since it lessens the administrative strain placed on instructors by making it simpler for them to create and deliver tests. Online testing tools can also guarantee that all test takers are scored fairly by minimizing human error in grading.

For conducting the examination and doing away with the conventional approach, this project has the facilities to match the technological world. The project undergoes a significant amount of study, beginning with the business model and the project's goals and objectives; highlighting the problem scenario and the project that will be used to solve it. The project was carried out using a methodology that was based on research in order to deliver a definitive outcome for the stated target and objectives. This report's foundation is the record of the research that was done to finish the project.

Table of Contents

1. Introduction:	1
1.1. Project Description:.....	2
1.2. Current scenario:	3
1.3. Problem domain:.....	4
1.4. Problem Solution:	6
1.5. Aims and objectives:.....	8
1.5.1. Aim:.....	8
1.5.2. Objectives:	8
1.6. Report structure:	9
1.6.1. Background:	9
1.6.2. Development:	9
1.6.3. Testing and Analysis:	9
1.6.4. Conclusion:	9
2. Background:	10
2.1. About the end users:.....	10
2.2. Understanding the solution:.....	10
2.3. Similar projects:	12
2.4. Comparison of the projects:	13
3. Development:	15
3.1. Considered Methodology:.....	15
3.2. Selected methodology:	16
3.3. Phases of methodology:	17
3.4. Survey results:	18
3.4.1. Pre-Survey results:.....	18
3.4.2. Post-survey results:.....	18
3.5. Requirement analysis:	19
3.5.1. For Admin:.....	19
3.5.2. For Staff:	20
3.5.3. For Teacher:.....	75
3.5.4. For Student:	102
3.6. Design:	123

3.6.1. System Design:	123
3.7. Implementation:.....	127
3.7.1. Register apps:.....	127
3.7.2. Model Design (Database design):.....	128
3.7.3. Views:	134
3.7.4. Template:	136
3.7.5. Forms:.....	139
3.7.6. URL:.....	142
4. Testing and Analysis:	146
4.1. Test Plan:	146
4.1.1. White Box Testing, Unit Testing:.....	146
4.1.2. Black Box Testing, System Testing:.....	146
4.2. Unit Testing:.....	147
4.2.1. Test case: Create Admin (Superuser).....	147
4.2.2. Test Case: To test User Registration:	149
4.2.3. Test Case: View User List in Database:.....	152
4.2.4. Test Case: Exam Registration:.....	154
4.2.5. Test Case: View Exam Result List in Database:	156
4.2.6. Test Case: Create Question (Question to be registered in Database):	158
4.2.7. Test Case: Add Answers (Answers to be Registered in Database):	161
4.2.8. Test Case: Calling ‘set_password’ using other models than Django User Model:	163
4.2.9. Test Case: To enter staff page directly without creating a single user at the very beginning:.....	164
4.2.10. Test Case: To add existing field name in Django User Admin:	166
4.3. System Testing:	168
4.4. Critical Analysis:	169
4.4.1. Test material:	169
4.4.2. Evaluation of testing:	169
4.4.3. Conclusion (Test Analysis):.....	170
5. Conclusion:	171
5.1. Legal, Social and Ethical Issues:	171

5.1.1. Legal Issues:	171
5.1.2. Social Issues:	172
5.1.3. Ethical Issues:	173
5.2. Advantages:.....	174
5.3. Limitations:.....	176
5.4. Future work:.....	177
5.4.1. Unification with Learning Management System:	177
5.4.2. Mobile compatibility:.....	177
5.4.3. Integrating multiple languages:	177
5.4.4. Adaptive testing:.....	177
5.4.5. Accessibility:.....	177
5.4.6. Integration of AI:.....	177
5.4.7. Integration of data science:	177
5.4.8. Allowing the test-takers to conduct examination with different answer types:	177
5.4.9. Blockchain Integration:.....	177
5.4.10. Interaction Tools:.....	177
5.4.11. User personalisation features:.....	177
6. Bibliography:	178
Bibliography	178
7. Appendix:	182
7.1. Appendix A: Pre-Survey:	182
7.1.1. Pre-Survey Form:.....	182
7.1.2. Sample of Filled Pre-Survey Form:	185
7.1.3. Pre-Survey Result:	188
7.2. Appendix B: Post-Survey:.....	194
7.2.1. Post-Survey Form:	194
7.2.2. Sample of Filled Post-Survey Form:.....	197
7.2.3. Post-Survey Result:	202
7.3. Appendix C: Sample Codes:.....	207
7.3.1. Sample Code of the UI:.....	207
7.3.2. Sample Code of the View:.....	213

7.4. Designs:.....	219
7.4.1. Gantt Chart:.....	219
7.4.2. Work Breakdown Structure:	222
7.4.3. Milestone:.....	223
7.4.4. Wireframe:.....	224
7.4.5. Prototype (UI/UX (Graphic designing/Page layout)):.....	237
7.5. Screenshots of the system:	252
7.5.1. Home Page:	252
7.5.2. Staff:.....	253
7.5.3. Teacher:.....	269
7.5.4. Student:.....	278
7.5.5. Database:.....	288
7.5.6. Admin:.....	288
7.6. Appendix F: User Feedback:	289
7.6.1. User Feedback Form:	289
7.6.2. Sample of the Filled User Feedback Form:	291
7.7. Appendix G: Future Work:	294
7.7.1. Future Work (Elaborated):.....	294
7.7.2. Readings for the Future Work:	302
7.8. Similar Projects (Elaborated):	306
7.8.1. Speed Exam:.....	306
7.8.2. Eklavya:.....	309
7.8.3. Class Marker:	312
7.9. Considered Methodology (Elaborated):	314
7.9.1. Waterfall methodology:.....	314
7.9.2. Agile methodology:.....	316
7.9.3. Rational Unified Process (RUP):	318
7.10. Survey results analysis:	320
7.10.1. Pre-Survey results:.....	320
7.10.2. Post-survey results:.....	326
7.11. System Testing:	330
7.11.1. User Login:.....	330

7.11.2. User registration:.....	336
7.11.3. Test Case: To View User List:.....	344
7.11.4. Test Case: To Remove User:.....	346
7.11.5. Exam registration:	348
7.11.6. Test case: To View exam list:.....	356
7.11.7. Test Case: To View Exam Result List:	357
7.11.8. Class Registration:	359
7.11.9. Section Registration:	363
7.11.10. Test Case: Subject Registration (Valid subject Information is given):	367
7.11.11. Test Case: Staff Position Registration (Valid Staff Position Information is given):	369
7.11.12. Test Case: To Post Notice (Valid Notice Information is given):.....	371
7.11.13. Test Case: To view notice:	373
7.11.14. To Create Question:.....	376
7.11.15. Test Case: To View Question List:	381
7.11.16. Attend Examination:	383
7.11.17. To Change Password:.....	396

Table of Figures

Figure 1 Paper-based evaluation (Gomes, 2020)	5
Figure 2 Pile of papers (user4541274, n.d.)	5
Figure 3 Scholar as a solution (The application itself)	6
Figure 4 Prototyping Model Phases, (Martin, 2022)	16
Figure 5 Use Case diagram (Admin)	19
Figure 6 Staff (Use Case diagram).....	20
Figure 7 Staff Login (Wireframe)	22
Figure 8 Staff Login (Prototype Design)	23
Figure 9 Staff Login (Activity Diagram).....	24
Figure 10 Staff Login (Sequence Diagram)	25
Figure 11 Staff registration (Wireframe)	27
Figure 12 Teacher registration (Wireframe)	28
Figure 13 Student registration (Wireframe)	29
Figure 14 Staff registration (Prototype)	30
Figure 15 Teacher registration (Prototype).....	31
Figure 16 Student registration (Prototype)	32
Figure 17 Staff registration (Activity Diagram)	33
Figure 18 Teacher registration (Activity Diagram)	34
Figure 19 Student registration (Activity Diagram).....	35
Figure 20 Staff registration (Sequence Diagram)	36
Figure 21 Teacher registration (Sequence Diagram)	36
Figure 22 Student registration (Sequence Diagram)	37
Figure 23 View Staff List (Activity Diagram)	39
Figure 24 View Teacher List (Activity Diagram)	40
Figure 25 View Student List (Activity Diagram)	41
Figure 26 View Staff List (Sequence Diagram)	42
Figure 27 View Teacher List (Sequence Diagram).....	43
Figure 28 View Student List (Sequence Diagram)	44
Figure 29 Remove Staff (Activity Diagram)	46
Figure 30 Remove Teacher (Activity Diagram)	47
Figure 31 Remove Student (Activity Diagram)	48
Figure 32 Remove Staff (Sequence Diagram)	49
Figure 33 Remove Teacher (Sequence Diagram).....	50
Figure 34 Remove Student (Sequence Diagram)	51
Figure 35 Register Exam (By Staff) (Activity Diagram).....	53
Figure 36 Register Exam (By Staff) (Sequence Diagram).....	53
Figure 37 View Exam List (Activity Diagram)	55
Figure 38 View Exam List (Sequence Diagram).....	56
Figure 39 Register Class (Activity Diagram).....	57
Figure 40 Register Class (Sequence Diagram).....	58
Figure 41 Register Section (Activity Diagram).....	59

Figure 42 Register Section (Sequence Diagram)	60
Figure 43 Register subject (Activity Diagram)	61
Figure 44 Register Subject (Sequence Diagram)	62
Figure 45 Register Staff Position (Activity Diagram)	64
Figure 46 Register Staff Position (Sequence Diagram)	64
Figure 47 Post Notice (Wireframe)	66
Figure 48 Post Notice (Prototype)	67
Figure 49 Post Notice (Activity Diagram)	68
Figure 50 Post Notice (Sequence Diagram)	69
Figure 51 View Notice (Staff Page) (Activity Diagram)	70
Figure 52 View Notice (Staff Page) (Sequence Diagram)	71
Figure 53 Teacher (Use Case diagram)	75
Figure 54 Teacher Login (Wireframe)	77
Figure 55 Teacher Login (Prototype Design)	78
Figure 56 Teacher Login (Activity Diagram)	79
Figure 57 Teacher Login (Sequence Diagram)	80
Figure 58 Register Exam (Activity Diagram)	81
Figure 59 Register Exam (Sequence Diagram)	82
Figure 60 Create Question (Wireframe)	84
Figure 61 Create Question (Prototype)	85
Figure 62 Create Question (Activity Diagram)	86
Figure 63 Create Question (Sequence Diagram)	87
Figure 64 Question List (Activity Diagram)	89
Figure 65 Question List (Sequence Diagram)	90
Figure 66 Exam Result List (Activity Diagram)	92
Figure 67 Exam Result List (Sequence Diagram)	93
Figure 68 Post notice (Activity Diagram)	95
Figure 69 Post notice (Sequence Diagram)	96
Figure 70 View Notice (Activity Diagram)	97
Figure 71 View Notice (Sequence Diagram)	98
Figure 72 Change password (teacher) (Activity Diagram)	100
Figure 73 Change password (teacher) (Sequence Diagram)	101
Figure 74 Student (Use Case diagram)	102
Figure 75 Student Login (Wireframe)	104
Figure 76 Student Login (Prototype Design)	105
Figure 77 Student Login (Activity Diagram)	106
Figure 78 Student Login (Sequence Diagram)	107
Figure 79 Attend Examination (Wireframe)	109
Figure 80 Attend Examination (Prototype)	110
Figure 81 Attend Examination (Activity Diagram)	111
Figure 82 Attend Exam (Sequence Diagram)	112
Figure 83 View Result (Wireframe)	114

Figure 84 View Result (Prototype)	115
Figure 85 View Result (Activity Diagram)	116
Figure 86 View Result (Sequence Diagram)	117
Figure 87 View Notice (Activity Diagram)	118
Figure 88 Change password (student) (Activity Diagram)	121
Figure 89 Change password (student) (Sequence Diagram)	122
Figure 90 Use Case Diagram.....	123
Figure 91 Main system architecture	124
Figure 92 System architecture for examination	125
Figure 93 Entity Relationship Diagram.....	126
Figure 94 Register apps	127
Figure 95 account app (model)	128
Figure 96 account app (model)(continued1).....	129
Figure 97 account app (model)(continued 2).....	129
Figure 98 account app (model) (continued 3).....	130
Figure 99 exam material app (model).....	131
Figure 100 exam material app (model) (continued 1).....	132
Figure 101 exam material app (model) (continued 2).....	132
Figure 102 main app (model)	133
Figure 103 View for account app (Registration)	134
Figure 104 View for account app (Login).....	134
Figure 105 View for exam material app	135
Figure 106 View for main ap	135
Figure 107 Template for account app.....	136
Figure 108 Template for exam material app.....	137
Figure 109 Template for main app	138
Figure 110 Form for account app	139
Figure 111 Forms for exam material app	140
Figure 112 Forms for main app	141
Figure 113 Main URL of the project.....	142
Figure 114 URL (account app)	142
Figure 115 URL (exam material app)	143
Figure 116 URL (main app).....	144
Figure 117 URL (main app) (continued)	145
Figure 118 To create admin (superuser).....	147
Figure 119 To create admin (superuser). (Test successful)	148
Figure 120 To test user registration.	150
Figure 121 To test user registration. (Test successful)	150
Figure 122 To test user registration (continued). (Test successful).....	151
Figure 123 To view user list in database. (Test successful)	153
Figure 124 Exam Registration.....	155
Figure 125 Exam Registration (Test successful)	155

Figure 126 To view exam result list in database. (Test successful)	157
Figure 127 To create question (question to be registered in database).	159
Figure 128 To create question (question to be registered in database). (continued) ..	159
Figure 129 To create question (question to be registered in database). (Test successful)	160
Figure 130 To create answers (answer to be registered in database).	162
Figure 131 To create answers (answer to be registered in database). (Test successful)	162
Figure 132 Calling 'set_password' using other models than Django User Model.....	163
Figure 133 To enter staff page through URL directly without creating a single user at the very beginning. (Test successful)	165
Figure 134 To add existing field name in Django User Admin. (Test successful).....	167
Figure 135 Pre-Survey Form (fig 1).....	182
Figure 136 Pre-Survey Form (fig 2).....	183
Figure 137 Pre-Survey Form (fig 3).....	183
Figure 138 Pre-Survey Form (fig 4).....	184
Figure 139 Sample of Filled Pre-Survey Form (fig 1)	185
Figure 140 Sample of Filled Pre-Survey Form (fig 2)	186
Figure 141 Sample of Filled Pre-Survey Form (fig 3)	187
Figure 142 Pre-Survey Result (fig 1)	188
Figure 143 Pre-Survey Result (fig 2)	188
Figure 144 Pre-Survey Result (fig 3)	189
Figure 145 Pre-Survey Result (fig 4)	189
Figure 146 Pre-Survey Result (fig 5)	190
Figure 147 Pre-Survey Result (fig 6)	190
Figure 148 Pre-Survey Result (fig 7)	191
Figure 149 Pre-Survey Result (fig 8)	192
Figure 150 Pre-Survey Result (fig 9)	193
Figure 151 Post-Survey Form (fig1)	194
Figure 152 Post-Survey Form (fig2)	195
Figure 153 Post-Survey Form (fig3)	196
Figure 154 Post-Survey Form (fig4)	197
Figure 155 Sample of Filled Post-Survey Form (fig1).....	198
Figure 156 Sample of Filled Post-Survey Form (fig 2).....	199
Figure 157 Sample of Filled Post-Survey Form (fig3).....	200
Figure 158 Sample of Filled Post-Survey Form (fig 5).....	201
Figure 159 Post-Survey Result (fig 1)	202
Figure 160 Post-Survey Result (fig 2)	202
Figure 161 Post-Survey Result (fig 3)	203
Figure 162 Post-Survey Result (fig 4)	203
Figure 163 Post-Survey Result (fig 5)	204
Figure 164 Post-Survey Result (fig 6)	204

Figure 165 Post-Survey Result (fig 7)	205
Figure 166 Post-Survey Result (fig 8)	206
Figure 167 Index page (UI)	207
Figure 168 Teacher Registration (UI)	208
Figure 169 Teacher Login (UI)	208
Figure 170 Teacher Notice Board (UI)	209
Figure 171 Staff Page UI.....	209
Figure 172 Quiz page (UI).....	210
Figure 173 For Class Section (Template).....	210
Figure 174 For Class Section(continued) (Template).....	211
Figure 175 Create Question (Template).....	211
Figure 176 Class Registration (Template).....	212
Figure 177 Index page (View)	213
Figure 178 Teacher Registration (View).....	214
Figure 179 Teacher Login (View)	214
Figure 180 Quiz Main (View)	215
Figure 181 Create Question (View).....	215
Figure 182 Save Quiz (View)	216
Figure 183 Section selection (View)	216
Figure 184 Generic View.....	217
Figure 185 Class Registration (View)	217
Figure 186 User Logout (View)	218
Figure 187 Gantt Chart (full view)	219
Figure 188 Gantt chart (Larger view 1).....	220
Figure 189 Gantt chart (Larger view 2).....	221
Figure 190 Work Breakdown Structure	222
Figure 191 Milestone.....	223
Figure 192 Home page (Wireframe)	224
Figure 193 Student Registration (Wireframe).....	225
Figure 194 Staff Registration (Wireframe).....	226
Figure 195 Teacher Registration (Wireframe)	227
Figure 196 Student Login (Wireframe)	228
Figure 197 Teacher Login (Wireframe)	229
Figure 198 Staff Login (Wireframe)	230
Figure 199 Set questions (Wireframe).....	231
Figure 200 Take the exam (Wireframe).....	232
Figure 201 Notice Board (Wireframe).....	233
Figure 202 Staff portal (Wireframe)	234
Figure 203 Teacher Portal (Wireframe)	235
Figure 204 Student Portal (Wireframe).....	236
Figure 205 Student Portal view results (Wireframe)	237
Figure 206 Home page (Prototype design)	238

Figure 207 Student Registration (Prototype design).....	239
Figure 208 Teacher Registration (Prototype design).....	240
Figure 209 Staff Registration (Prototype design)	241
Figure 210 Student Login (Prototype design)	242
Figure 211 Teacher Login (Prototype design)	243
Figure 212 Staff Login (Prototype design).....	244
Figure 213 Student Portal (Prototype design)	245
Figure 214 Teacher Portal (Prototype design)	246
Figure 215 Staff Portal (Prototype design)	247
Figure 216 Conduct examination (Prototype design)	248
Figure 217 Take examination (Prototype design).....	249
Figure 218 View result (Prototype design)	250
Figure 219 Post notice (Prototype design)	251
Figure 220 Scholar Home Page	252
Figure 221 Scholar Home Page (continued)	252
Figure 222 Staff Login	253
Figure 223 Staff Portal	253
Figure 224 Student Registration.....	254
Figure 225 Teacher Registration.....	254
Figure 226 Staff Registration.....	255
Figure 227 Student List	255
Figure 228 Teacher List	256
Figure 229 Staff List	256
Figure 230 Student Detail.....	257
Figure 231 Teacher Detail.....	257
Figure 232 Staff Detail.....	258
Figure 233 Delete student.....	258
Figure 234 Delete Teacher.....	259
Figure 235 Delete Staff	259
Figure 236 Exam Registration.....	260
Figure 237 Exam List	260
Figure 238 Exam Detail.....	261
Figure 239 Exam Update	261
Figure 240 Delete Exam.....	262
Figure 241 Exam result list.....	262
Figure 242 Class registration	263
Figure 243 Section registration	264
Figure 244 Subject registration	264
Figure 245 Staff Position registration	265
Figure 246 Staff Notice Board	265
Figure 247 Add Notice.....	266
Figure 248 Staff View Notice.....	266

Figure 249 Staff setting	267
Figure 250 Staff Update	267
Figure 251 Staff change password.....	268
Figure 252 Teacher Login.....	269
Figure 253 Teacher Portal.....	269
Figure 254 Exam registration	270
Figure 255 Create question.....	270
Figure 256 Create question (continued)	271
Figure 257 Question List	271
Figure 258 Question Detail.....	272
Figure 259 Delete question	272
Figure 260 Update question	273
Figure 261 Update question (continued)	273
Figure 262 Exam result list.....	274
Figure 263 Teacher Notice Board	274
Figure 264 Add notice	275
Figure 265 View Notice	275
Figure 266 Teacher setting	276
Figure 267 Teacher Update	276
Figure 268 Teacher change password.....	277
Figure 269 Login credentials provided to student.....	278
Figure 270 Student Login	279
Figure 271 Student Portal	279
Figure 272 Available Exams.....	280
Figure 273 Begin Exam.....	280
Figure 274 Attend Exam.....	281
Figure 275 Attend Exam (continued)	281
Figure 276 Student View Result.....	282
Figure 277 Time over alert message.....	282
Figure 278 Student Notice View	283
Figure 279 Student settings	283
Figure 280 Student Update Information	284
Figure 281 Student Change Password.....	284
Figure 282 Student Change Password successful confirmation	285
Figure 283 Student forgot password	285
Figure 284 Student forgot password email sent confirmation.....	286
Figure 285 Link sent to student email to reset new password (in case of forgot password).....	286
Figure 286 Student forgot password set new password.....	287
Figure 287 Student forgot password (new password reset successful confirmation) ..	287
Figure 288 Database.....	288
Figure 289 Admin dashboard	288

Figure 290 User Feedback Form (fig 1)	289
Figure 291 User Feedback Form (fig 2)	290
Figure 292 User Feedback Form (fig 3)	290
Figure 293 Sample Filled User Feedback Form (fig 1).....	291
Figure 294 Sample Filled User Feedback Form (fig 2).....	292
Figure 295 Sample Filled User Feedback Form (fig 3).....	293
Figure 296 Unification with LMS (Prototype Design).....	294
Figure 297 Mobility compatibility (Prototype Design)	295
Figure 298 Integrating multiple languages (Prototype Design).....	296
Figure 299 Integration of Data Science (Prototype Design)	298
Figure 300 Allowing the test-takers to conduct examination with different answer types (Prototype Design)	299
Figure 301 User personalisation features (Preferred Interface) (Prototype Design)....	301
Figure 302 Working with pandas on Django.....	302
Figure 303 Using speech recognition feature	303
Figure 304 Implementation of multiple languages	303
Figure 305 Creative way of learning with the implementation of white board.....	304
Figure 306 After unification with Learning Management System; implementation of course recommendation.....	305
Figure 307 After unification with Learning Management System; implementation of course recommendation (continued)	305
Figure 308 Speed Exam dashboard.....	306
Figure 309 Speed Exam create a new exam	307
Figure 310 Speed Exam candidates list.....	307
Figure 311 Speed Exam add new question.....	308
Figure 312 Eklavya dashboard.....	309
Figure 313 Eklavya exam creation	310
Figure 314 Eklavya assign to exam.....	310
Figure 315 Eklavya candidate results.....	311
Figure 316 Eklavya assign examiner	311
Figure 317 Class Marker dashboard	312
Figure 318 Class Marker create test	313
Figure 319 Class Marker set questions	313
Figure 320 Waterfall Methodology (Team, 2022)	314
Figure 321 Agile methodology (InterQuality, n.d.)	316
Figure 322 Rational Unified Process (RUP) (Janse, 2019)	319
Figure 323 Survey Summary.....	320
Figure 324 Have you ever used an online based or computer-based examination? ...	320
Figure 325 Do you prefer your examination to be online or computer system based?	321
Figure 326 Have you ever come across such application?	321
Figure 327 Can it be the new and better way of evaluating papers?	322
Figure 328 Would you suggest it to be used in educational institutes?	322

Figure 329 Online examination is better than the traditional approach of conducting the examination. Why or why not? (Series 1).....	323
Figure 330 Online examination is better than the traditional approach of conducting the examination. Why or why not? (Series 2).....	324
Figure 331 Survey summary	326
Figure 332 How would you rate your application experience?	326
Figure 333 How was the user interface and user experience of the application?	327
Figure 334 Do you think this application would help the purpose of eradicating traditional approach of conducting examination?	327
Figure 335 Would you recommend educational institute to implement this application?	328
Figure 336 Do you think this application is an easier way to conduct and attend examination?	328
Figure 337 Was the application easy to use and navigate through different features?329	
Figure 338 To test user login (non-existing username is given).	331
Figure 339 To test user login (non-existing username is given). (Test successful)	331
Figure 340 To test user login (invalid password is given).....	333
Figure 341 To test user login (invalid password is given). (Test successful).....	333
Figure 342 To test user login (valid username and password are given).	335
Figure 343 To test user login (valid username and password are given). (Test successful)	335
Figure 344 To test user registration (invalid email is given).	337
Figure 345 To test user registration (invalid email is given). (Test successful)	338
Figure 346 To test user registration (not fluffing password requirements).....	340
Figure 347 To test user registration (not fluffing password requirements). (Test successful)	341
Figure 348 To test user registration (valid user information is given). (Test successful)	343
Figure 349 To test view user list. (Student List)	345
Figure 350 To test view user list. (Teacher List).....	345
Figure 351 To test view user list. (Staff List)	345
Figure 352 To remove user.	347
Figure 353 To remove user. (Test Successful)	347
Figure 354 To register exam (negative number of questions is given). (Test successful)	349
Figure 355 To register exam (negative time is given). (Test successful).....	351
Figure 356 To register exam (negative score to pass is given). (Test successful)	353
Figure 357 To register exam (valid exam information is given). (Test successful)	355
Figure 358 To view exam list. (Test successful).....	356
Figure 359 To view exam result list. (Test successful)	358
Figure 360 To view exam result list. (continued) (Test Successful)	358
Figure 361 To test class registration (negative class is given). (Test successful)	360

Figure 362 To test class registration (valid class information is given). (Test successful)	362
Figure 363 To test section registration (class is not selected). (Test successful).....	364
Figure 364 To test section registration (valid section information is given).	366
Figure 365 To test section registration (valid class information is given). (Test successful)	366
Figure 366 To test subject registration (valid subject information is given).	368
Figure 367 To test subject registration (valid subject information is given). (Test successful)	368
Figure 368 To test staff position registration (valid staff position information is given).370	
Figure 369 To test staff position registration (valid staff position information is given). (Test successful)	370
Figure 370 To test post notice (valid notice information is given).....	372
Figure 371 To test post notice (valid notice information is given). (Test successful)...	372
Figure 372 To view notice. (Test successful) (Staff Notice View).....	374
Figure 373 To view notice. (Test successful) (Teacher Notice View)	374
Figure 374 To view notice. (Test successful) (Student Notice View)	375
Figure 375 To add multiple answers	377
Figure 376 To add multiple answers (Test successful)	377
Figure 377 To create question.	379
Figure 378 To create question. (Fill answers)	379
Figure 379 To create question. (Test successful)	380
Figure 380 To view question list. (Test successful)	382
Figure 381 To view available exams. (Test successful)	384
Figure 382 To attend the particular exam. (Test successful).....	386
Figure 383 To attend the particular exam. (Test successful).....	386
Figure 384 To attend the particular exam. (continued) (Test successful).....	387
Figure 385 To test the working of the countdown timer (begin timer). (Test successful)	388
Figure 386 To test the working of the countdown timer (end timer). (Pop-up message) (Test successful)	390
Figure 387 To test the working of the countdown timer (end timer). (Result is displayed) (Test successful)	390
Figure 388 To submit exam paper. (Test successful)	392
Figure 389 To submit exam paper. (continued) (Test successful).....	392
Figure 390 To submit exam paper. (Test successful)	393
Figure 391 To view results (for student). (Test successful)	395
Figure 392 To Change Password (incorrect old password is entered) (Test successful)	397
Figure 393 To Change Password (new password confirmation doesn't match the new password) (Test successful).....	399
Figure 394 To Change Password (If the entered details are valid):	401

Table of Tables

Table 1 Comparison of the projects	14
Table 2 Staff Login (High level use case).....	21
Table 3 Register student/teacher (High level use case)	26
Table 4 Register staff (High level use case).....	26
Table 5 View staff, teacher, student list (High level use case)	38
Table 6 Remove staff, teacher, student (High level use case)	45
Table 7 Register Exam (By Staff) (High level use case).....	52
Table 8 View Exam List (High level use case)	54
Table 9 Register Class (High level use case).....	57
Table 10 Register Section (High level use case).....	59
Table 11 Register Subject (High level use case).....	61
Table 12 Register Staff Position (High level use case).....	63
Table 13 Post Notice (High level use case).....	65
Table 14 View Notice (Staff) (High level use case)	70
Table 15 Change password (Staff) (High level use case)	72
Table 16 Change password (staff) (Activity Diagram)	73
Table 17 Change password (staff) (Sequence Diagram)	74
Table 18 Teacher Login (High level use case)	76
Table 19 Register Exam (By Teacher) (High level use case).....	81
Table 20 Create Question (High level use case)	83
Table 21 Question List (High level use case)	88
Table 22 View exam result list (High level use case)	91
Table 23 Post Notice (High level use case).....	94
Table 24 View Notice	97
Table 25 Change password (teacher) (High level use case).....	99
Table 26 Student Login (High level use case)	103
Table 27 Attend Examination (High level use case)	108
Table 28 View Result (Student) (High level use case)	113
Table 29 View notice (High level use case).....	118
Table 30 Change password (student) (High level use case)	120
Table 31 To create admin (superuser).	147
Table 32 To test user registration (valid user information is given).	149
Table 33 To view user list in database.	152
Table 34 Exam Registration.	154
Table 35 To view exam result list in database.....	156
Table 36 To create question (question to be registered in database).....	158
Table 37 To create answers (answer to be registered in database).....	161
Table 38 Calling 'set_password' using other models than Django User Model.	163

Table 39 To enter staff page through URL directly without creating a single user at the very beginning.....	164
Table 40 To add existing field name in Django User Admin.....	166
Table 41 To test user login (non-existing username is given).....	330
Table 42 To test user login (invalid password is given).....	332
Table 43 To test user login (valid username and password are given).....	334
Table 44 To test user registration (invalid email is given).....	336
Table 45 To test user registration (not fluffing password requirements).....	339
Table 46 To test user registration (valid user information is given).....	342
Table 47 To test view user list.....	344
Table 48 To remove user.....	346
Table 49 To register exam (negative number of questions is given).....	348
Table 50 To register exam (negative time is given).....	350
Table 51 To register exam (negative score to pass is given).....	352
Table 52 To register exam (valid exam information is given).....	354
Table 53 To view exam list.....	356
Table 54 To view exam result list.....	357
Table 55 To test class registration (negative class is given).....	359
Table 56 To test class registration (valid class information is given).....	361
Table 57 To test section registration (class is not selected).....	363
Table 58 To test section registration (valid class information is given).....	365
Table 59 To test subject registration (valid subject information is given).....	367
Table 60 To test staff position registration (valid staff position information is given).....	369
Table 61 To test post notice (valid notice information is given).....	371
Table 62 To view notice.....	373
Table 63 To add multiple answers.....	376
Table 64 To create question.....	378
Table 65 To view question list.....	381
Table 66 To view available exams.....	383
Table 67 To attend the particular exam.....	385
Table 68 To test the working of the countdown timer (begin timer).....	388
Table 69 To test the working of the countdown timer (end timer).....	389
Table 70 To submit exam paper.....	391
Table 71 To view results (for student).....	394
Table 72 To Change Password (incorrect old password is entered).....	396
Table 73 To Change Password (new password confirmation doesn't match the new password).....	398
Table 74 To Change Password (If the entered details are valid):.....	400

1. Introduction:

In this world, the outcomes of several tests are used to carry out anything that must be upgraded. Whether testing software, programs, or people. This application is built around that idea. To be specific, the application's main idea is to skill test individuals at any level under the corresponding problem sets assigned by the teacher. This program can be used to completely replace the outdated and conventional method of conducting and attending exams. Since technology has recently taken over the world. Everything appears to be improving over time, from food delivery to consumers' homes to financial sectors with online payment systems. The same educational system and the way it is managed appear to be trailing despite the daily advancement of technology. Why can't the educational sector employ the most advanced technology but sensitive industries like banking and medicine can?

People who have received an education in a field are better equipped to think, feel, and act in ways that promote success and increase both their own and their community's degree of happiness (Al-Shuaibi, 2014).

The online testing system is presently viewed as a quickly developing examination approach because of its accuracy and pace. Additionally, fewer employees are needed to run the exam. Due to the reduction in student exam time, practically all businesses now administer tests using online testing platforms. Organizations can also easily monitor the progress of the student they give a test. The calculation of the outcome is quicker as a result. It also helps to lessen the need for paper (Muna R. Hameed, 2017).

In China, education in basic computer operating skills has been widely adopted. The skills encompass those needed to use Windows, Microsoft Office, networking, etc. an electronic government foundation integrated with a range of courses. Today, all undergraduate students must pass the Computer Skills course, and the corresponding Computer Operating Exams are prerequisites for all city officials. Additionally, high schools are now implementing fundamental computer education. Since the late 1990s, thousands of people in Zhejiang Province have taken part in various levels of computer education and testing. A Web-based learning and testing system must be created to

successfully handle the issue of mass learning and evaluation of fundamental computer education (Yuan Zhenming, 2003).

1.1. Project Description:

Scholar is an effective web-based online examination system platform for administering tests and assessments remotely created in Django and JavaScript. The system makes use of two well-known technologies, the Python-based web framework Django, and the flexible client-side scripting language JavaScript, to create a dependable and scalable platform that can meet the demands of organizations of all sizes in the education and commercial sectors. The use of Django enables a lot of features, such as user authentication, database management, and administration capabilities. It enables test creators to design, plan, and manage exams and offers a wide range of functionalities like multi-level authentication, timed exams, and automatic grading. The use of JavaScript also enhances user experience by providing a more interactive and dynamic interface that makes it easier for students to navigate the exam, assess their progress, and get rapid feedback.

By mixing Django and JavaScript, the online exam system is made scalable, responsive, and user-friendly. It also enables the development of a wide range of custom features and services, such as real-time reporting, analytics, and data visualization, in order to satisfy the unique requirements of distinct organizations. For institutions looking for a rapid, secure, and user-friendly remote learning and testing platform, the Django and JavaScript-based online exam system is ideal. By introducing greater flexibility, convenience, and cost-effectiveness, it has changed the traditional exam process, and in the current digital era, it is progressively turning into a crucial tool for businesses and educational institutions.

1.2. Current scenario:

Because of its precision and speed, the online testing system is currently thought of as a rapidly evolving examination approach. Less personnel are required to conduct the exam as well. Nearly all organizations now use online testing systems to deliver exams due to the reduction in student exam time. Additionally, organizations may quickly track a student's development after giving them a test. As a result, the calculation of the result is quicker. Additionally, it helps reduce the need for paper (Muna R. Hameed, 2017).

Basic computer operating skills instruction has become very popular in China. The abilities cover everything from networking to using Windows, Microsoft Office, and an electronic government foundation coupled with a variety of courses. Today, all undergraduate students are required to pass the Computer Skills course, and all city officials must pass the related Computer Operating Exams. High schools are also increasingly offering introductory computer courses. In Zhejiang Province, thousands of people have participated in various levels of computer instruction and testing since the late 1990s. To properly manage the challenge of mass learning and assessment of fundamental computer education, a web-based learning and testing system must be developed (Yuan Zhenming, 2003).

1.3. Problem domain:

Considering how little time and effort is required to examine exam papers and produce results reports, online examination systems are very helpful to educational institutions for exam preparation. It is simpler for educational institutions to monitor students' progress when they use online tests. The Scholastic Institute and training facilities employ this strategy most effectively since it makes it easier to manage the tests and efficiently get results. Up until recently, exam and result preparation were done manually, which required more time to complete (Muna R. Hameed, 2017).

The drawbacks examining the traditional way through the paper-based are discussed below.

- i) Administration of evaluations using paper requires a lot of time. It can take many months to complete the cycle of administering the forms, collecting, and analysing the data, disseminating the results, and acting on the feedback (Explorance, 2013).
- ii) The high expense of paper-based testing or evaluation is one of the main issues. The procedure, which includes printing the problem sets, giving out answer sheets to the participants, and paying the paper checker, ends in a bloated and pointless expenditure of money.
- iii) Method for gathering data. When everything is kept in a paper file and packed up, it is challenging to view and reuse the same record in the future.
- iv) Due to the exam papers' inflexible layout, there is less flexibility in the procedure. If the questions have errors, the document must be updated. Furthermore, it prohibits altering the questions.
- v) A response that needs human effort or has low confidence (for example, handwriting that is impossible to read, is invalid, or is not valid based on the established norms) must be reviewed. (pappersurvey, 2021).
- vi) A huge quantity of paper is used while giving tests on paper. Because of this, trees are needlessly taken down for paper.

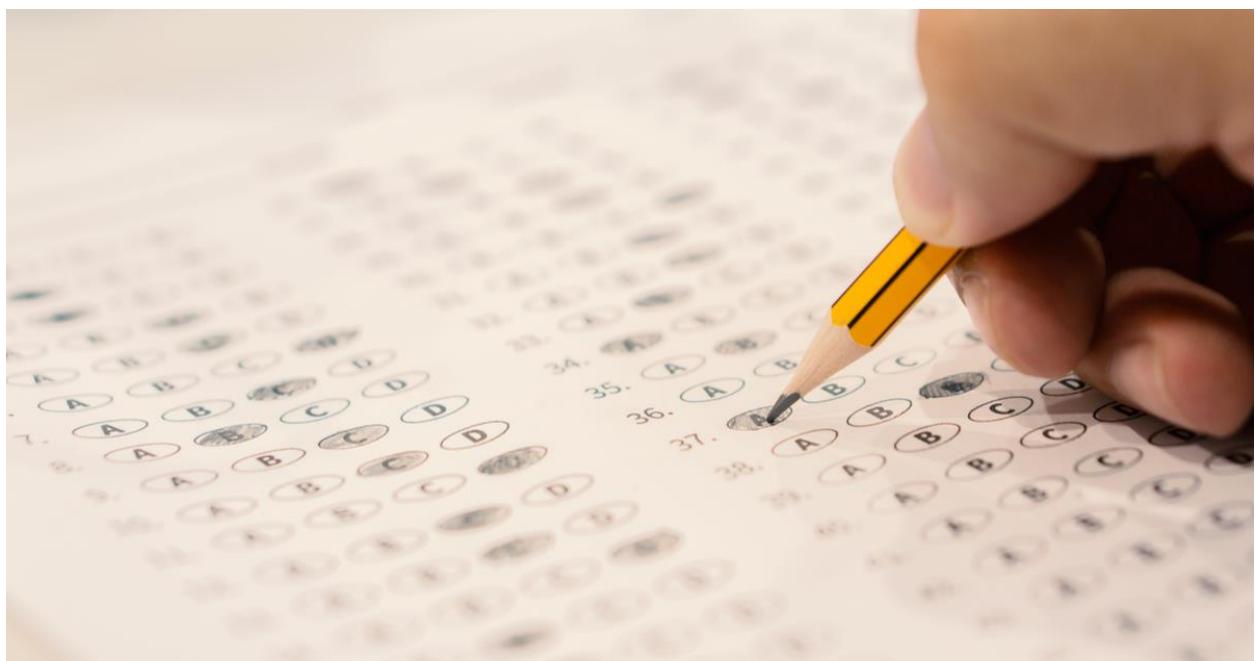


Figure 1 Paper-based evaluation (Gomes, 2020)

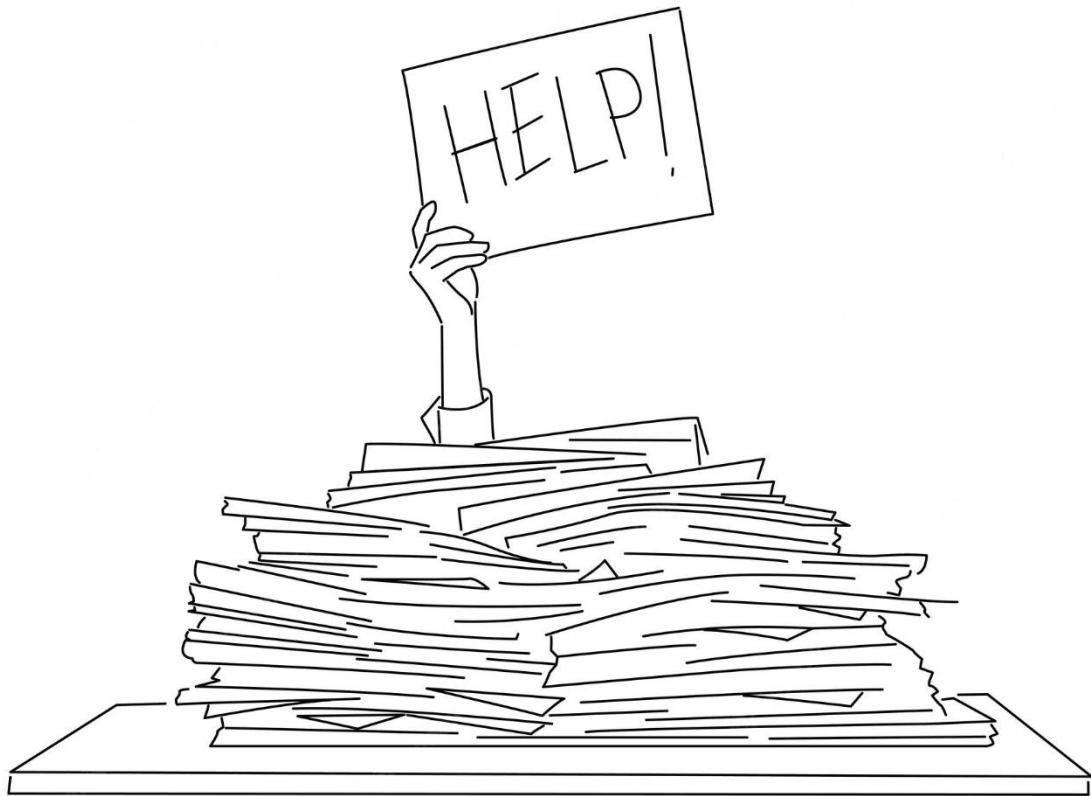


Figure 2 Pile of papers (user4541274, n.d.)

1.4. Problem Solution:



Figure 3 Scholar as a solution (The application itself)

- i) As a solution to the problem, using the mode of technology that allows the eradication of the use of paper-based evaluation and examination is the best option. This application provides an easy user interface for all the participating members.
- ii) Not only conducting and attending the evaluation process. The results will also be stated for each individual and will also be recorded for future purposes.
- iii) As all the information or the records are being stored in a database, it makes it a lot safer, more manageable, and easier to handle the data properly.
- iv) This application boosts the efficiency of work by properly utilizing the amount of force and reducing the time consumed.
- v) With less paper being used and discarded during paper-based testing, fewer trees will need to be cut down, which will help preserve the environment.
- vi) The cost of using and distributing materials to conduct an exam is also reduced by this application.
- vii) It resolves issues with poor and atrocious handwriting and correct or incorrect responses depending on predetermined standards.

- viii) This application is a suitable step to ensure that technology is used in education, enabling it to advance and improve institutional sectors.
- ix) The technology makes it far simpler to correct issues with printing mistakes and last-minute alterations than paper-based testing. When conducting critical reviews and examinations, this is quite beneficial.

1.5. Aims and objectives:

1.5.1. Aim:

The product aims to provide the enterprise with a system that allows replacing various evaluations and examinations that have been taken on a paper basis in the past. The product also allows them to store the record and the information of their participants and students and use that data for further purposes.

1.5.2. Objectives:

The objectives to achieve the aim are listed below:

- i) To understand the working mechanism of a web application and learn to express the product through it.
- ii) To research the features to be added to the application to make it fully useful.
- iii) To research the tools, ide, and different software to enhance the application itself.
- iv) To work with the data using the most appropriate database for better security and functionality of the product.
- v) To learn and implement API to leverage the existing code on the different ends of the application (front end and back end).
- vi) To provide a fast, efficient, secure, and trustable application to conduct and attend evaluation tests or examinations.
- vii) To design the application to be user-friendly in every context.

1.6. Report structure:

1.6.1. Background:

Background research is a fundamental component of the project. It not only provides a quick synopsis but also goes into detail on each project component. It contains the information required concerning the project as well as a description of the approach that was used in the project.

1.6.2. Development:

This section covers the amount of work completed to the date of this project report. It includes the prototype (UI/UX), wireframe, UML diagrams, and survey. It also covers the development and implementation part of the project.

1.6.3. Testing and Analysis:

Testing of various features included in the project is carried out under testing sector. Testing has been done as per the requirement of the features.

The results of these test cases are later discussed under the analysis part of the report.

1.6.4. Conclusion:

This sector discusses the about the advantages of the project as well its drawbacks or limitations. Proper reasoning has been provided to justify it. It also discusses about the future works to be put in the project.

2. Background:

2.1. About the end users:

The extraordinary evolution of the educational system over time has been greatly aided by technological advancement. Teachers can now easily track student achievement while also improving the learning environment thanks to learning management systems and smart classrooms (Admin, 2022).

The implementation of the online exam system significantly improved the previous evaluation strategy. Due to complete automation, the review process is today thorough, precise, and speedy. Along with making life easier for educators and teachers, this technological advancement has benefited students and test-takers (Admin, 2022).

2.2. Understanding the solution:

What exactly is an online examination system?

The complete schedule of an online test, including all its components and features, is referred to as the examination system. Online exam software is used by the online examination system to create, conduct, and evaluate tests. A variety of benefits come with this kind of testing technique. Some of these include the fact that it does away with the necessity for paper for the question-and-answer sheets and any form of manual work that would be excessive for an offline test (Admin, 2022).

Background on the working mechanism of the online examination system:

- a) Creation of test: A useful and compelling online test can be created by the administrator using a range of features. With the variety of question types provided by the online test development platform, a subjective, objective, or MCQ-style test can be easily created (Admin, 2022).
- b) Conducting tests: Conducting a test on a big scale is not a challenge with online tests. Online exam software can accommodate many students at once. Many candidates can

be tested effectively owing to the "candidate management" capability. It enables the test-taker to classify the candidates into the appropriate groups and, during the test, assign various tests to each group, conducting several tests at once. In addition, the students are divided into groups according to their grades or class (Admin, 2022).

c) Evaluation of test: The last and most important parts of any test are its result, report, and feedback. The full report system provided by the online test method is both immediate and incredibly accurate. The software may instantly score the test paper following marks assigned by the exam's creator. Following the test, a report is created with scores for each subject, section, and question (Admin, 2022).

2.3. Similar projects:

Similar projects elaboration in appendix section [*7.8. Similar Projects \(Elaborated\)*](#):

2.4. Comparison of the projects:

S.no	Features	Scholar (My project)	Speed Exam	Eklavya	Class Marker
1.	Register	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
2.	Login	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3.	Post Notice	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4.	View Notice	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
5.	Assign Staff	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
6.	Conduct Exam	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
7.	Take Exam	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
8.	Generate Report	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
9.	View Report	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

10.	Student credentials are provided through email	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
11.	Billing	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
12.	Notification	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
13.	Feedback chat app	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
14.	Friendly User Interface	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Table 1 Comparison of the projects

3. Development:

3.1. Considered Methodology:

- i) Waterfall methodology
- ii) Agile methodology
- iii) Rational Unified Process (RUP)
- iv) Prototyping Model (Evolutionary Prototyping)

Elaboration of considered methodology in appendix section *7.9. Considered Methodology (Elaborated)*:

3.2. Selected methodology:

3.2.1. Prototyping Model (Evolutionary Prototyping):

The chosen methodology for this application is evolutionary prototyping.

Using the evolutionary prototyping method, the software developer or development team first produces a prototype. The final product is created after collecting initial consumer feedback and creating subsequent prototypes, each with enhanced or expanded features (Sherrell, 2013).

Prototyping Model Phases (SDLC phases):

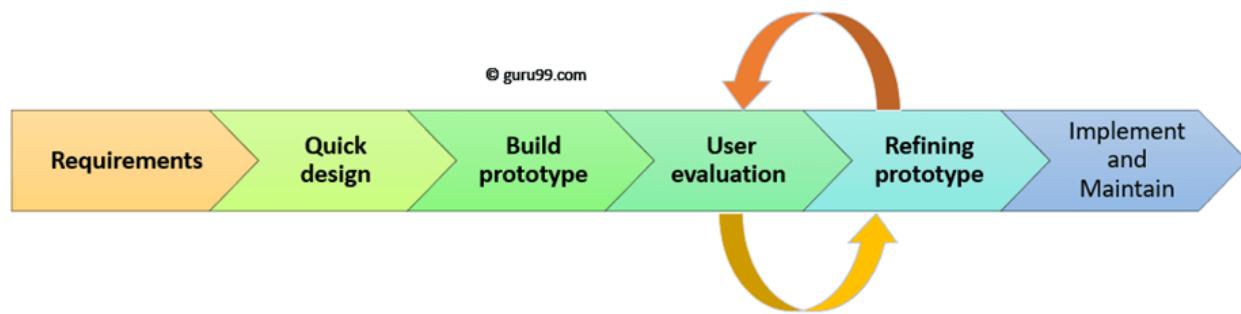


Figure 4 Prototyping Model Phases, (Martin, 2022)

Reasons for choosing this methodology:

- i) One of this methodology's main elements is its support for the product's evolving environment. This component makes it simpler for me to meet the client's needs more effectively.
- ii) This methodology offers a superior risk analysis as changes are encouraged which saves my time and resources.
- iii) This methodology requires user participation in the product's development phase to provide a better product.
- iv) There is room for improvement, which enables me to the addition of new and improved features to the product.
- v) With this methodology, errors and functional gaps are found right away, enabling me to develop a better product.

3.3. Phases of methodology:

i) Requirements gathering and analysis: The first step in prototyping is to do a requirement analysis. At this stage, the system's needs are thoroughly established. (Martin, 2022).

Throughout this process, I would ask the client questions (survey in this project) to determine what they wanted from the product and move forward with their answers.

ii) Quick design: The second stage could consist of a quick design or a rough design. During this stage, the system's fundamental design is developed. But it's not a fully realized design yet. It provides the user with a quick overview of the system. The rapid design aids in the development of the prototype (naimishsahu08, 2021).

To give the client information (survey information) on the product's basic design, I would therefore give them a quick design of the product.

iii) Build of a prototype: Using the information obtained during rapid design, I would create a true prototype of the product at this step. It is a condensed form of the necessary system.

iv) Initial user evaluation: After creating a prototype, I would give it to the client (survey in this project) for a preliminary assessment. The prototype-refinement process considers any necessary modifications and consumer requirements.

v) Prototype refinement: I would adjust the prototype as needed in response to the client's (survey in this project) recommendations and criticisms if they were unhappy with the present version. The refinement procedure is carried on until the client's demands are satisfied. I would develop a final system based on the customer-accepted prototype after the client (survey in this project) is satisfied with the finished outcome.

vi) Implementation of the product: When the final prototype is made, I will extensively test the system using the right testing techniques and put the final product to use.

vii) Maintenance of the product: I would perform routine maintenance on the product to ensure that it works well over the long term.

3.4. Survey results:

3.4.1. Pre-Survey results:

Pre-Survey results elaboration in appendix section [*7.10.1. Pre-Survey results:*](#)

3.4.2. Post-survey results:

Post-Survey results elaboration in appendix section [*7.10.2. Post-survey results:*](#)

3.5. Requirement analysis:

*Requirement analysis for admin has not been performed separately as admin is able to perform the activities of all 3 other actors (staff, teacher, and student).

3.5.1. For Admin:

3.5.1.1 Use case diagram:

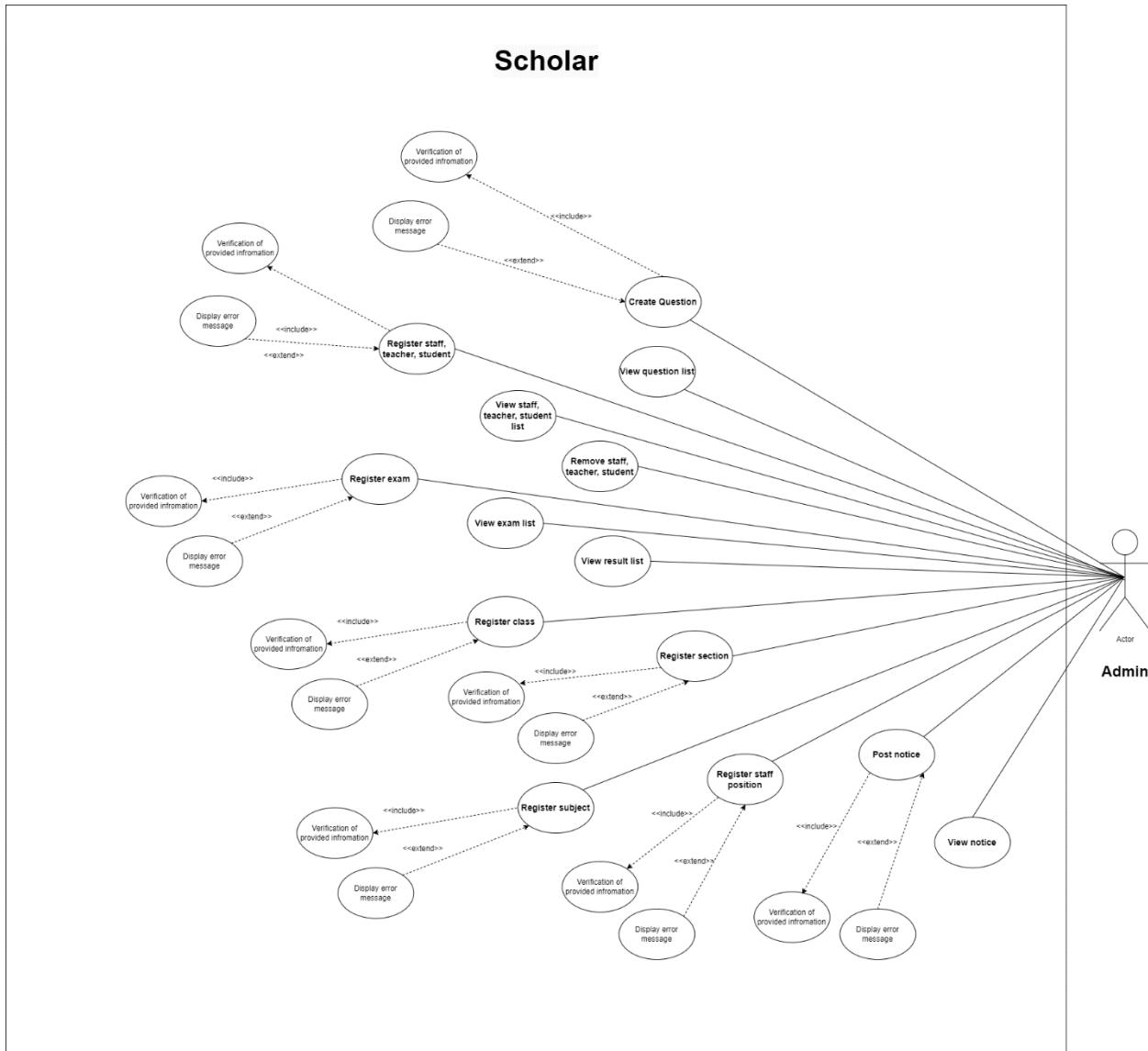


Figure 5 Use Case diagram (Admin)

3.5.2. For Staff:

3.5.2.1. Use case diagram:

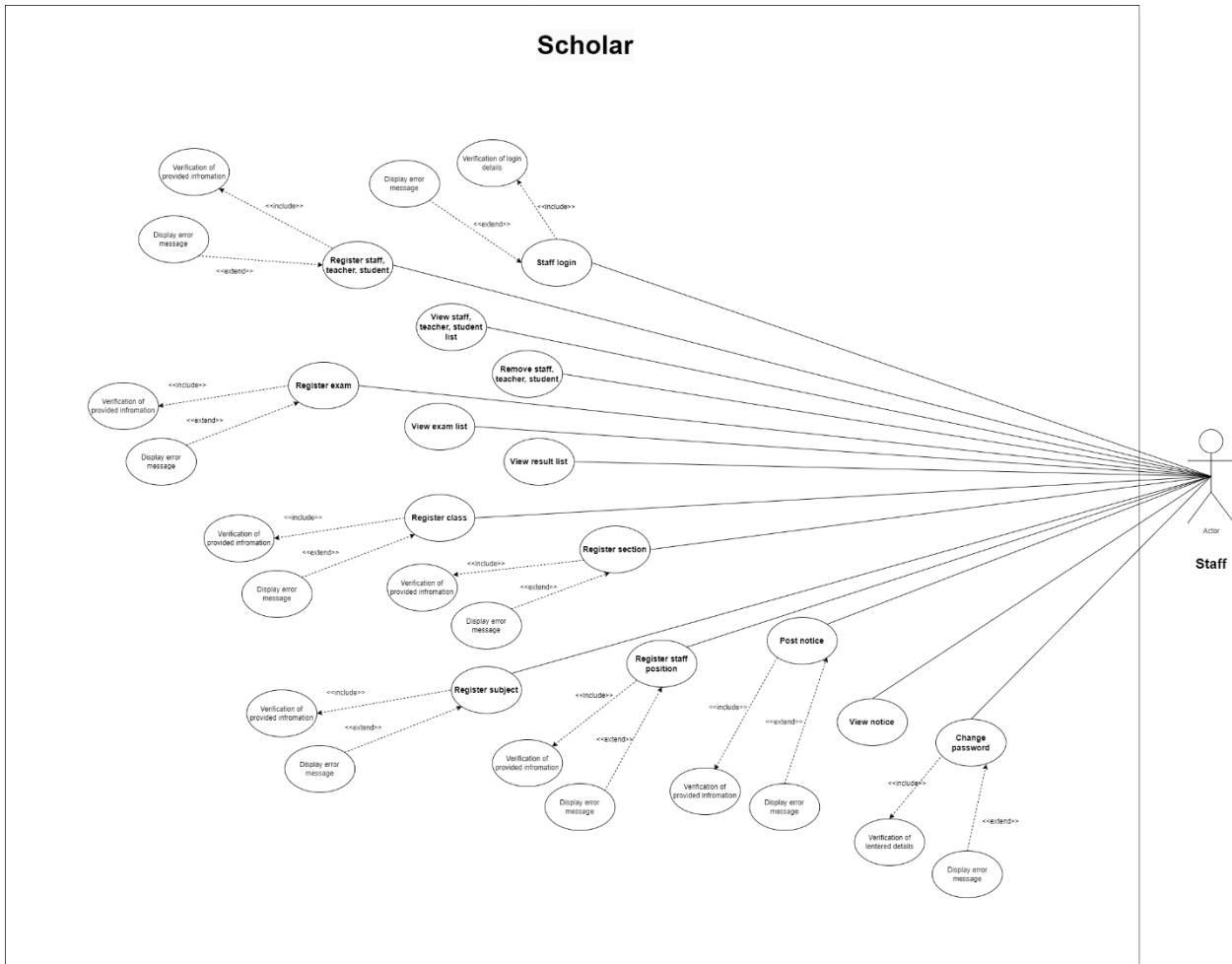


Figure 6 Staff (Use Case diagram)

3.5.2.2. Staff Login:

3.5.2.2.1. High level use case:

Use Case:	Staff Login
Actors:	Staff
Description:	After the registration of the staff, he/she is provided with the login username and password through email. The staff can log in to the system using the correct username and password.

Table 2 Staff Login (High level use case)

3.5.2.2.2. Wireframe:

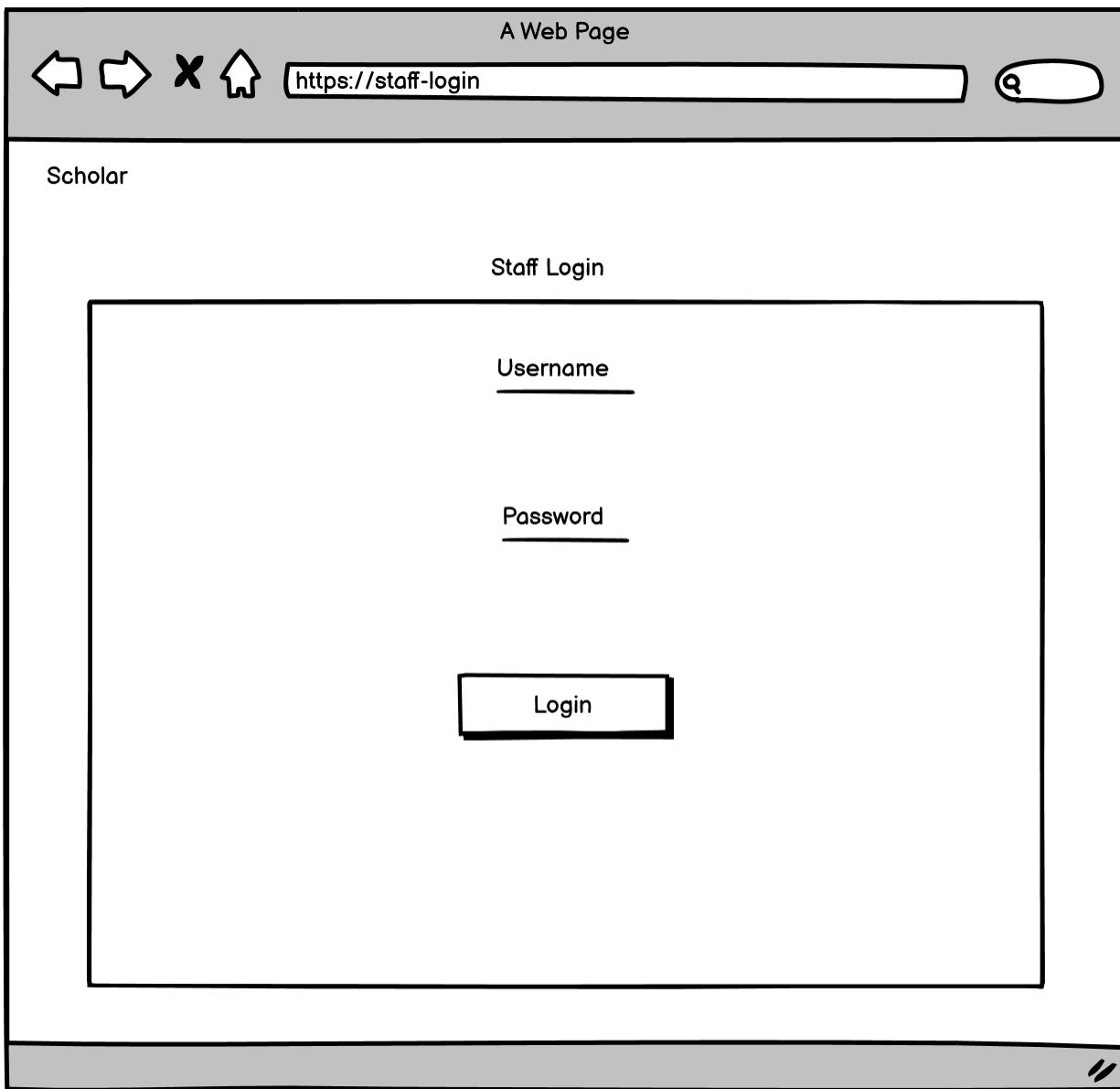


Figure 7 Staff Login (Wireframe)

3.5.2.2.3. Prototype Design:

Scholar

Staff Login

Username _____

Password _____
•••••••• 

LOGIN

Figure 8 Staff Login (Prototype Design)

3.5.2.2.4. Activity Diagram:

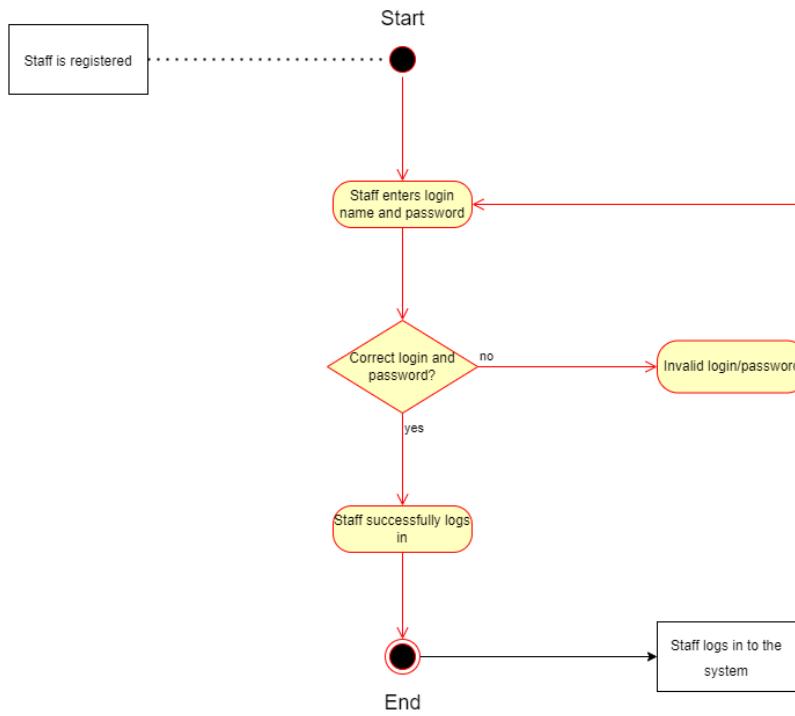


Figure 9 Staff Login (Activity Diagram)

3.5.2.2.5. Sequence Diagram:

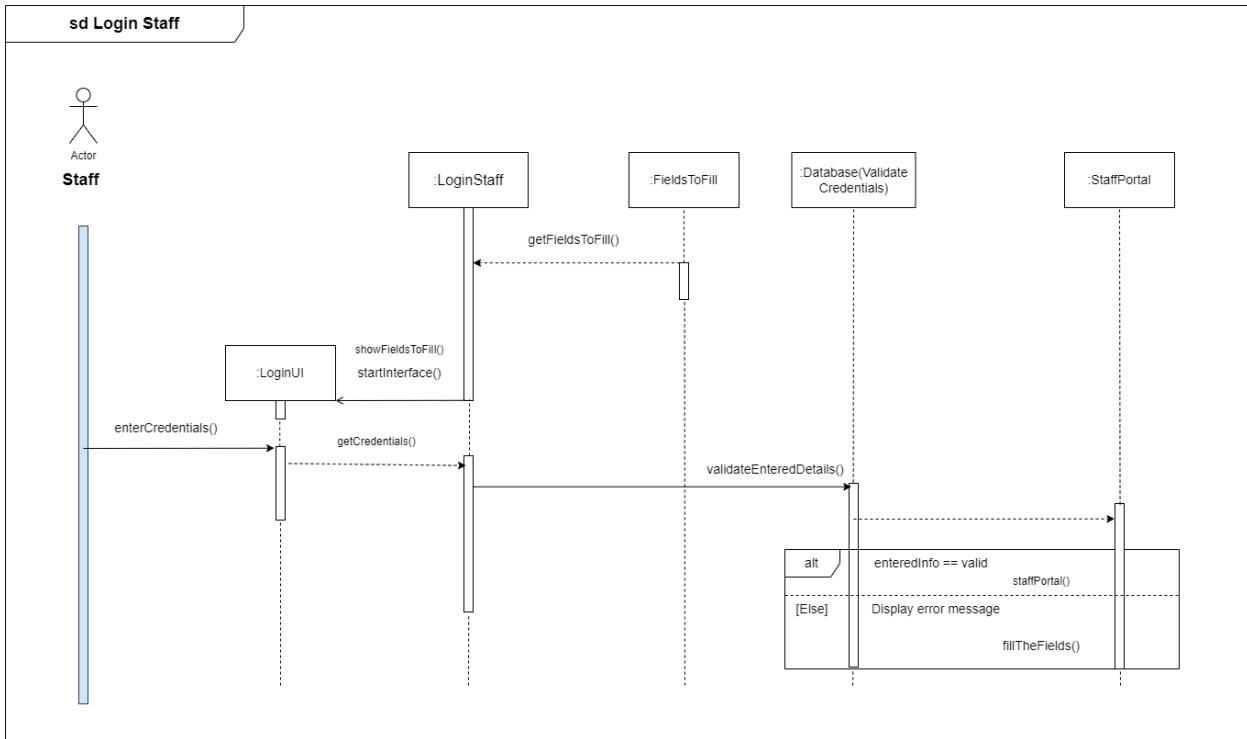


Figure 10 Staff Login (Sequence Diagram)

3.5.2.3. Register staff, teacher, student:

3.5.2.3.1. High level use case:

Use Case:	Register student/teacher
Actors:	Staff
Description:	A student or a teacher provides personal details to the institute itself and staff registers the student or a teacher respectively to become an online member of the system.

Table 3 Register student/teacher (High level use case)

Use Case:	Register staff
Actors:	Staff
Description:	A staff provides personal details to the institute itself and staff registers the new staff to become an online member of the system.

Table 4 Register staff (High level use case)

3.5.2.3.2. Wireframe:

The wireframe depicts a web browser window titled "A Web Page". The address bar shows the URL <https://staff-registration>. Below the browser header, the page content is titled "Scholar" and "Staff Registration". The main form area contains fields for "First Name", "Last Name", "Username", "Email", "Password", "Password Confirmation", a dropdown menu labeled "Student" with a downward arrow, and another dropdown menu labeled "Position" with a downward arrow. At the bottom left of the form is a "Register" button.

A "Search" icon is located in the top right corner of the browser header. A double-line symbol is positioned at the bottom right of the page content area.

Figure 11 Staff registration (Wireframe)

A Web Page

https://teacher-registration

Scholar

Teacher Registration

First Name _____

Last Name _____

Username _____

Email _____

Password _____

Password Confirmation _____

Student ▾

Subjects ▾

Class ▾

Register

Figure 12 Teacher registration (Wireframe)

A Web Page

https://student-registration

Scholar

Student Registration

First Name _____

Last Name _____

Username _____

Email _____

Password _____

Password Confirmation _____

Student ▾

Subjects ▾

Class ▾

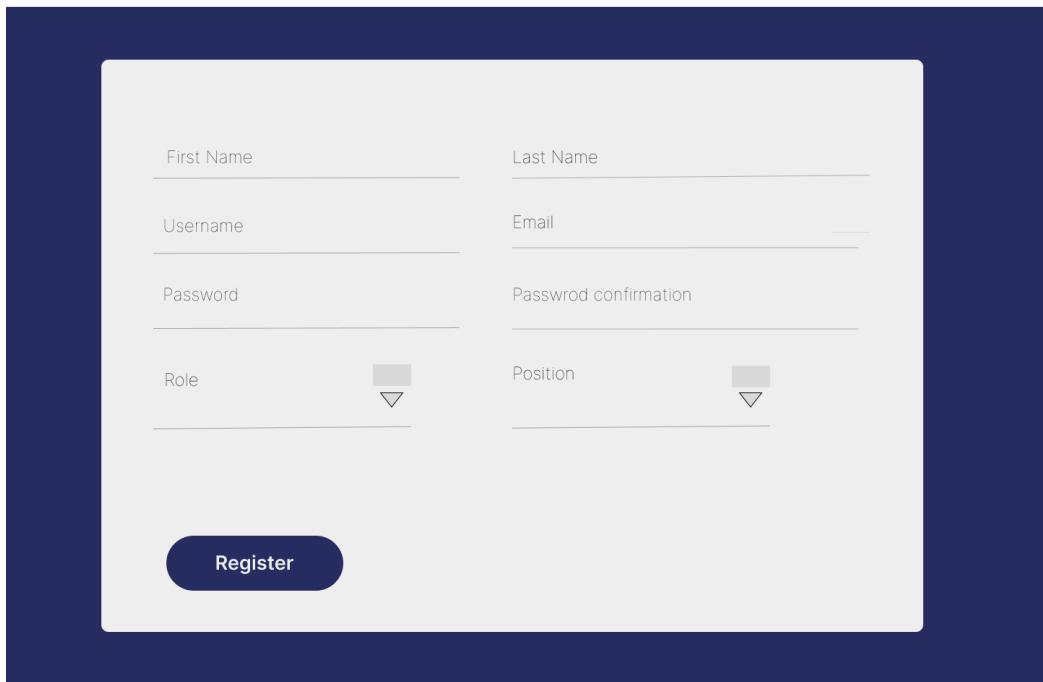
Register

Figure 13 Student registration (Wireframe)

3.5.2.3.3. Prototype:

Scholar

Staff Registration



The image shows a prototype of a staff registration form. The form is contained within a white rectangular box with rounded corners, set against a dark blue background. The fields are arranged in two columns. The left column contains 'First Name' and 'Username' inputs, followed by 'Password' and 'Role' inputs. The right column contains 'Last Name' and 'Email' inputs, followed by 'Passwrod confirmation' and 'Position' inputs. Each input field has a small downward arrow icon to its right. A large blue 'Register' button is located at the bottom left of the form area.

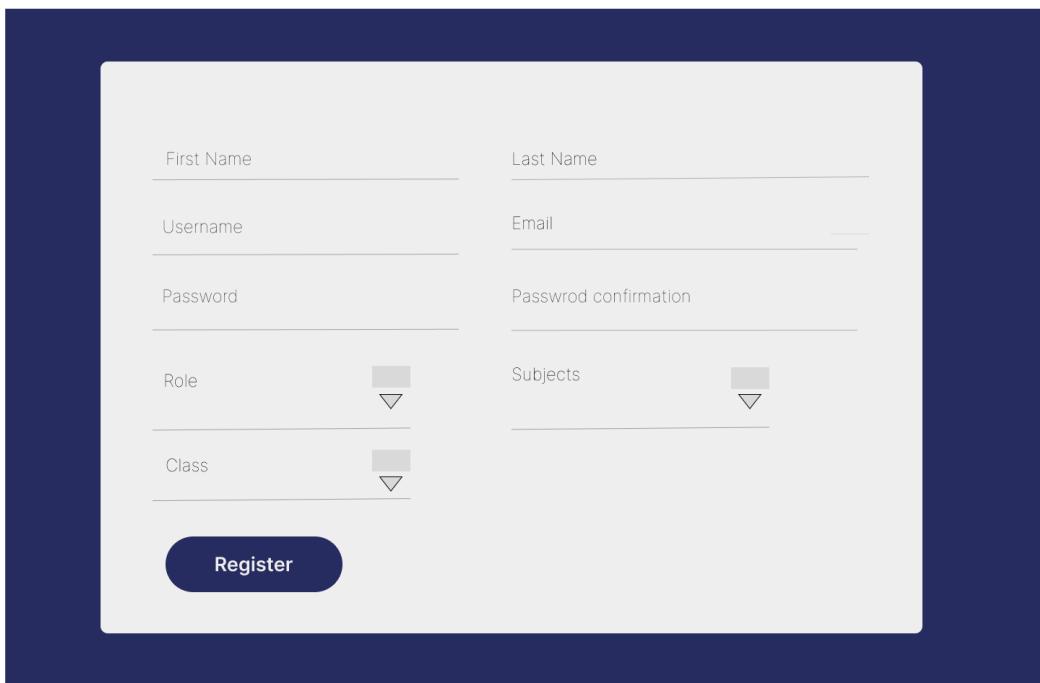
First Name	Last Name
Username	Email
Password	Passwrod confirmation
Role	Position

Register

Figure 14 Staff registration (Prototype)

Scholar

Teacher Registration



The image shows a prototype of a teacher registration form. It is contained within a white rectangular box with rounded corners, set against a dark blue background. The form fields are arranged in two columns. The left column contains 'First Name' and 'Username' input fields, followed by 'Password' and 'Role' dropdown menus. The right column contains 'Last Name' and 'Email' input fields, followed by 'Passwrod confirmation' and 'Subjects' dropdown menus. A large blue 'Register' button is positioned at the bottom center of the form.

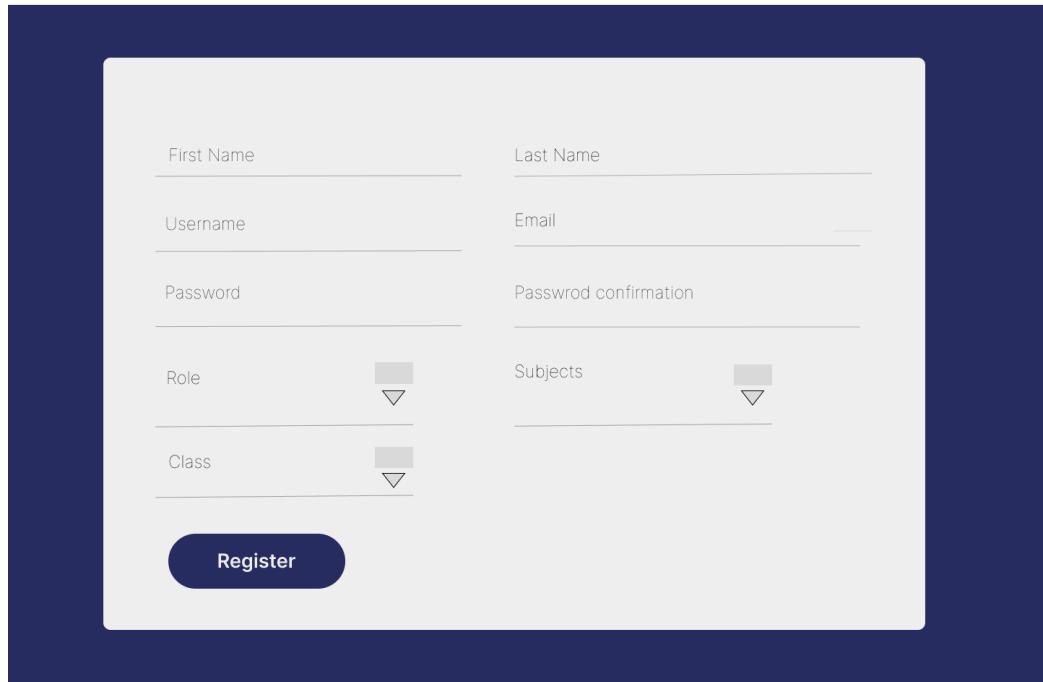
First Name	Last Name
Username	Email
Password	Passwrod confirmation
Role	Subjects
Class	

Register

Figure 15 Teacher registration (Prototype)

Scholar

Student Registration



A wireframe prototype of a student registration form. The form is contained within a white rectangular box with rounded corners, set against a dark blue background. The fields are arranged in two columns. The left column contains 'First Name' and 'Username' inputs, followed by 'Password' and 'Role' dropdown menus. The right column contains 'Last Name' and 'Email' inputs, followed by 'Passwrod confirmation' and 'Subjects' dropdown menus. A large 'Register' button is positioned at the bottom center of the form.

First Name	Last Name
Username	Email
Password	Passwrod confirmation
Role	Subjects
Class	

Register

Figure 16 Student registration (Prototype)

3.5.2.3.4. Activity Diagram:

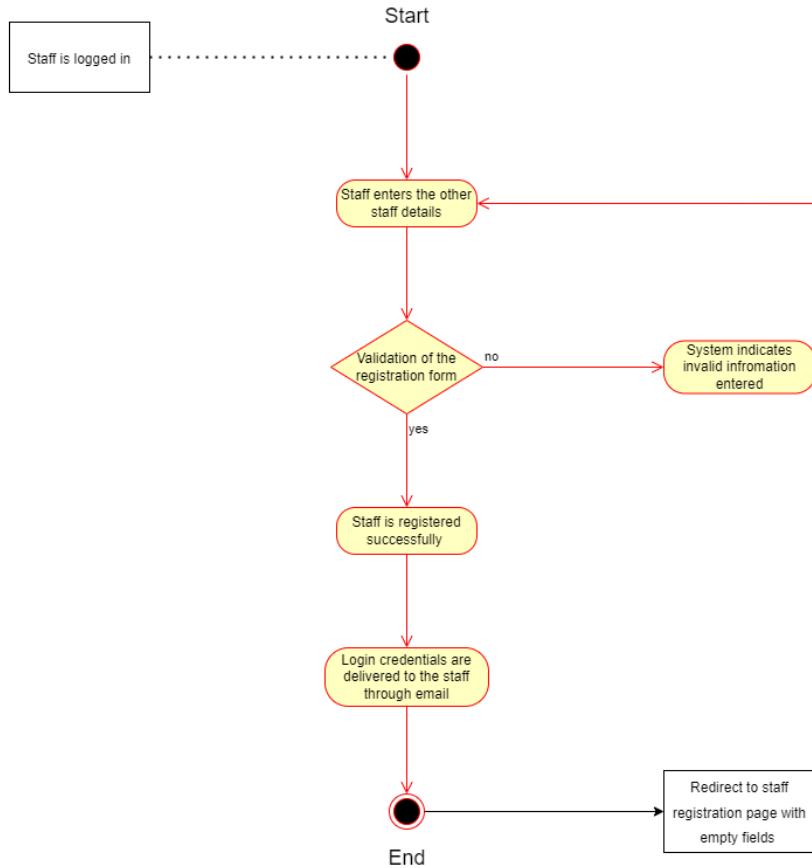


Figure 17 Staff registration (Activity Diagram)

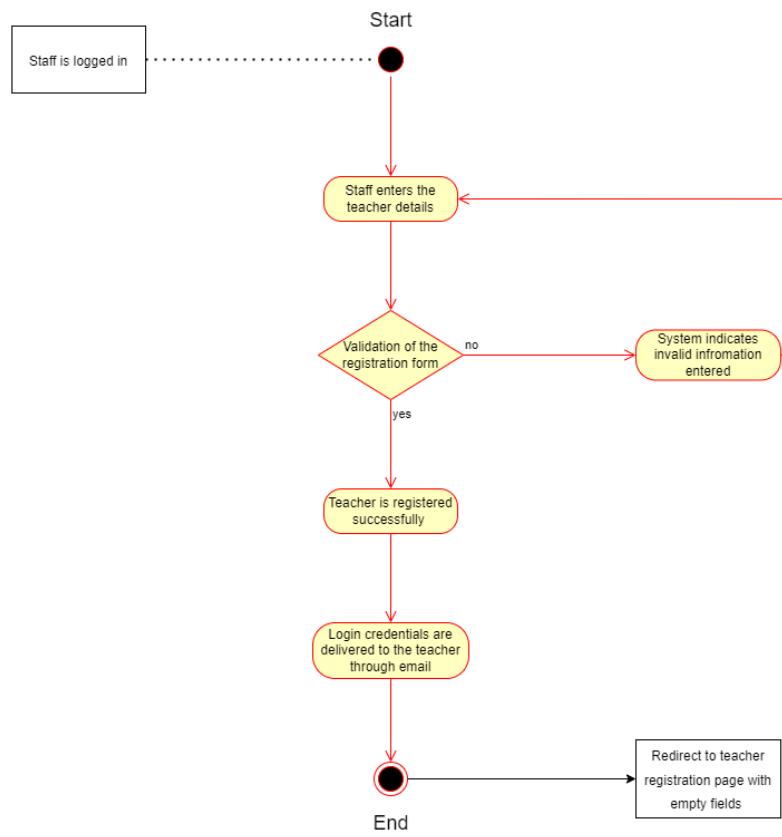


Figure 18 Teacher registration (Activity Diagram)

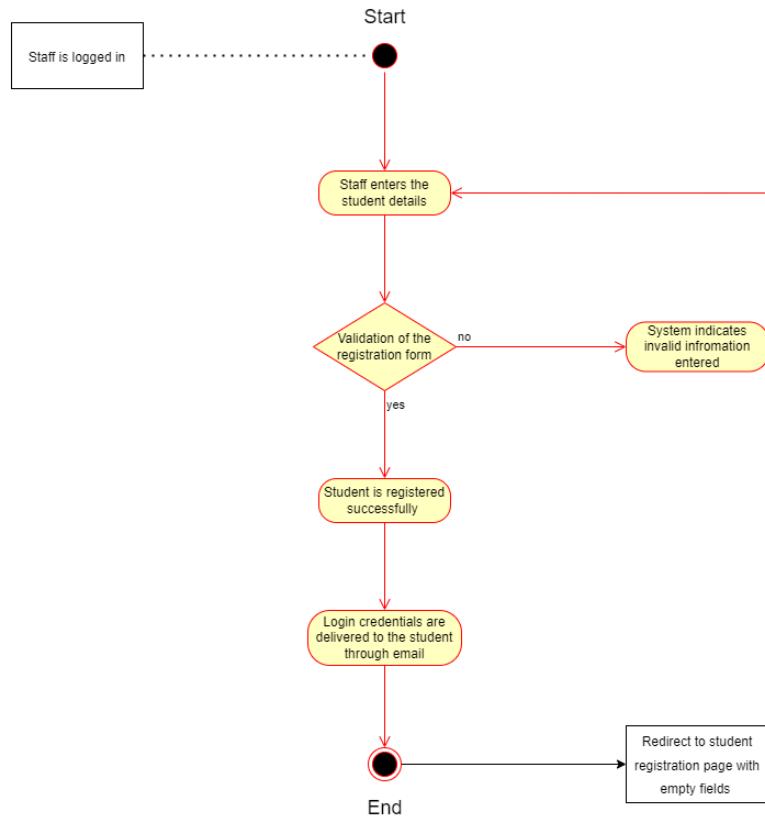


Figure 19 Student registration (Activity Diagram)

3.5.2.3.5. Sequence Diagram:

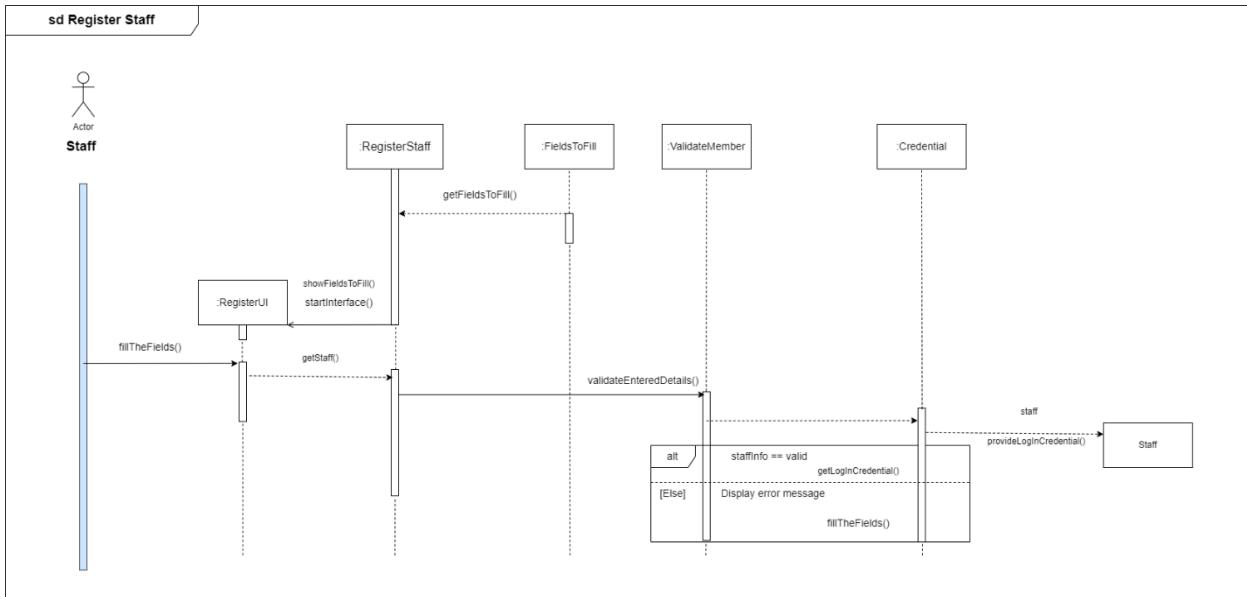


Figure 20 Staff registration (Sequence Diagram)

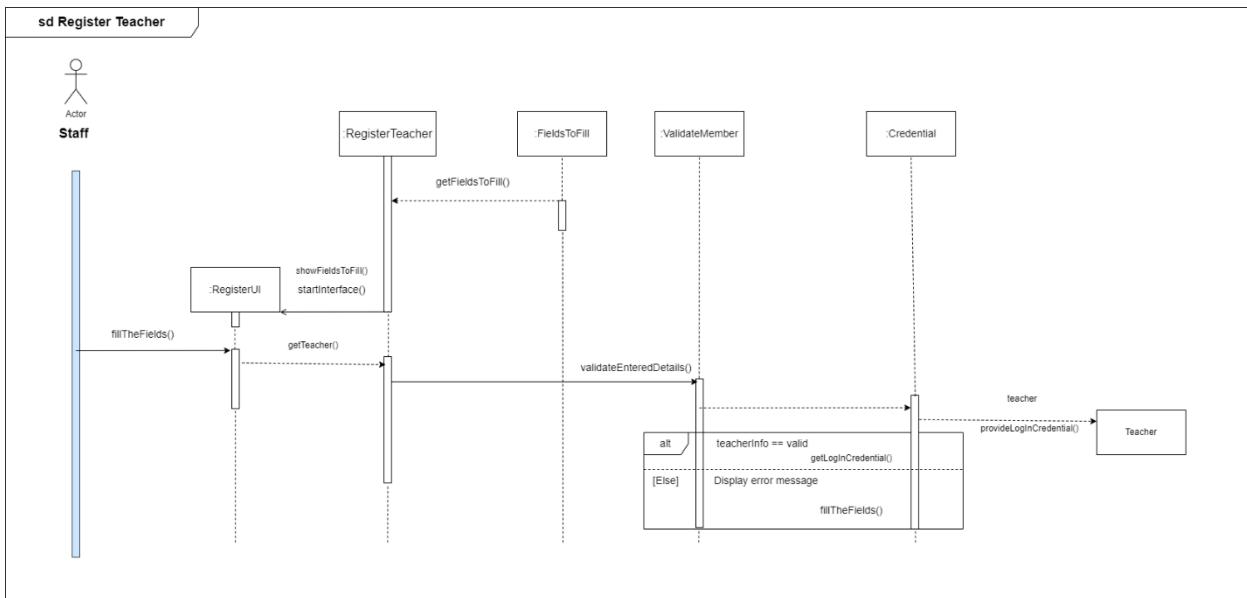


Figure 21 Teacher registration (Sequence Diagram)

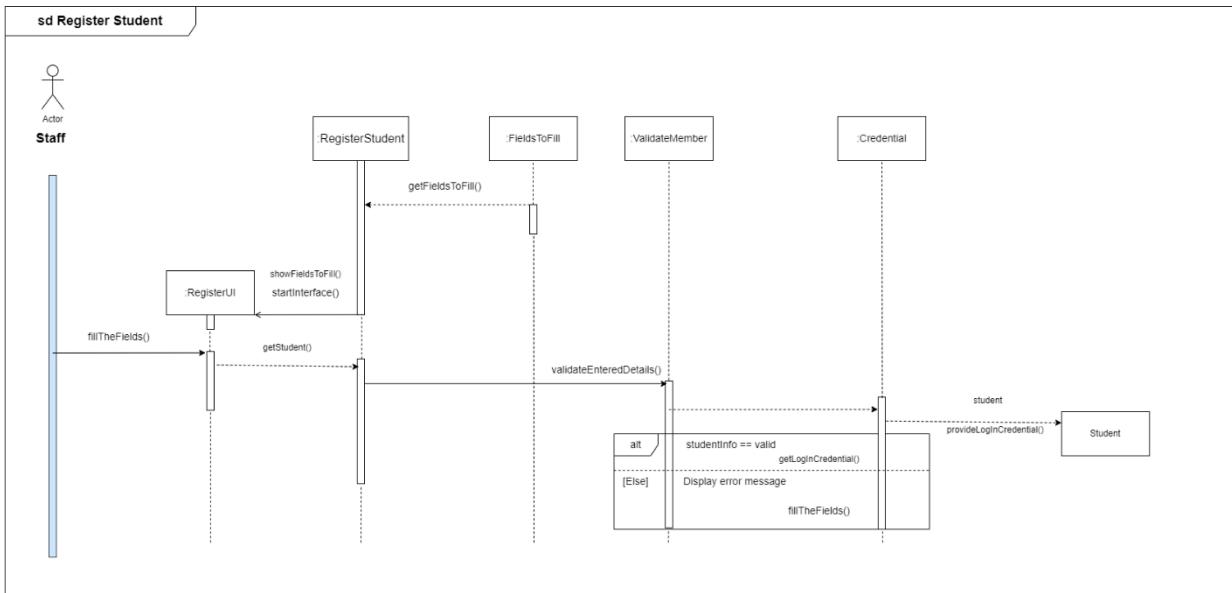


Figure 22 Student registration (Sequence Diagram)

3.5.2.4. View staff, teacher, student list:

3.5.2.4.1. High level use case:

Use Case:	View staff, teacher, student list
Actors:	Staff
Description:	Staff clicks on the Account View option from navigation bar. Then the staff clicks on the Staff, Teacher or Student option respectively to view the user list.

Table 5 View staff, teacher, student list (High level use case)

3.5.2.4.2. Activity Diagram:

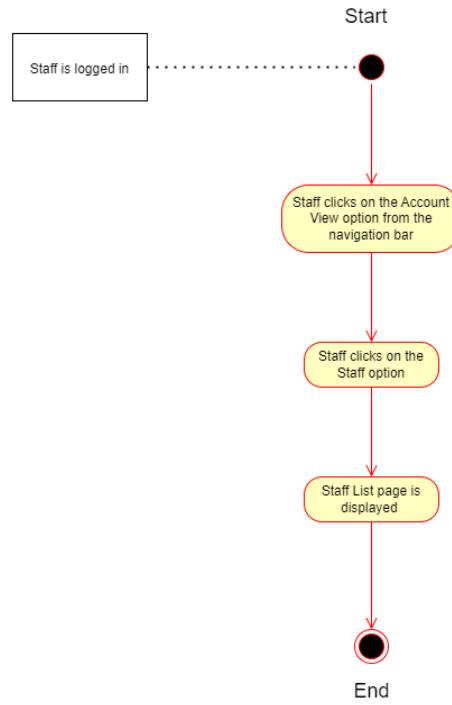


Figure 23 View Staff List (Activity Diagram)

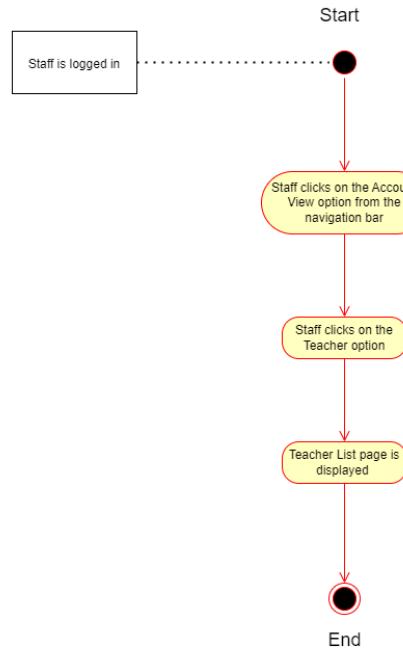


Figure 24 View Teacher List (Activity Diagram)

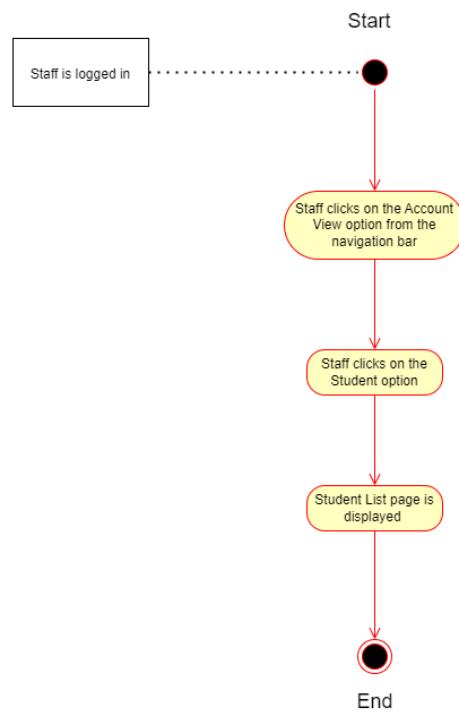


Figure 25 View Student List (Activity Diagram)

3.5.2.4.3. Sequence Diagram:

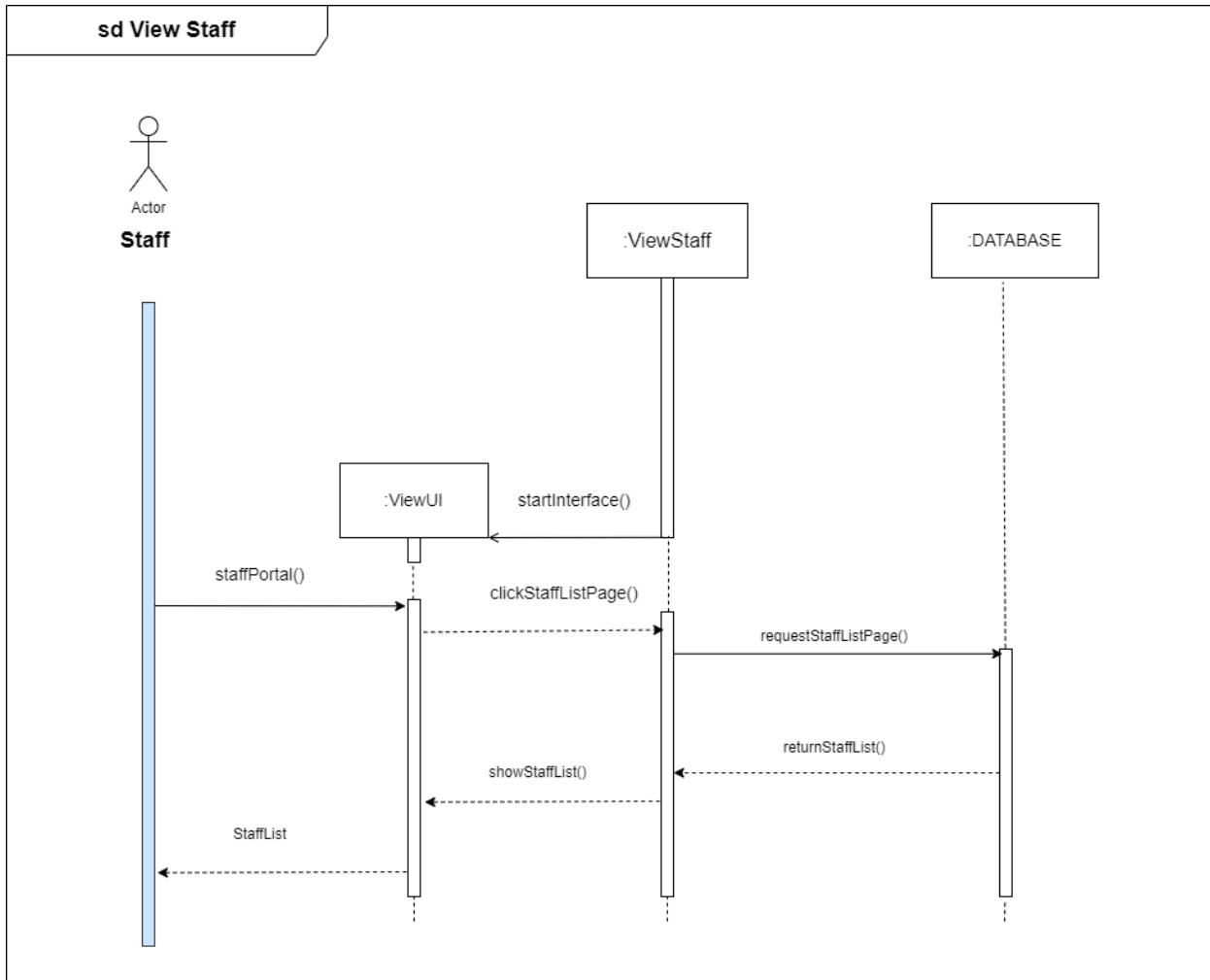


Figure 26 View Staff List (Sequence Diagram)

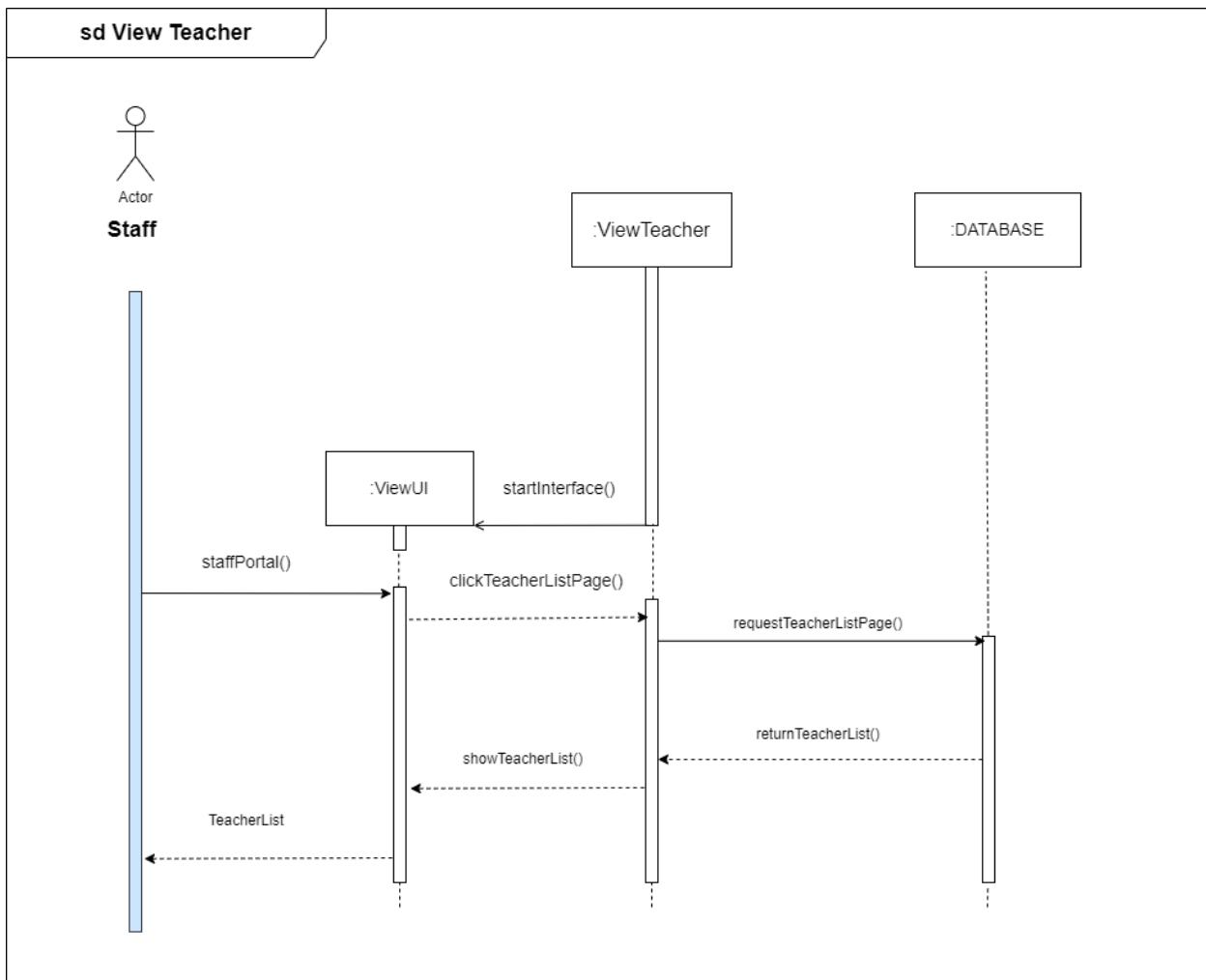


Figure 27 View Teacher List (Sequence Diagram)

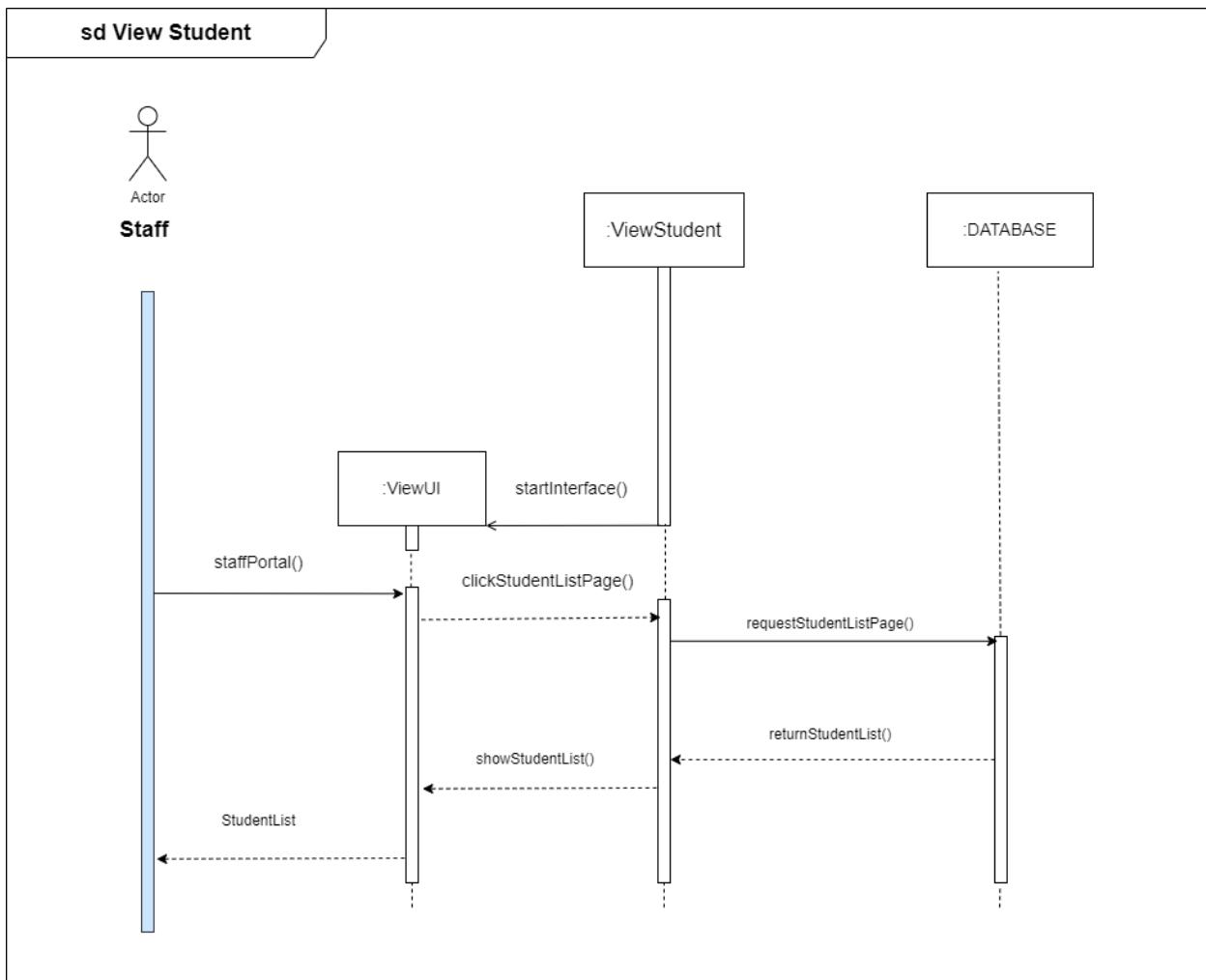


Figure 28 View Student List (Sequence Diagram)

3.5.2.5. Remove staff, teacher, student:

3.5.2.5.1. High level use case:

Use Case:	Remove staff, teacher, student
Actors:	Staff
Description:	Staff clicks on the Account View option from navigation bar. Then the staff clicks on the Staff, Teacher or Student option respectively to view the user list. Staff selects delete option for the user to be deleted.

Table 6 Remove staff, teacher, student (High level use case)

3.5.2.5.2. Activity Diagram:

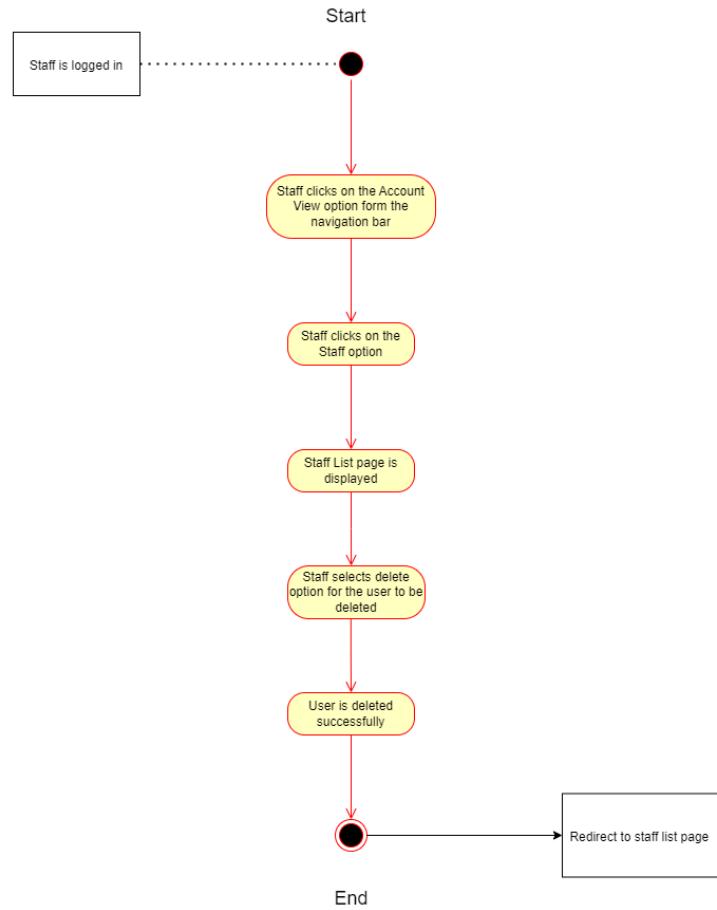


Figure 29 Remove Staff (Activity Diagram)

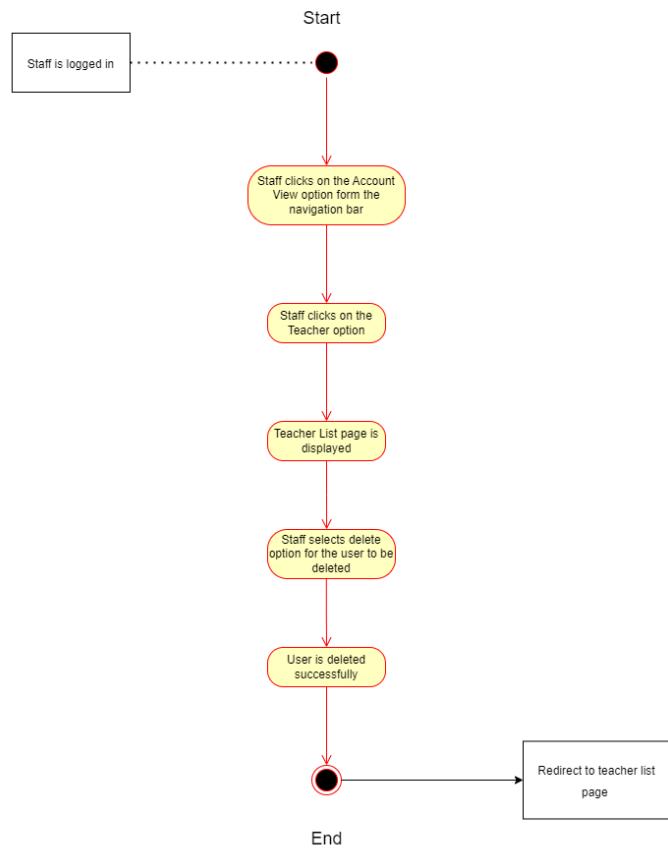


Figure 30 Remove Teacher (Activity Diagram)

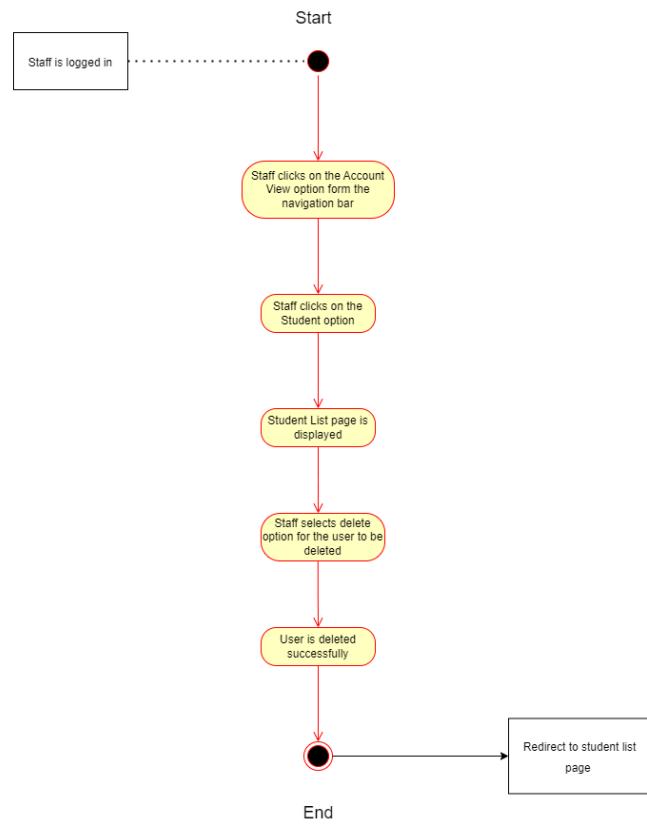


Figure 31 Remove Student (Activity Diagram)

3.5.2.5.3. Sequence Diagram:

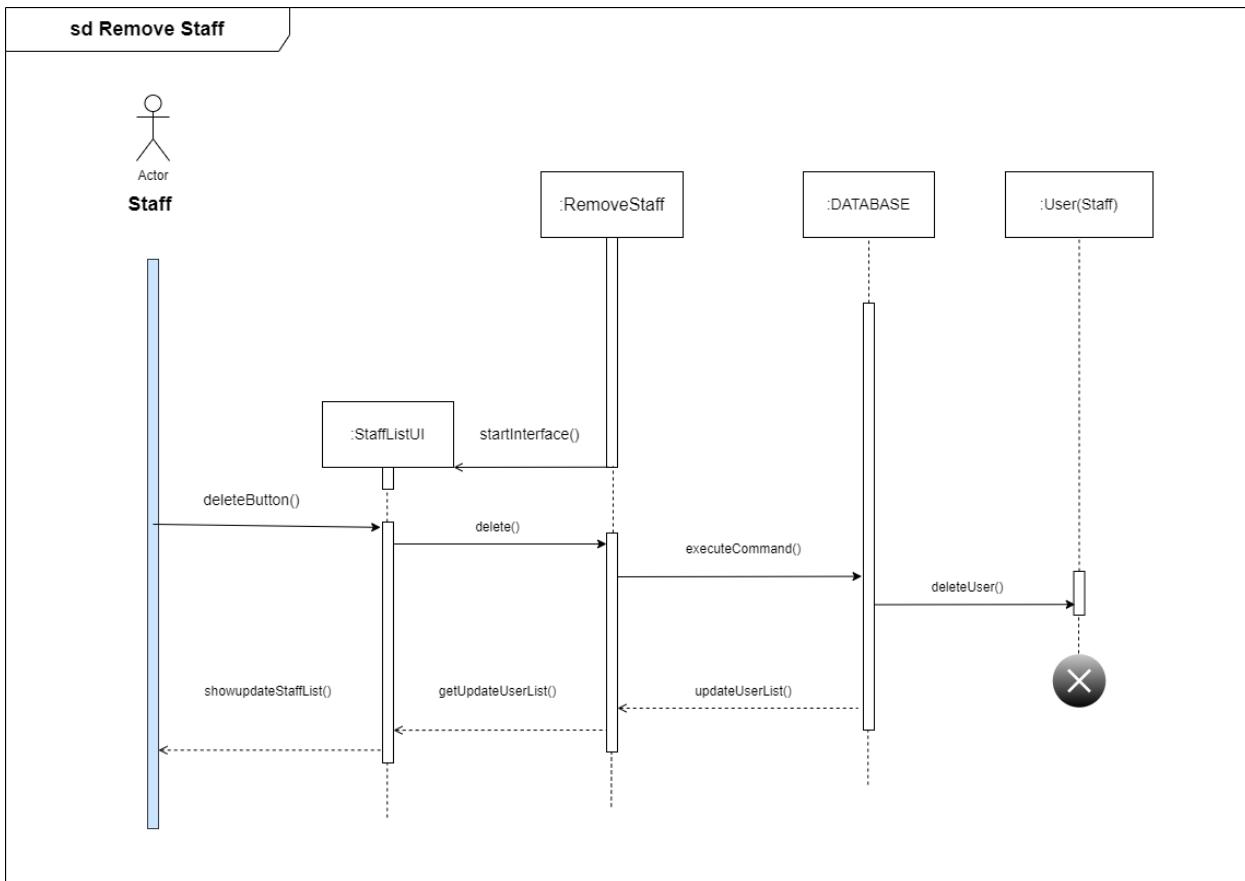


Figure 32 Remove Staff (Sequence Diagram)

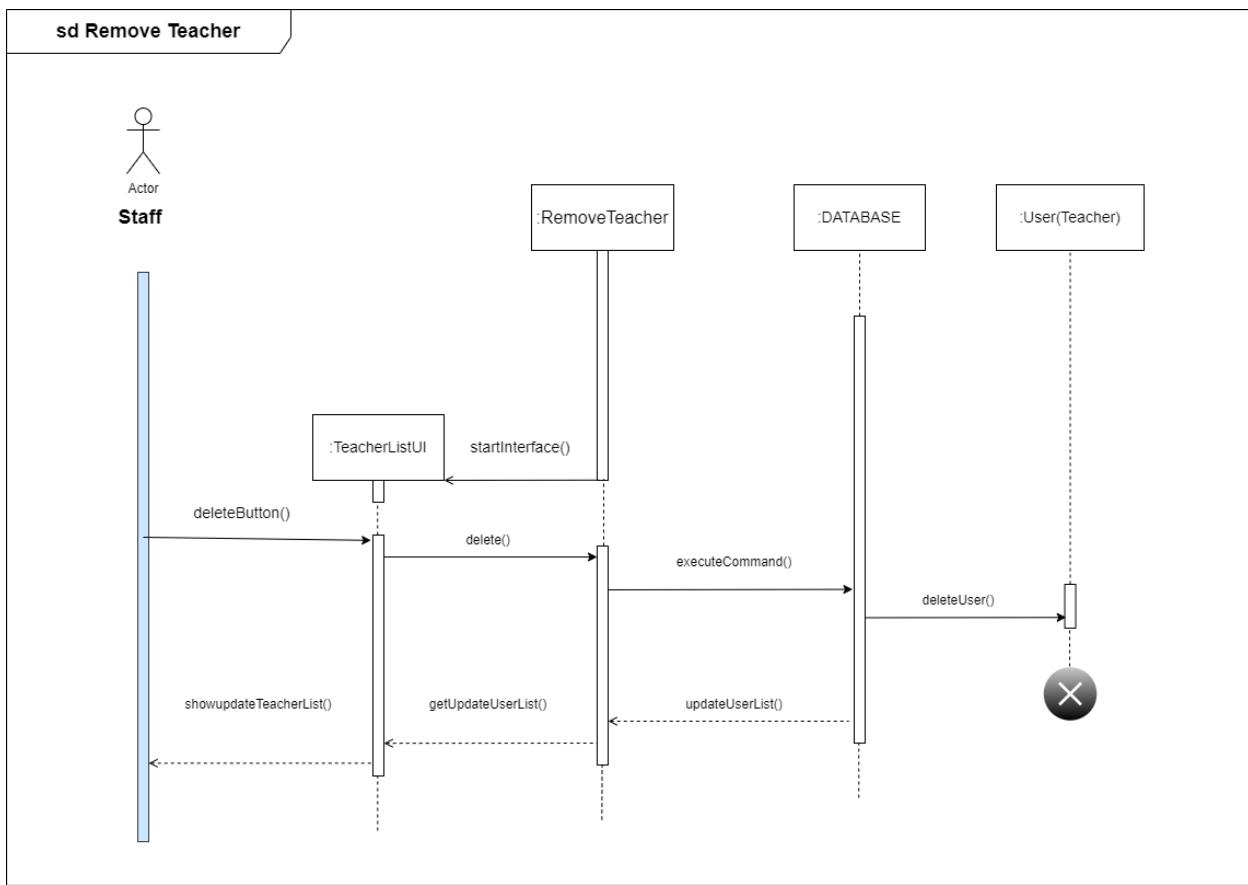


Figure 33 Remove Teacher (Sequence Diagram)

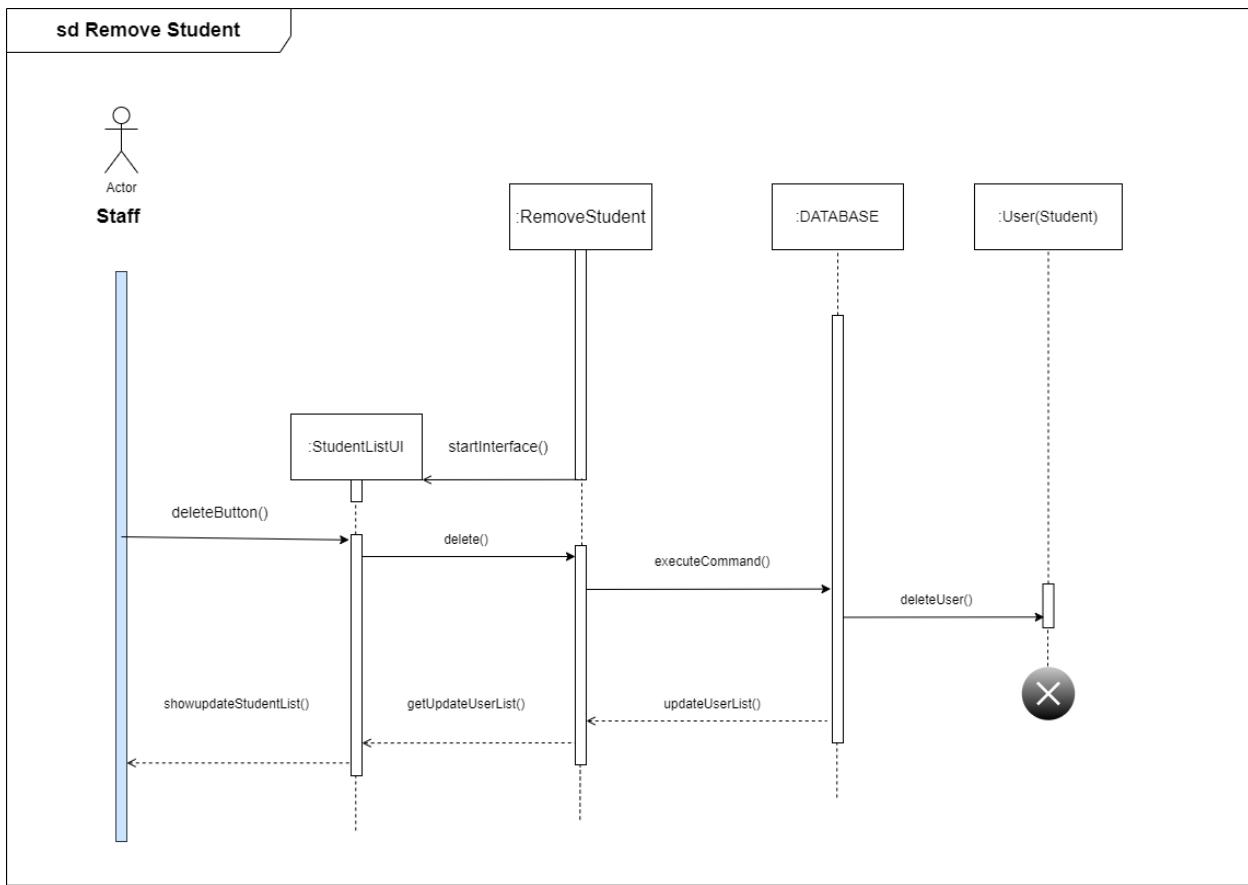


Figure 34 Remove Student (Sequence Diagram)

3.5.2.6. Register Exam (By Staff):

3.5.2.6.1. High level use case:

Use Case:	Register Exam (By staff)
Actors:	Staff
Description:	Staff enters exam details in the exam registration form. The exam gets registered after the validation of the entered details.

Table 7 Register Exam (By Staff) (High level use case)

3.5.2.6.3. Activity Diagram:

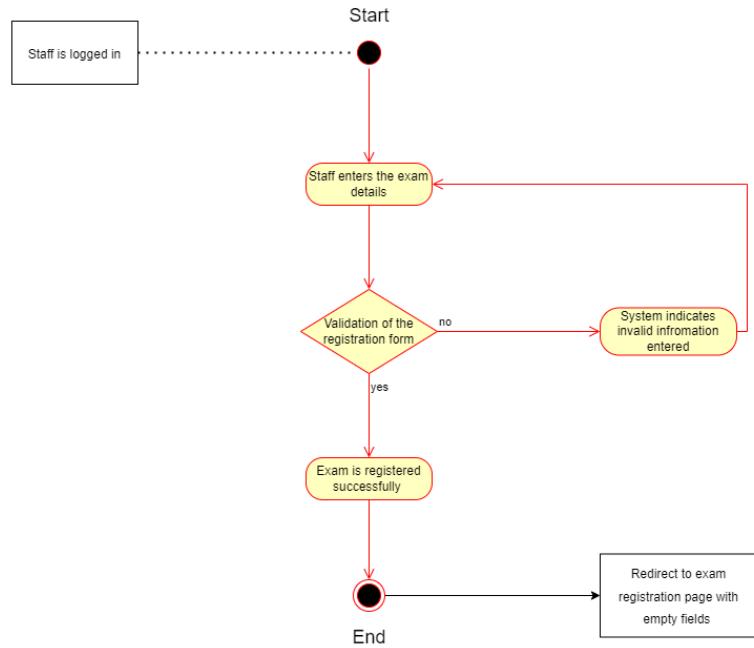


Figure 35 Register Exam (By Staff) (Activity Diagram)

3.5.2.6.3. Sequence Diagram:

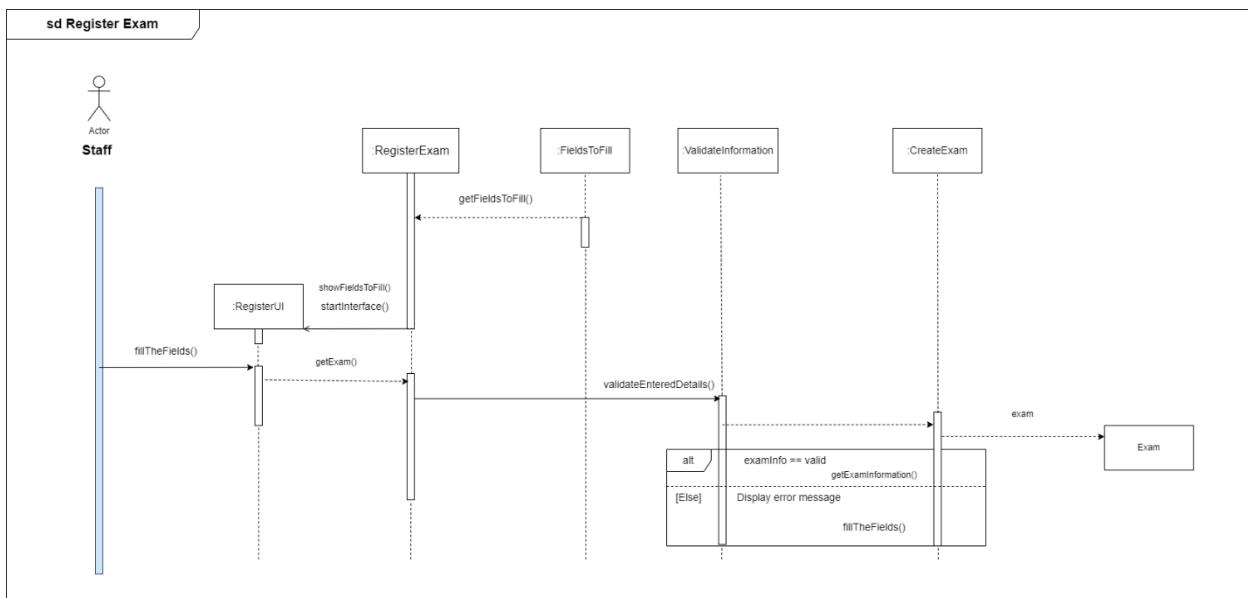


Figure 36 Register Exam (By Staff) (Sequence Diagram)

3.5.2.7. View Exam List:

3.5.2.7.1. High level use case:

Use Case:	View exam list
Actors:	Staff
Description:	Staff clicks on the Exam Material option from navigation bar. Then the staff clicks on the Exam List option to view the exam list.

Table 8 View Exam List (High level use case)

3.5.2.7.2. Activity Diagram:

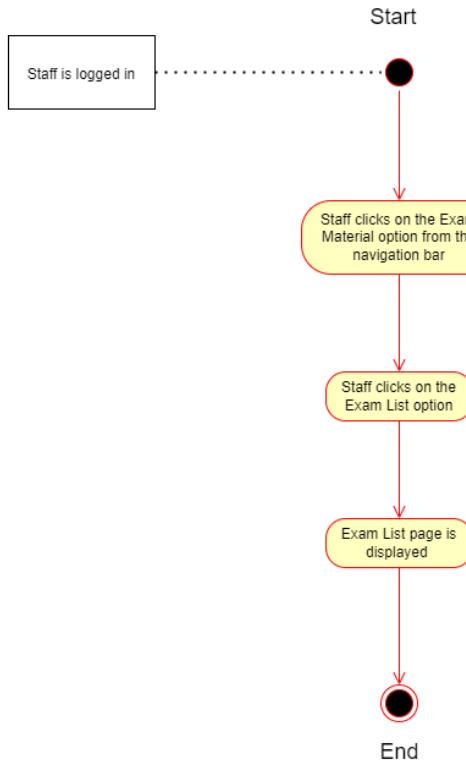


Figure 37 View Exam List (Activity Diagram)

3.5.2.7.3. Sequence Diagram:

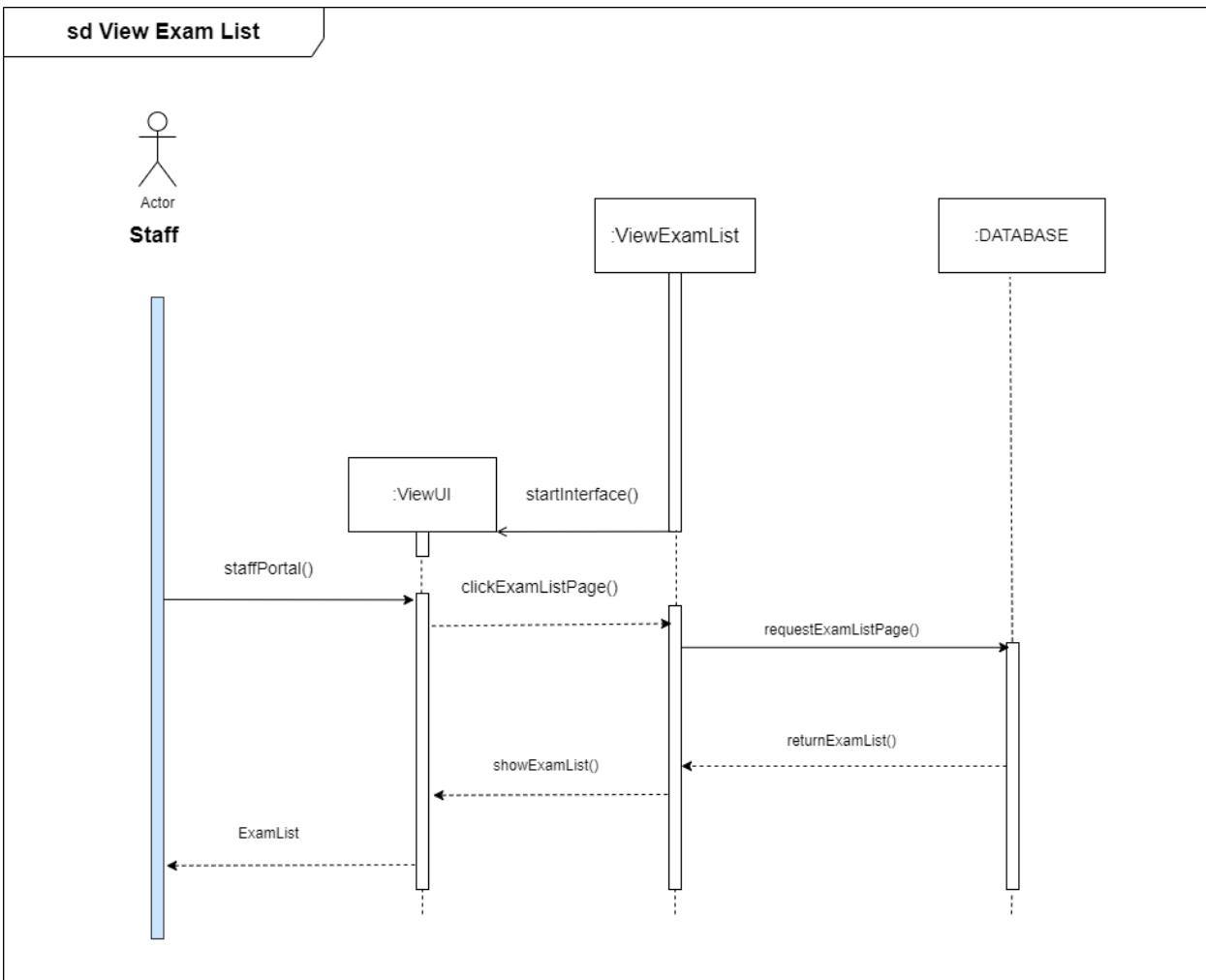


Figure 38 View Exam List (Sequence Diagram)

3.5.2.8. Register Class:

3.5.2.8.1. High level use case:

Use Case:	Register Class
Actors:	Staff
Description:	Staff enters class details in the class registration form. The class gets registered after the validation of the entered details.

Table 9 Register Class (High level use case)

3.4.2.8.2. Activity Diagram:

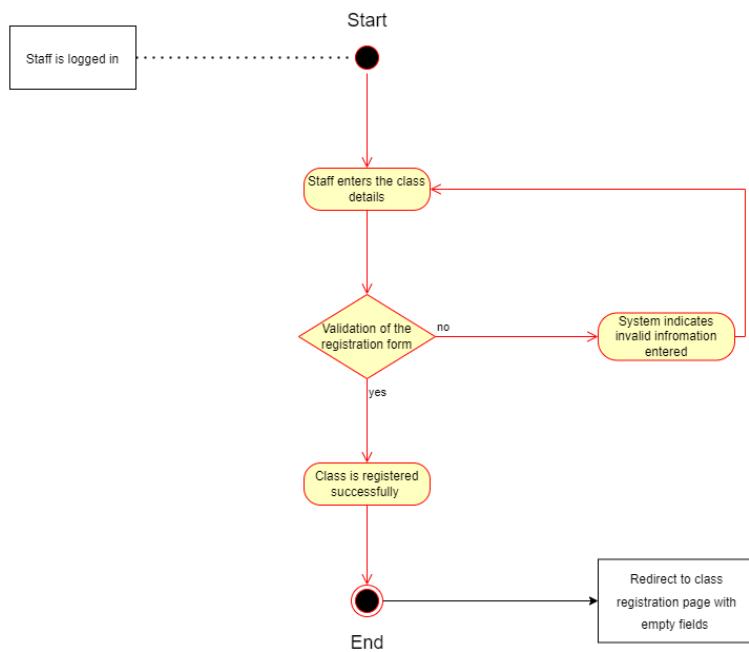


Figure 39 Register Class (Activity Diagram)

3.5.2.8.3. Sequence Diagram:

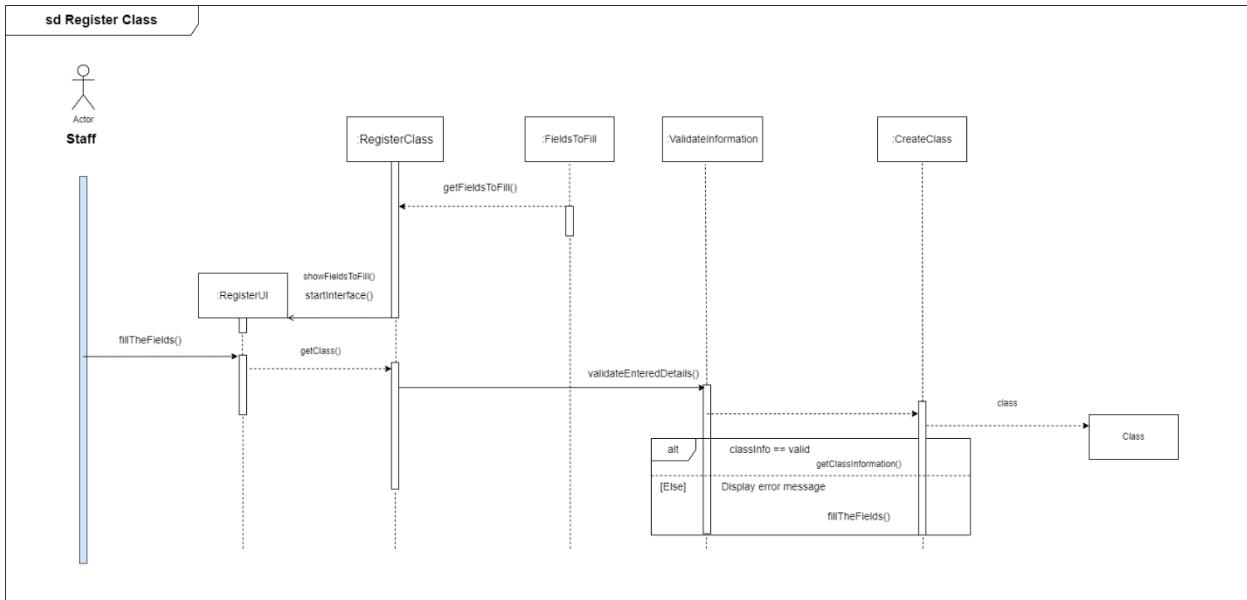


Figure 40 Register Class (Sequence Diagram)

3.5.2.9. Register Section:

3.5.2.9.1. High level use case:

Use Case:	Register Section
Actors:	Staff
Description:	Staff enters section details in the section registration form. The section gets registered after the validation of the entered details.

Table 10 Register Section (High level use case)

3.5.2.9.2. Activity Diagram:

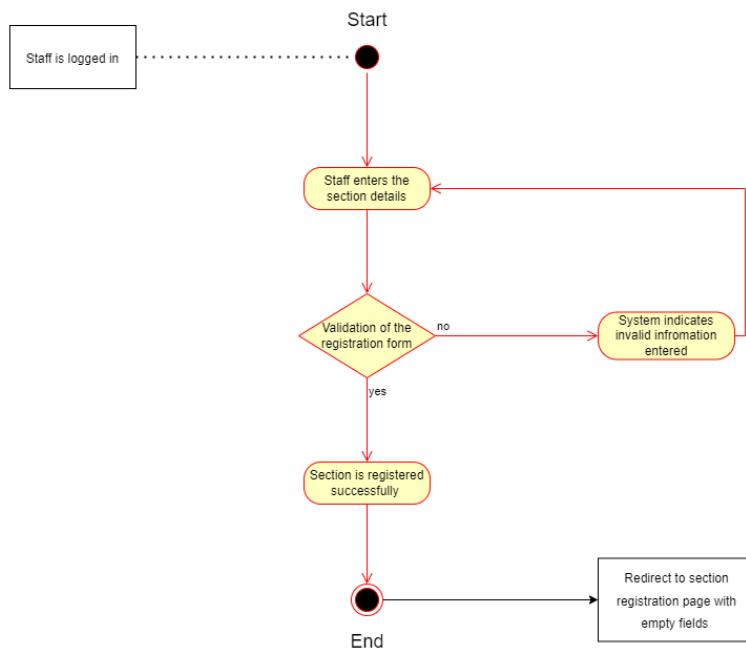


Figure 41 Register Section (Activity Diagram)

3.5.2.9.3. Sequence Diagram:

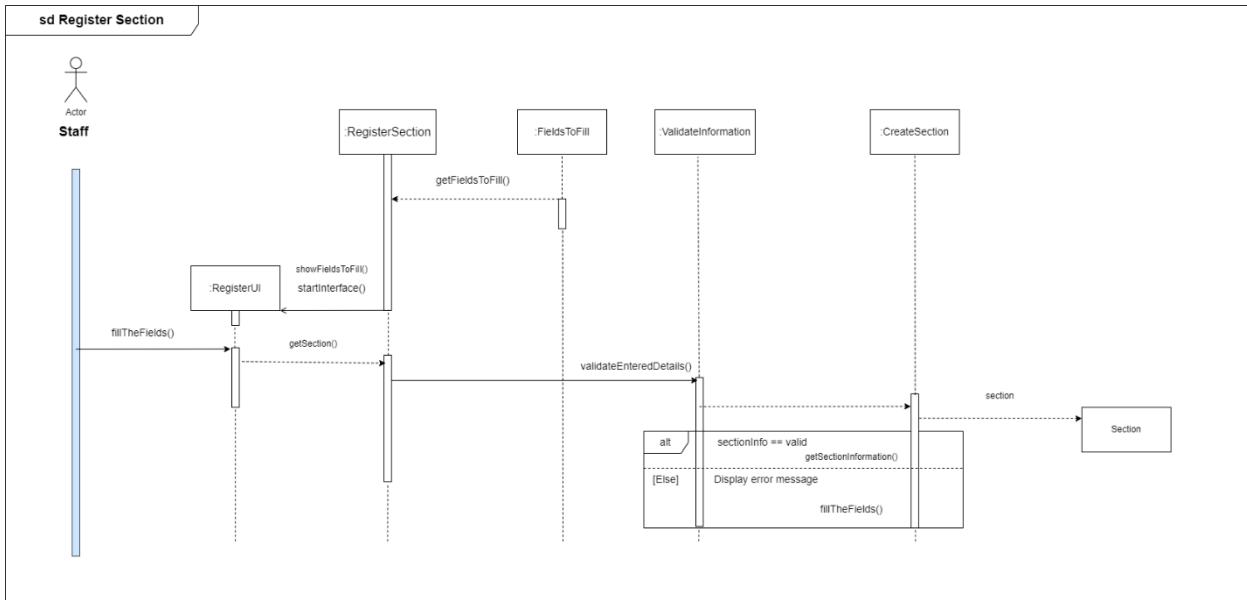


Figure 42 Register Section (Sequence Diagram)

3.5.2.10. Register Subject:

3.5.2.10.1. Wireframe:

Use Case:	Register Subject
Actors:	Staff
Description:	Staff enters subject details in the subject registration form. The subject gets registered after the validation of the entered details.

Table 11 Register Subject (High level use case)

3.5.2.10.2. Activity Diagram:

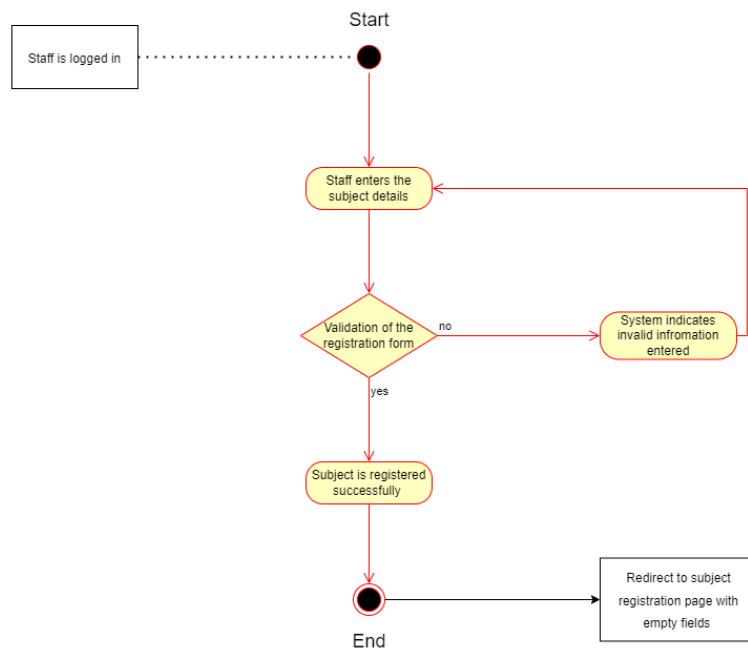


Figure 43 Register subject (Activity Diagram)

3.5.2.10.3. Sequence Diagram:

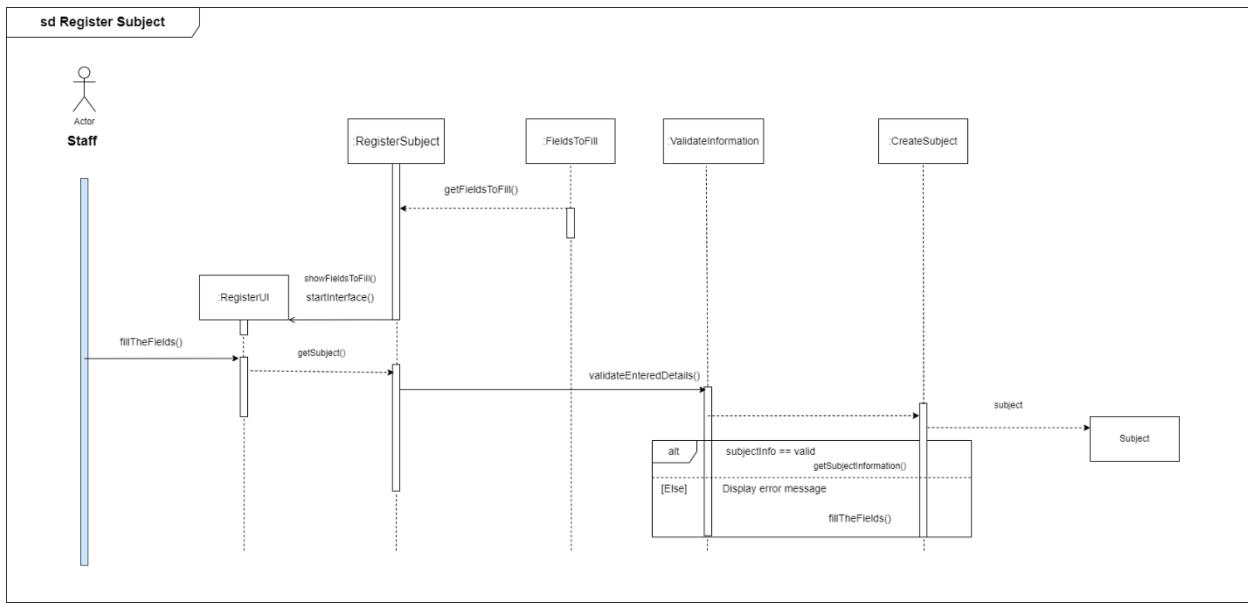


Figure 44 Register Subject (Sequence Diagram)

3.5.2.11. Register Staff Position:

3.5.2.11.1. High level use case:

Use Case:	Register Staff Position
Actors:	Staff
Description:	Staff enters staff position details in the staff position registration form. The staff position gets registered after the validation of the entered details.

Table 12 Register Staff Position (High level use case)

3.5.2.11.2. Activity Diagram:

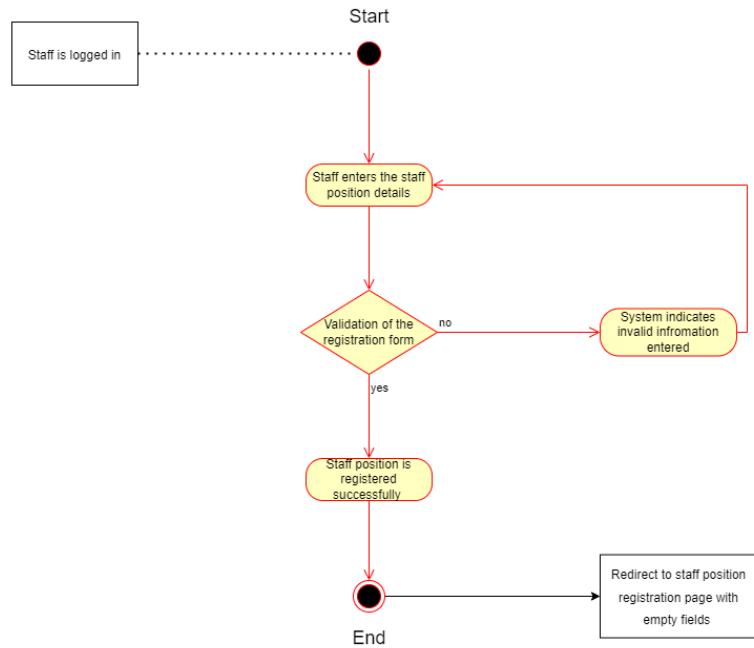


Figure 45 Register Staff Position (Activity Diagram)

3.5.2.11.3. Sequence Diagram:

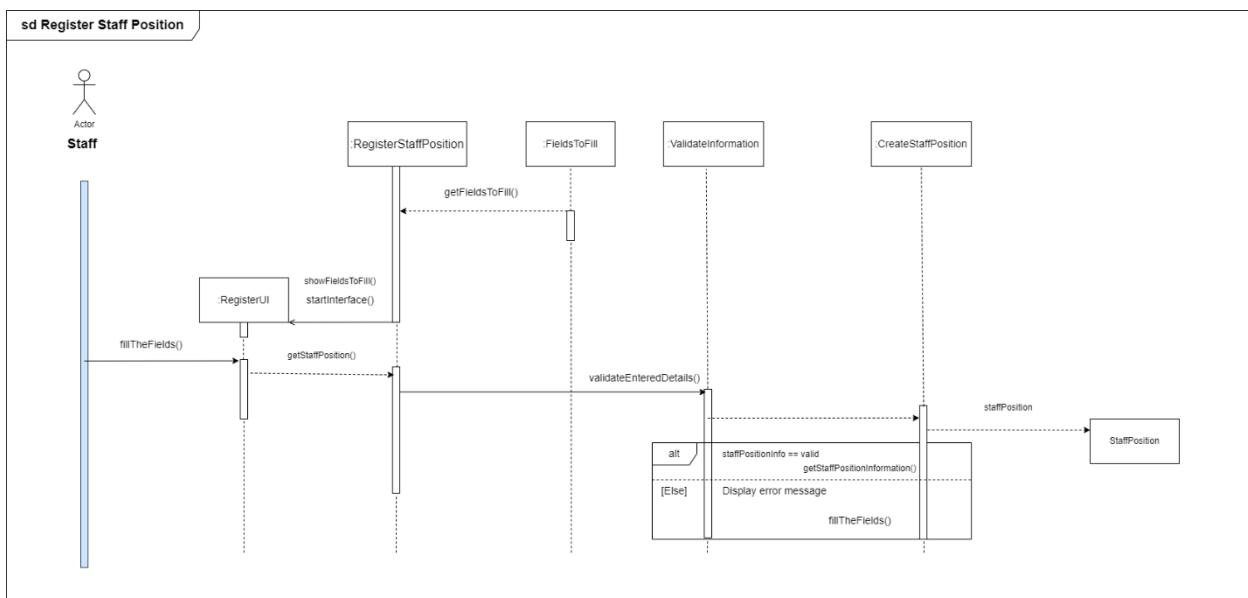


Figure 46 Register Staff Position (Sequence Diagram)

3.5.2.12. Post Notice:

3.5.2.12.1. High level use case:

Use Case:	Post Notice
Actors:	Staff
Description:	Staff enters notice details in the add notice form. The notice gets posted after the validation of the entered details.

Table 13 Post Notice (High level use case)

3.5.2.12.2. Wireframe:

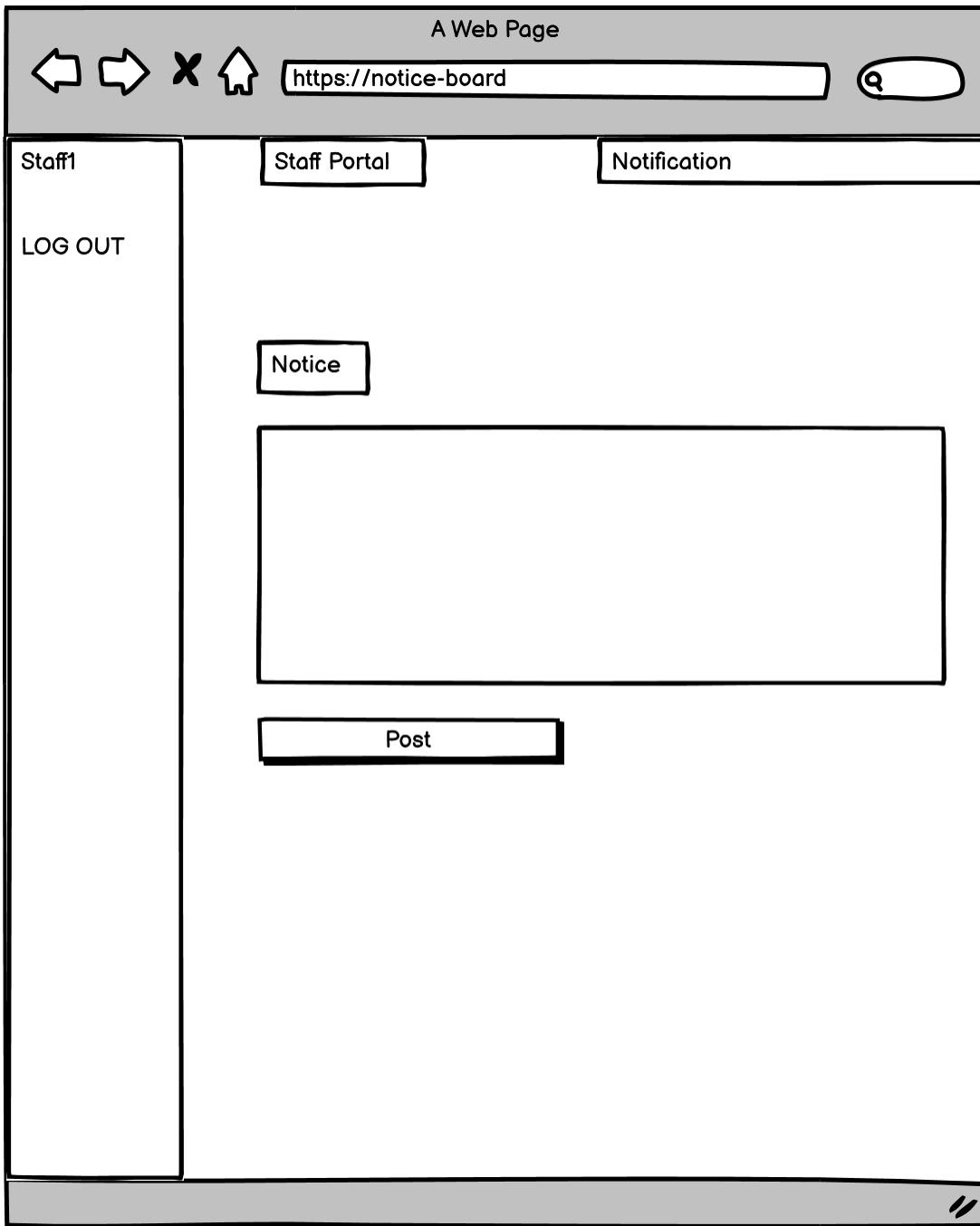


Figure 47 Post Notice (Wireframe)

3.5.2.12.3. Prototype:

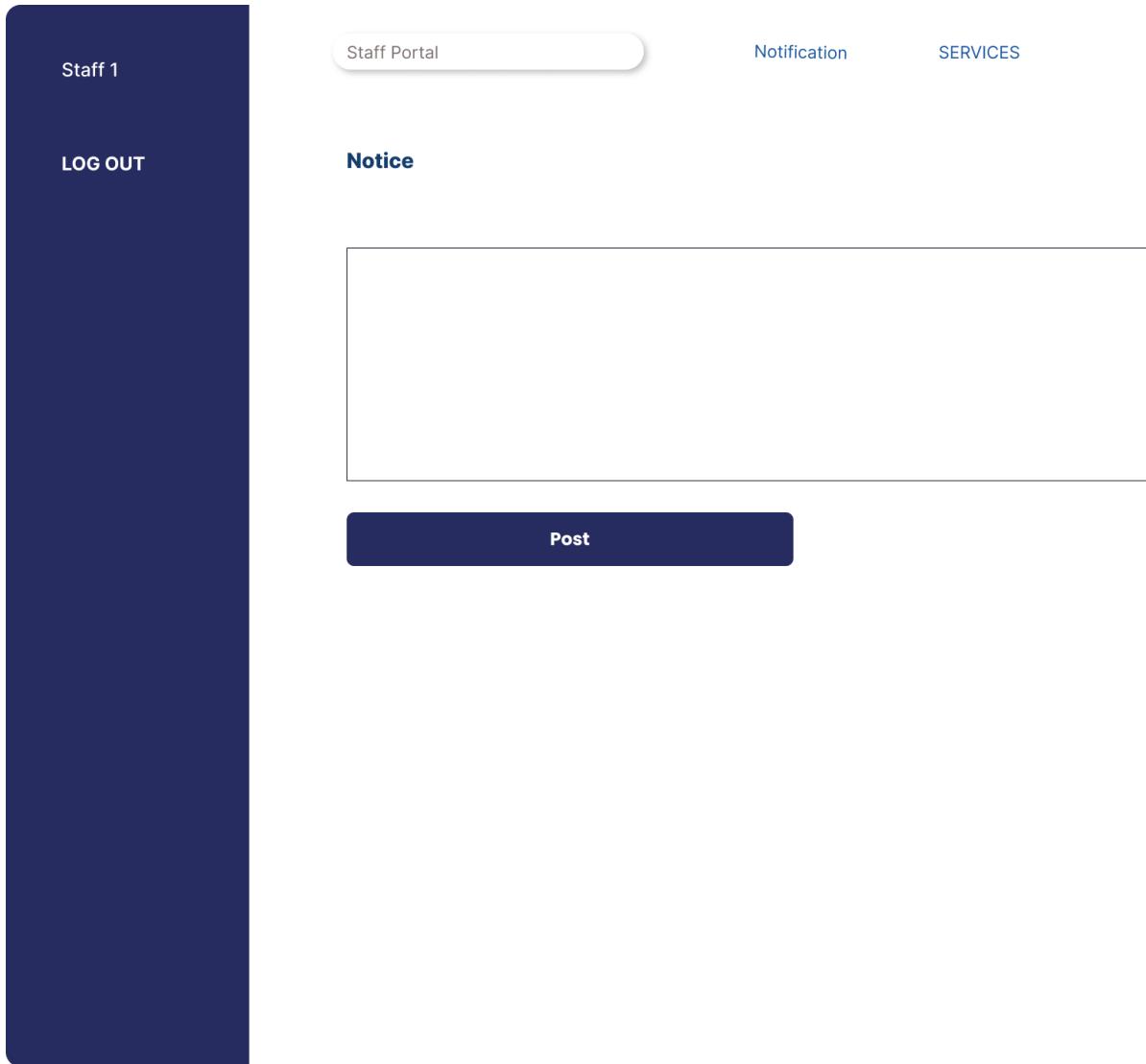


Figure 48 Post Notice (Prototype)

3.5.2.12.4. Activity Diagram:

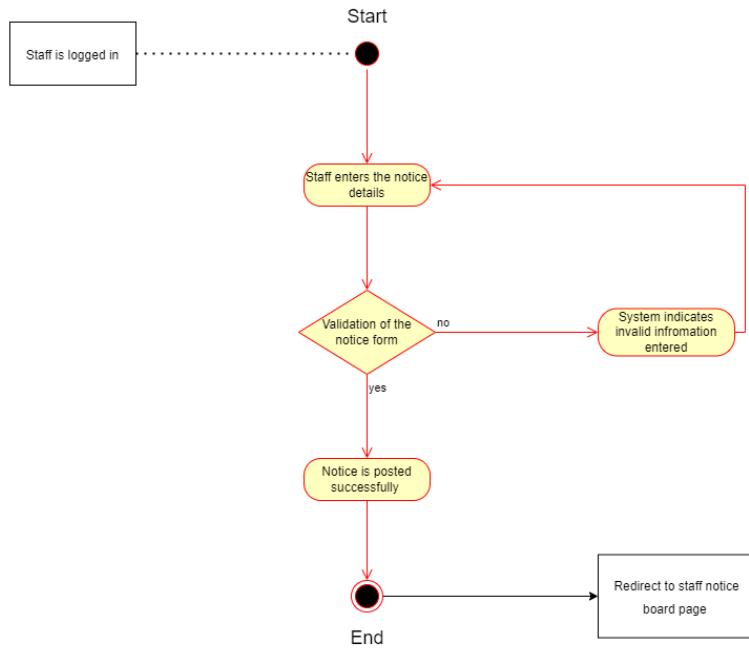


Figure 49 Post Notice (Activity Diagram)

3.5.2.12.5. Sequence Diagram:

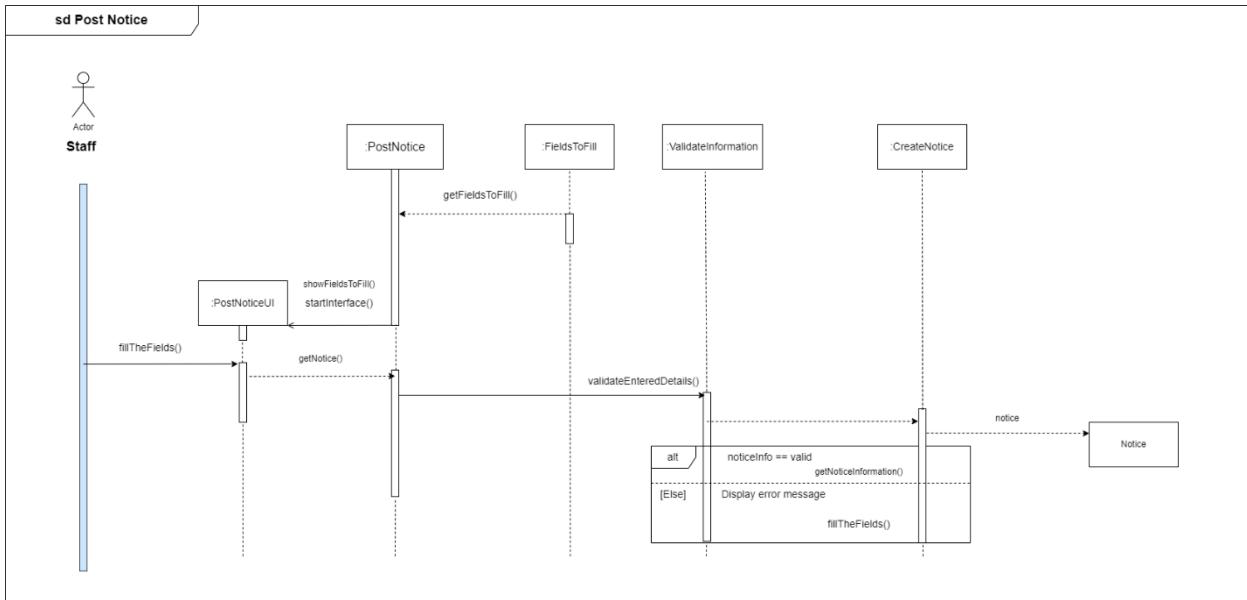


Figure 50 Post Notice (Sequence Diagram)

3.5.2.13. View Notice:

3.5.2.13.1. High level use case:

Use Case:	View Notice
Actors:	Staff
Description:	Staff clicks on the View Notice option from navigation bar. Then the Staff Notice View Page is displayed.

Table 14 View Notice (Staff) (High level use case)

3.5.2.13.2. Activity Diagram:

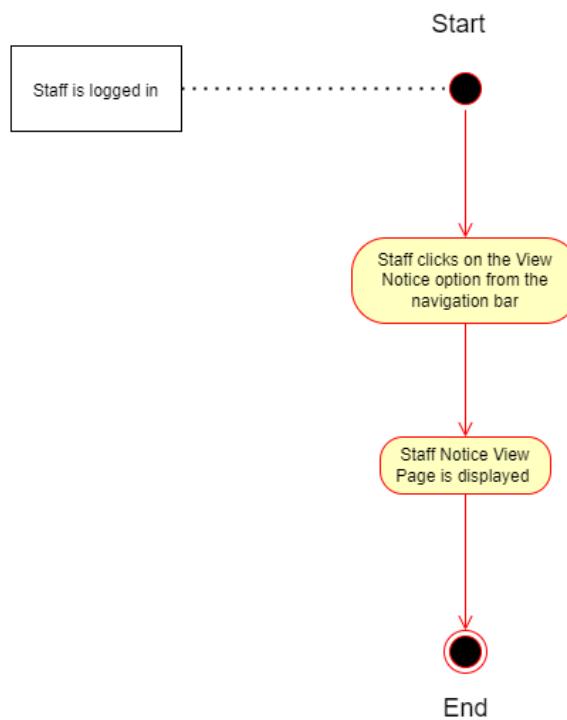


Figure 51 View Notice (Staff Page) (Activity Diagram)

3.5.2.13.3. Sequence Diagram:

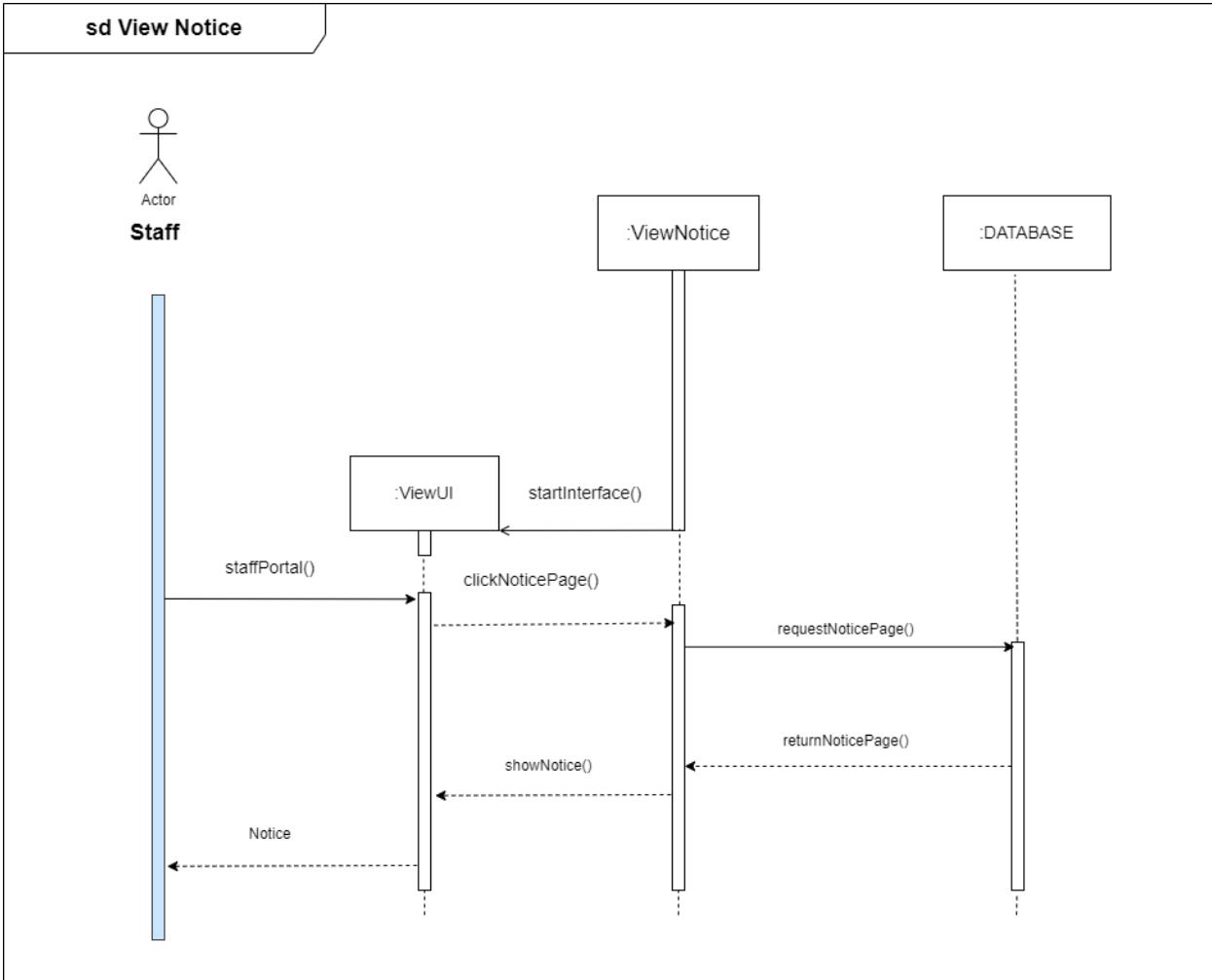


Figure 52 View Notice (Staff Page) (Sequence Diagram)

3.5.2.14. Change Password:

3.5.2.14.1. High level use case:

Use Case:	Change Password.
Actors:	Staff
Description:	Staff clicks on the Change Password option from settings. Then the password change form is displayed. Staff enters details to change password (old password, new password and confirm password). If the entered details are valid, system redirects to change password successful confirmation page.

Table 15 Change password (Staff) (High level use case)

3.5.2.14.2. Activity Diagram:

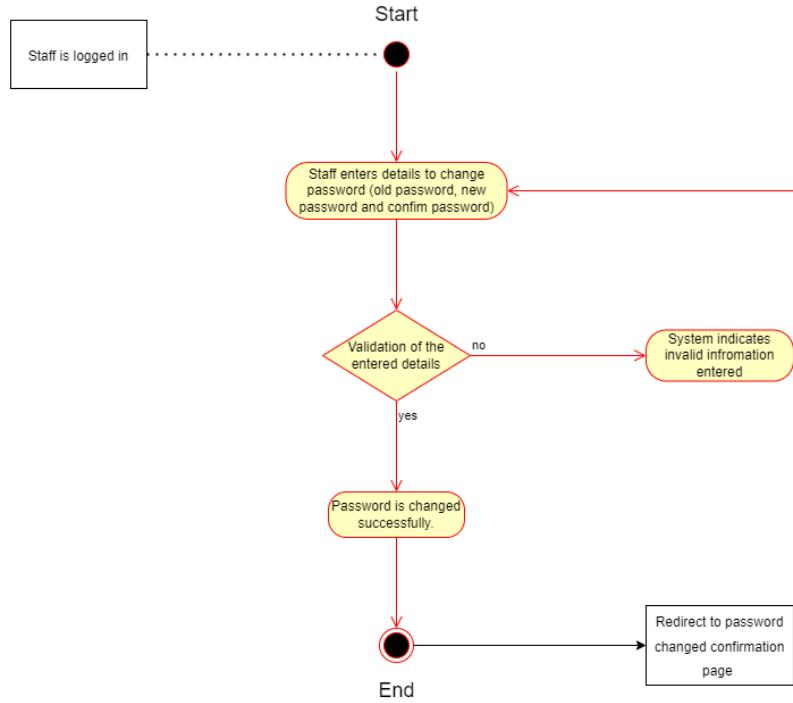


Table 16 Change password (staff) (Activity Diagram)

3.5.2.14.3. Sequence Diagram:

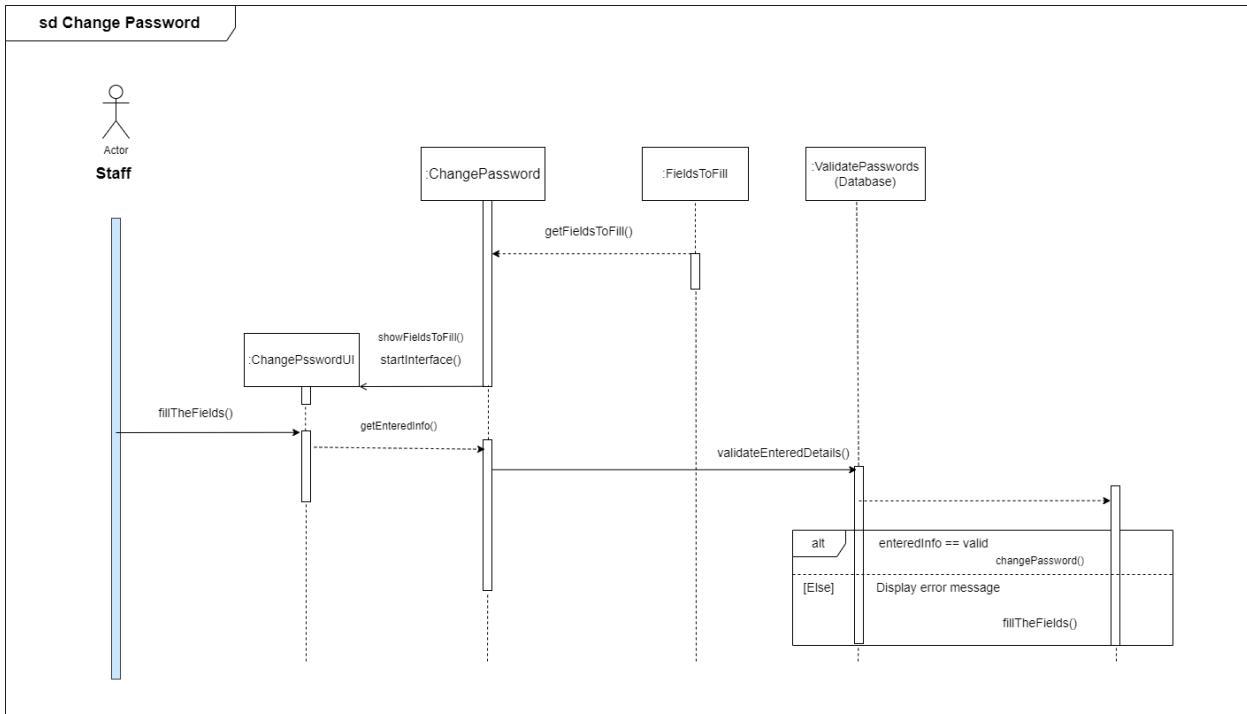


Table 17 Change password (staff) (Sequence Diagram)

3.5.3. For Teacher:

3.5.3.1. Use case diagram:

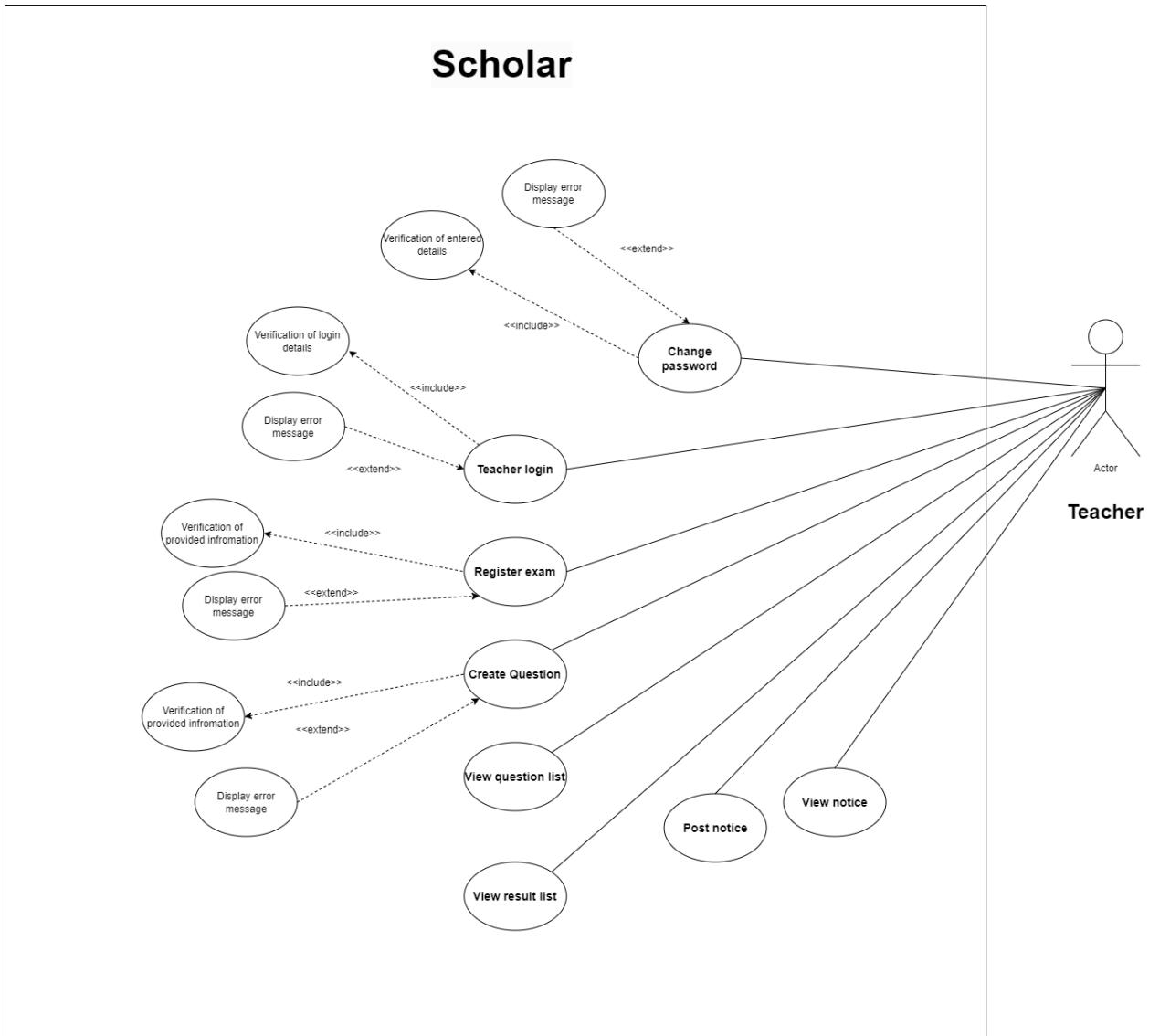


Figure 53 Teacher (Use Case diagram)

3.5.3.2. Teacher Login:

3.5.3.2.1. High level use case:

Use Case:	Teacher Login
Actors:	Teacher
Description:	After the registration of the teacher, he/she is provided with the login username and password through email. The teacher can log in to the system using the correct username and password.

Table 18 Teacher Login (High level use case)

3..3.2.2. Wireframe:

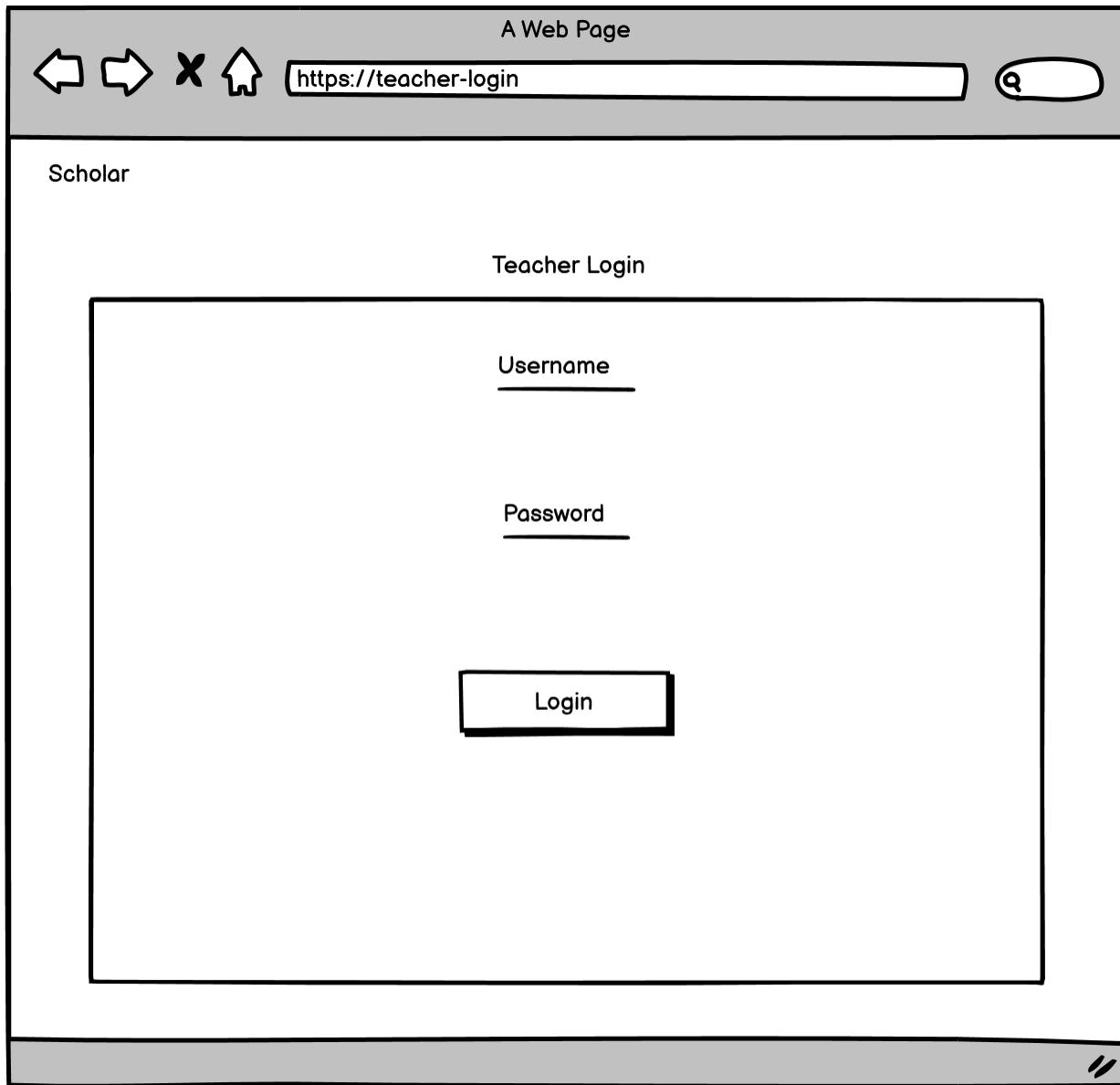
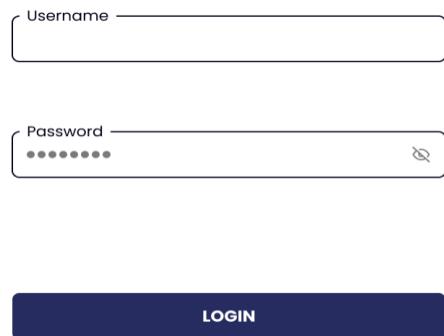


Figure 54 Teacher Login (Wireframe)

3.5.3.2.3. Prototype Design:

Scholar

Teacher Login



The image shows a prototype design for a Teacher Login page. At the top left, the word "Scholar" is written in a bold, black, sans-serif font. Below it, the title "Teacher Login" is centered in a bold, black, sans-serif font. The main form area contains two input fields: a "Username" field with a placeholder "Username" and a "Password" field with a placeholder "Password" and a password strength indicator showing six solid dots. A "LOGIN" button is located at the bottom of the form.

Username _____

Password _____

••••••••

LOGIN

Figure 55 Teacher Login (Prototype Design)

3.5.3.2.4. Activity Diagram:

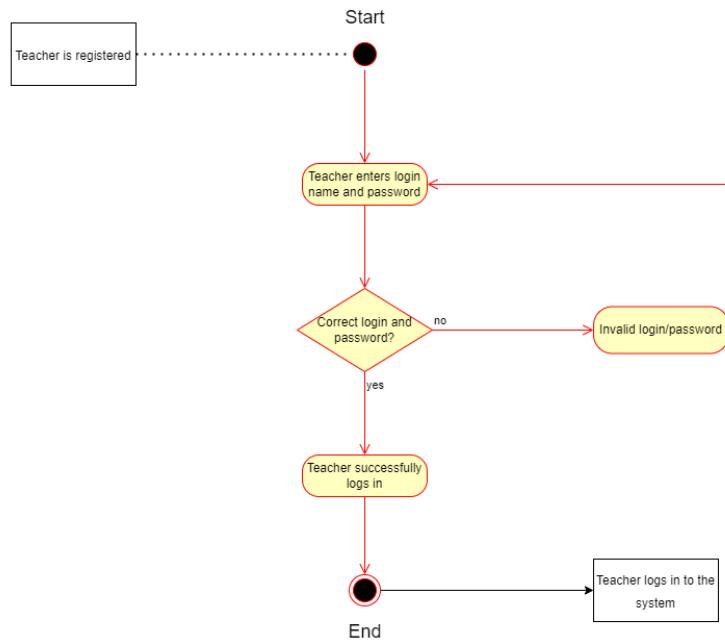


Figure 56 Teacher Login (Activity Diagram)

3.5.3.2.5. Sequence Diagram:

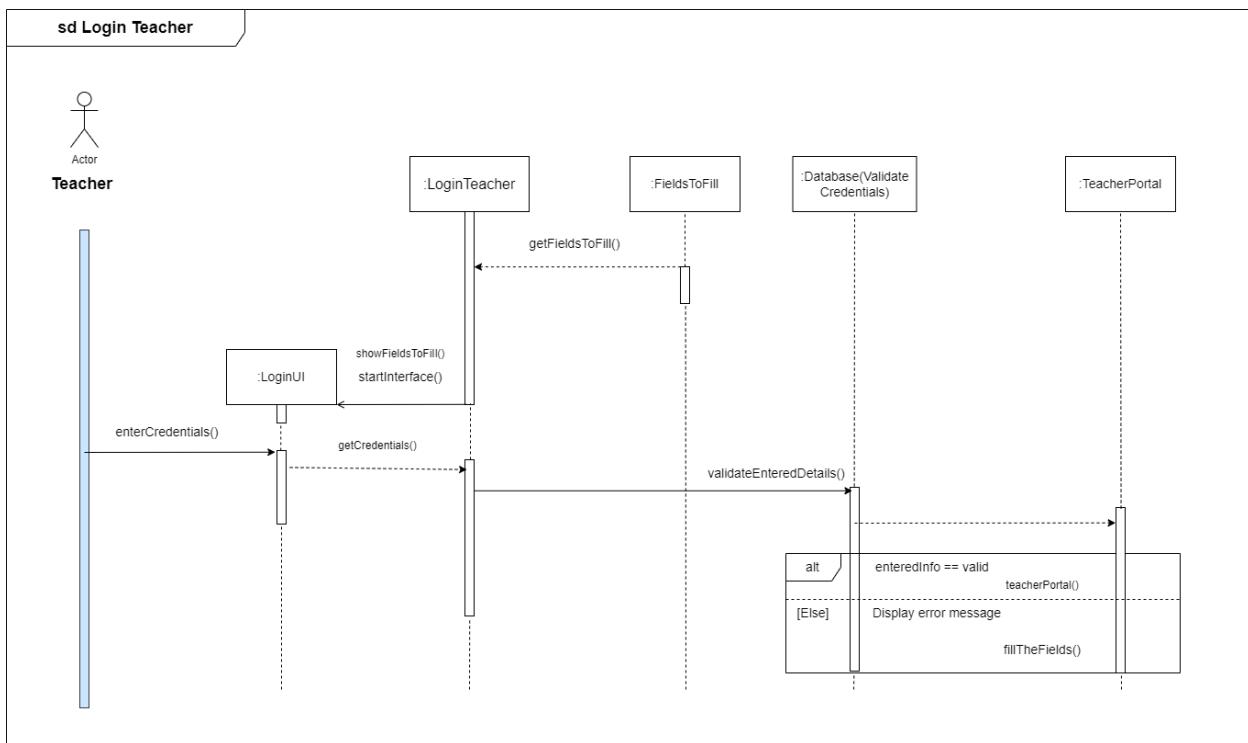


Figure 57 Teacher Login (Sequence Diagram)

3.5.3.3. Register Exam (By teacher):

3.5.3.3.1. High level use case:

Use Case:	Register Exam (By teacher)
Actors:	Teacher
Description:	Teacher enters exam details in the exam registration form. The exam gets registered after the validation of the entered details.

Table 19 Register Exam (By Teacher) (High level use case)

3.5.3.3.2. Activity Diagram:

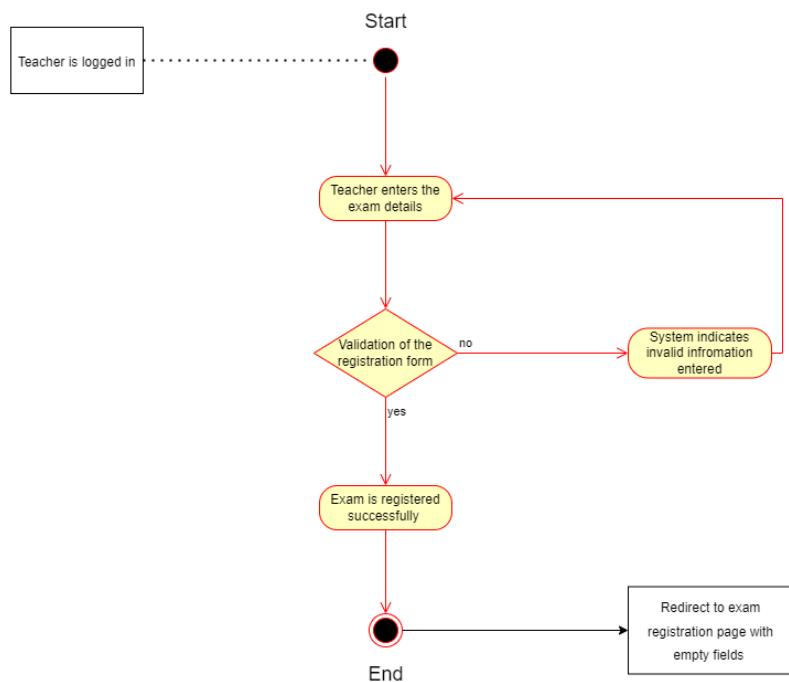


Figure 58 Register Exam (Activity Diagram)

3.5.3.3.3. Sequence Diagram:

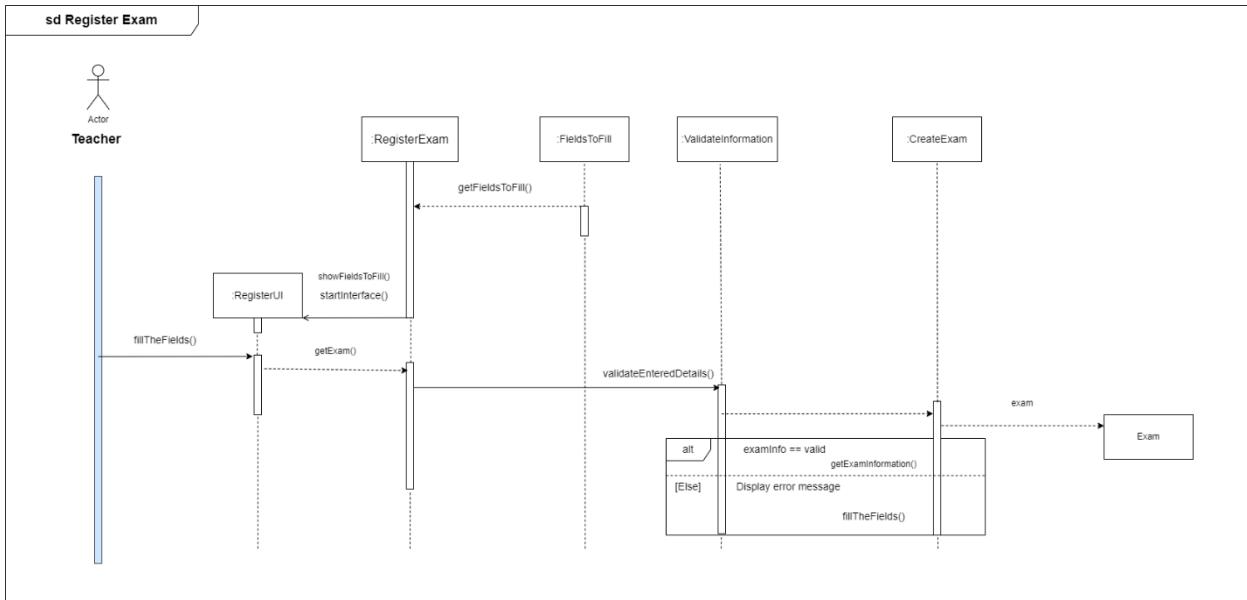


Figure 59 Register Exam (Sequence Diagram)

3.5.3.4. Create Question:

3.5.3.4.1. High level use case:

Use Case:	Create Question
Actors:	Teacher
Description:	Teacher selects the registered exam. Then teacher enters the question. Now teacher enters the answers to the respective question. Teacher selects the correct answer. The question gets created after the validation of the entered details.

Table 20 Create Question (High level use case)

3.5.3.4.2. Wireframe:

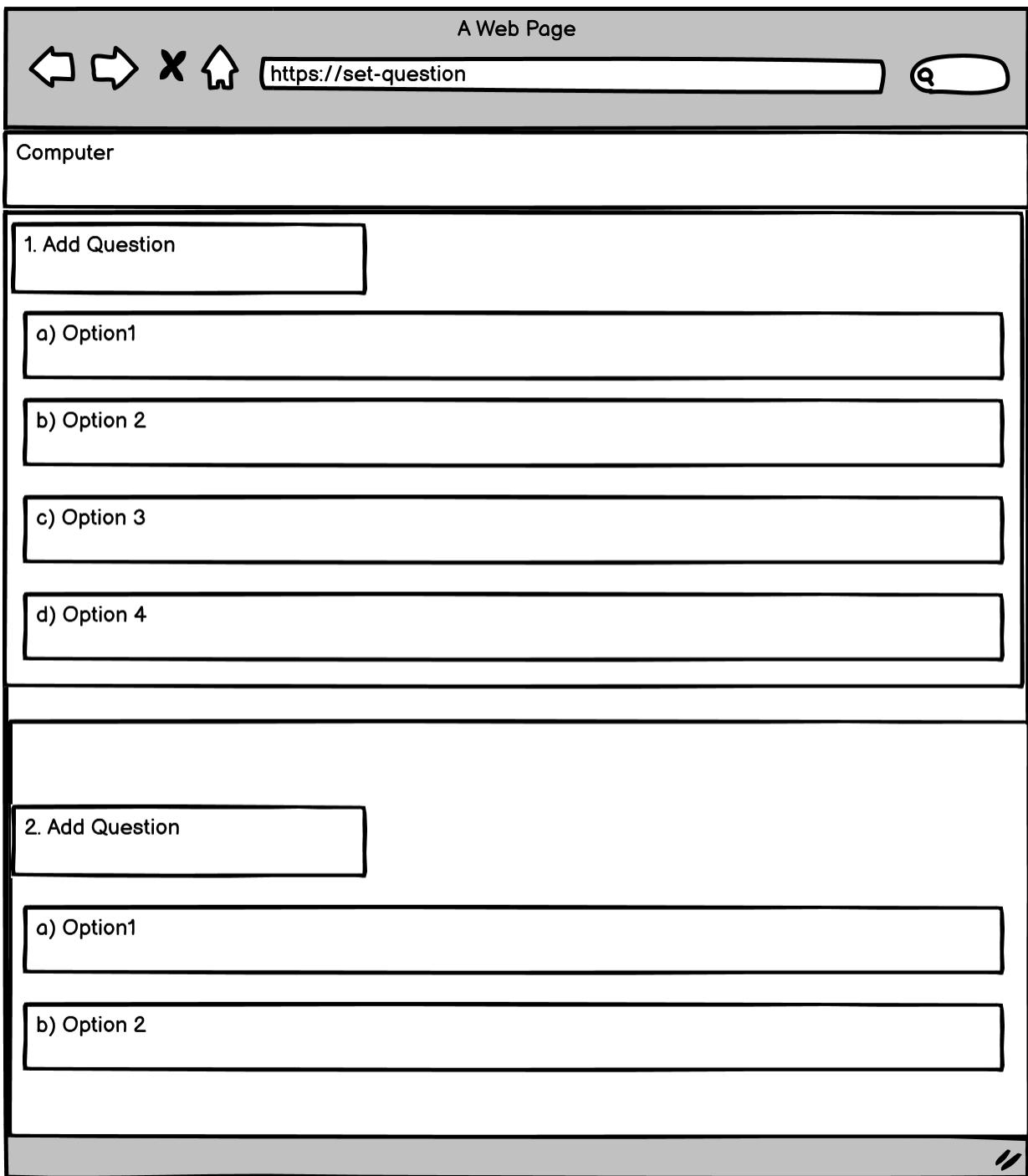


Figure 60 Create Question (Wireframe)

3.5.3.4.3. Prototype:

The screenshot shows a user interface for creating a question. At the top, there is a dark blue header bar with two buttons: "Teacher Portal" on the left and "Conduct Exam" on the right. Below the header, the word "Computer" is displayed in a rounded rectangular box. The main content area is titled "1st Terminal". It contains a question box labeled "1. Who was the father of computer" with four answer options: "Michael Faraday", "Charles Babbage", "Adam Smith", and "Add answer". Below this is another question box labeled "2. When was the first computer made?" with three answer options: "1945 AD", "1500 AD", "1290 AD", and "None of the above". The "None of the above" option is highlighted with a green background.

Teacher Portal

Conduct Exam

Computer

1st Terminal

1. Who was the father of computer

Michael Faraday

Charles Babbage

Adam Smith

Add answer

2. When was the first computer made?

1945 AD

1500 AD

1290 AD

None of the above

Figure 61 Create Question (Prototype)

3.5.3.4.4. Activity Diagram:

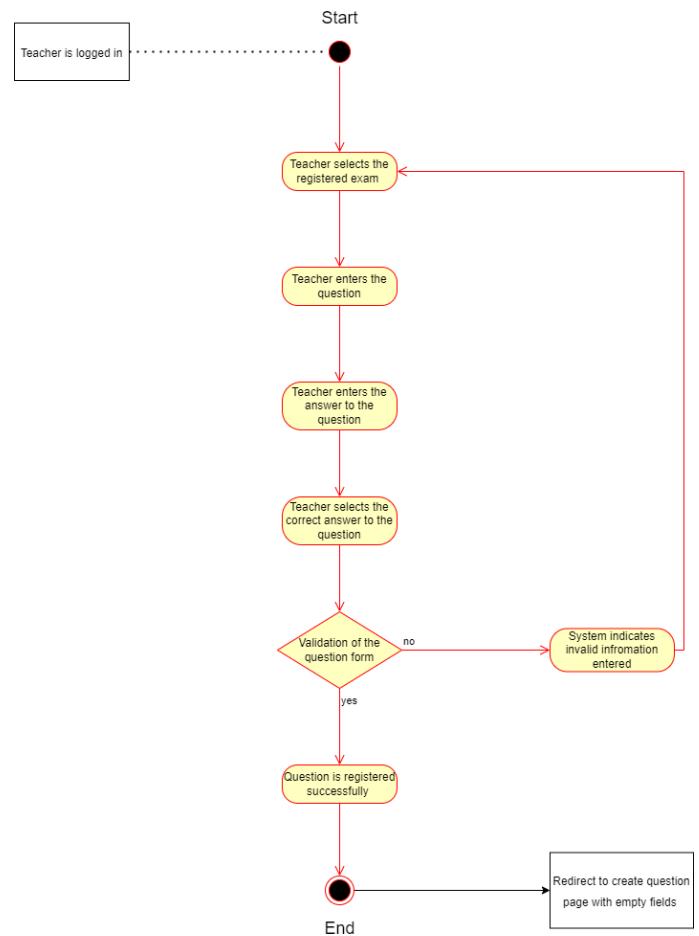


Figure 62 Create Question (Activity Diagram)

3.5.3.4.5. Sequence Diagram:

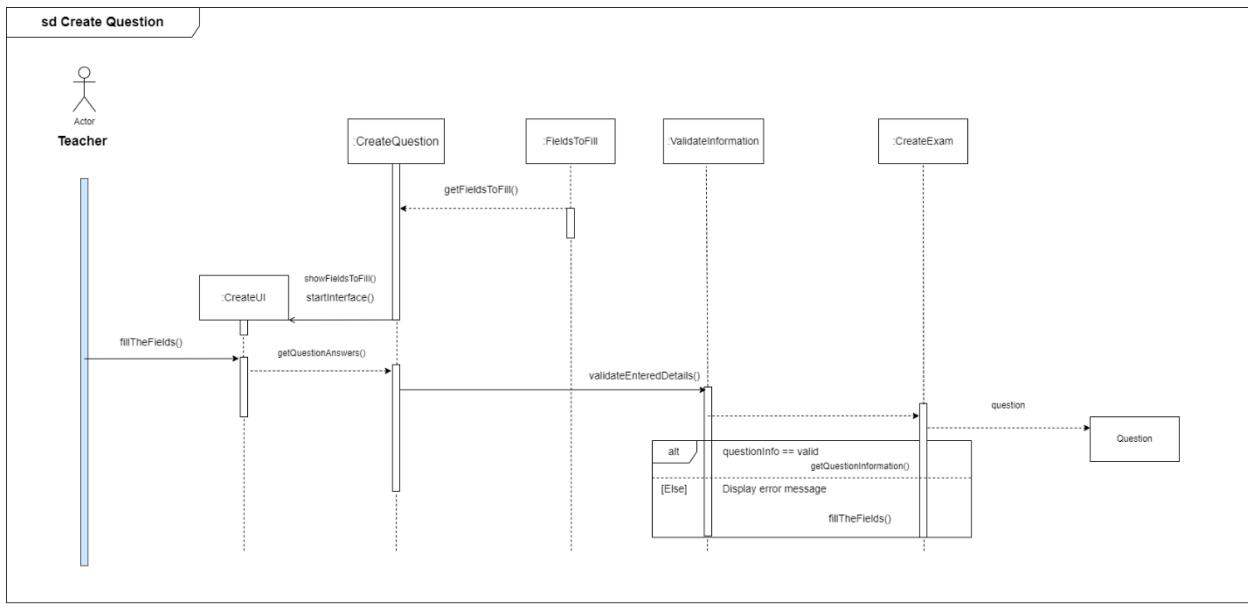


Figure 63 Create Question (Sequence Diagram)

3.5.3.5. Question List:

3.5.3.5.1. High level use case:

Use Case:	View question list
Actors:	Teacher
Description:	Teacher clicks on the Exam Material option from navigation bar. Then the teacher clicks on the Question List option to view the question list.

Table 21 Question List (High level use case)

3.5.3.5.2. Activity Diagram:

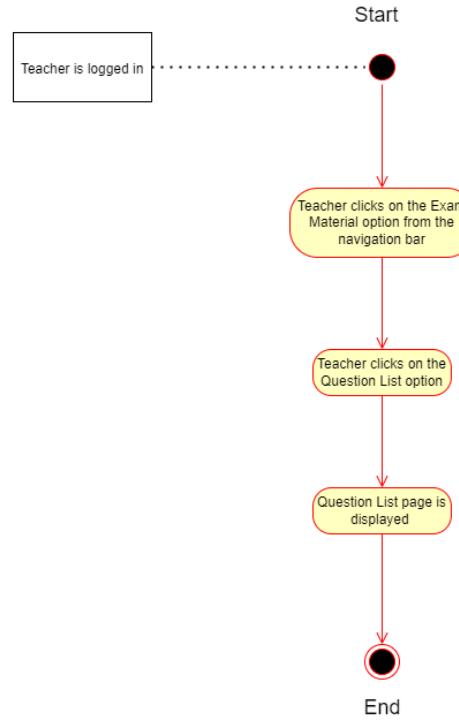


Figure 64 Question List (Activity Diagram)

3.5.3.5.3. Sequence Diagram:

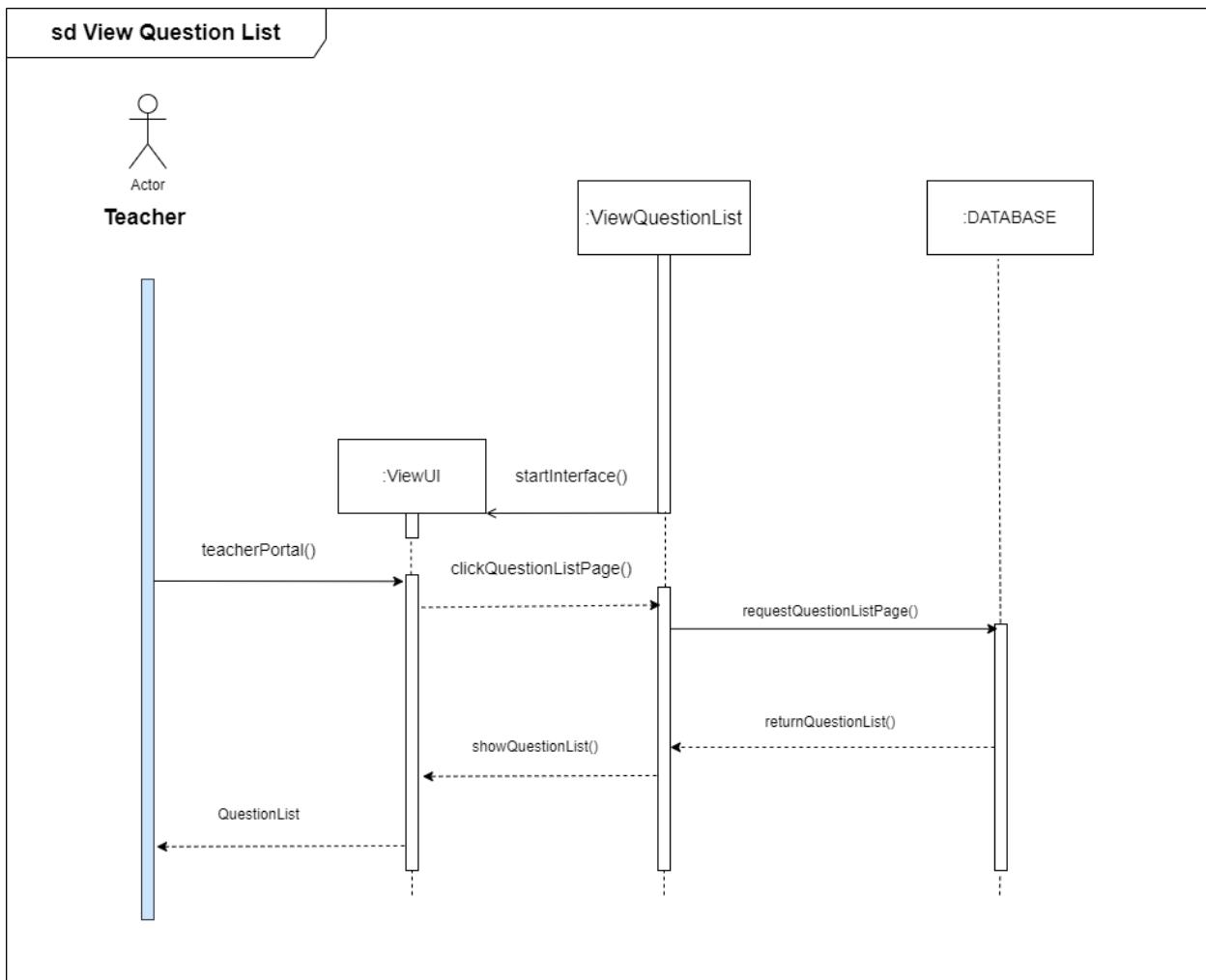


Figure 65 Question List (Sequence Diagram)

3.5.3.6. Exam Result List:

3.5.3.6.1. High level use case:

Use Case:	View exam result list
Actors:	Teacher
Description:	Teacher clicks on the Exam Material option from navigation bar. Then the teacher clicks on the Exam Result List option to view the exam result list.

Table 22 View exam result list (High level use case)

3.5.3.6.2. Activity Diagram:

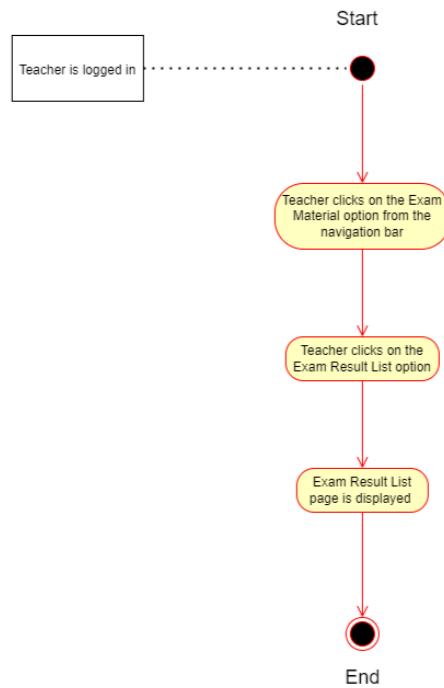


Figure 66 Exam Result List (Activity Diagram)

3.5.3.6.3. Sequence Diagram:

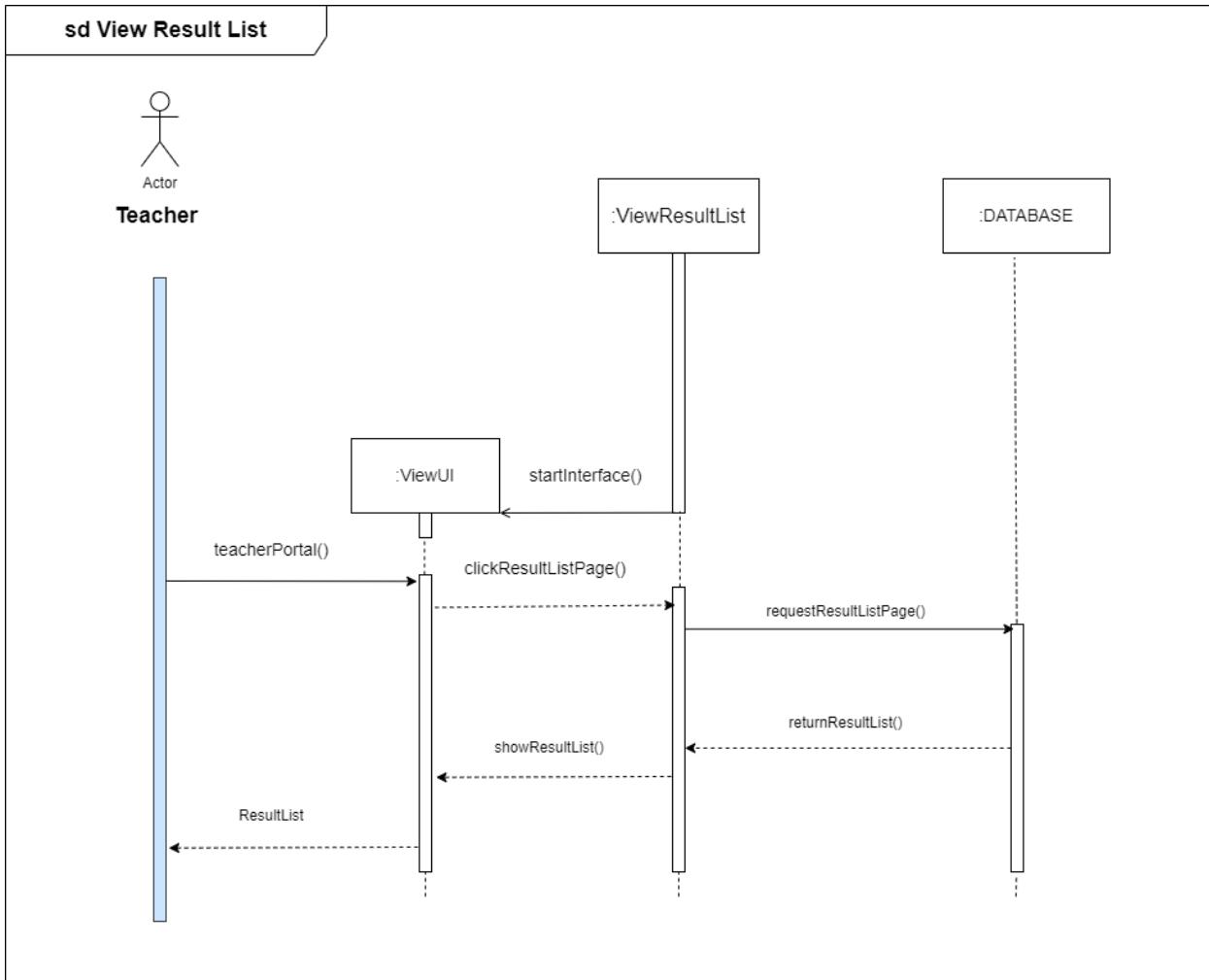


Figure 67 Exam Result List (Sequence Diagram)

3.5.3.7. Post Notice:

3.5.3.7.1. High level use case:

Use Case:	Post Notice
Actors:	Teacher
Description:	Teacher enters notice details in the add notice form. The notice gets posted after the validation of the entered details.

Table 23 Post Notice (High level use case)

3.5.3.7.2. Activity Diagram:

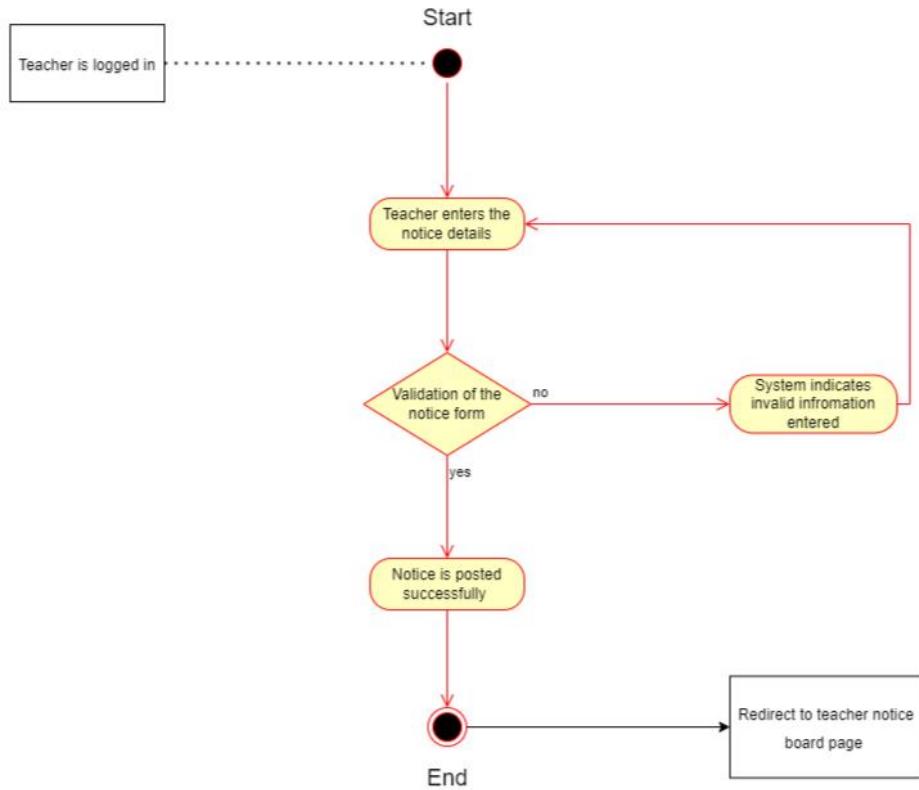


Figure 68 Post notice (Activity Diagram)

3.5.3.7.3. Sequence Diagram:

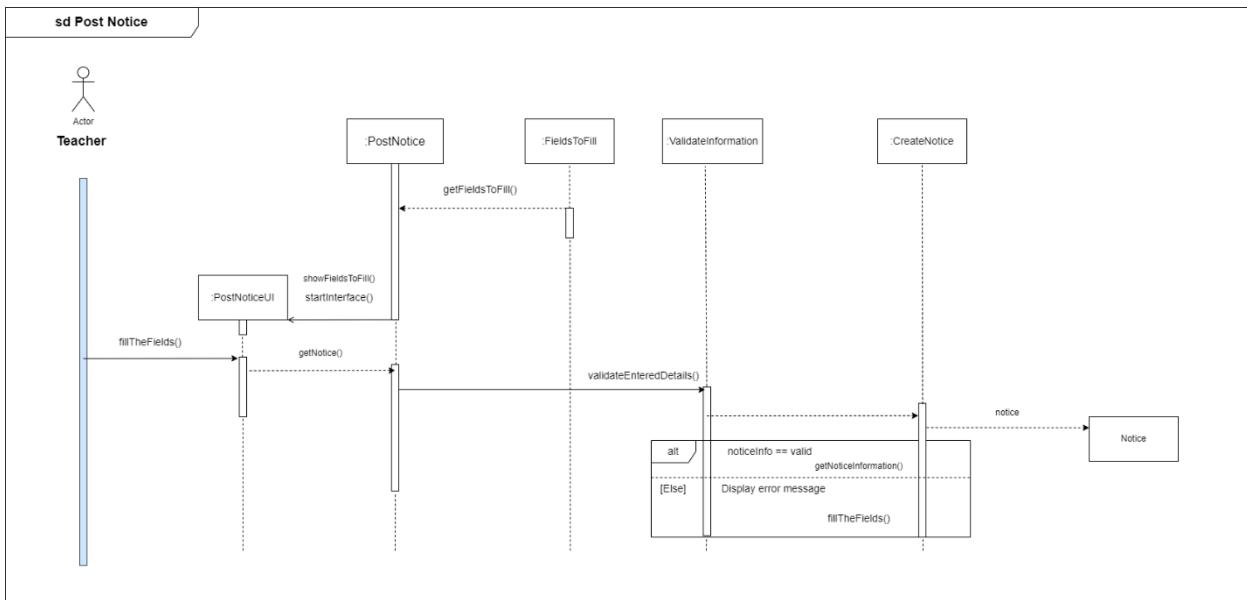


Figure 69 Post notice (Sequence Diagram)

3.5.3.8. View Notice:

3.5.3.8.1. High level use case:

Use Case:	View Notice
Actors:	Teacher
Description:	Teacher clicks on the View Notice option from navigation bar. Then the Teacher Notice View Page is displayed.

Table 24 View Notice

3.5.3.8.2. Activity Diagram:

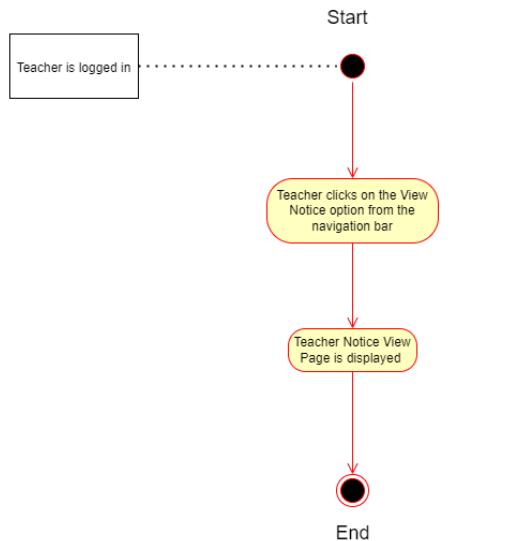


Figure 70 View Notice (Activity Diagram)

3.5.3.8.3. Sequence Diagram:

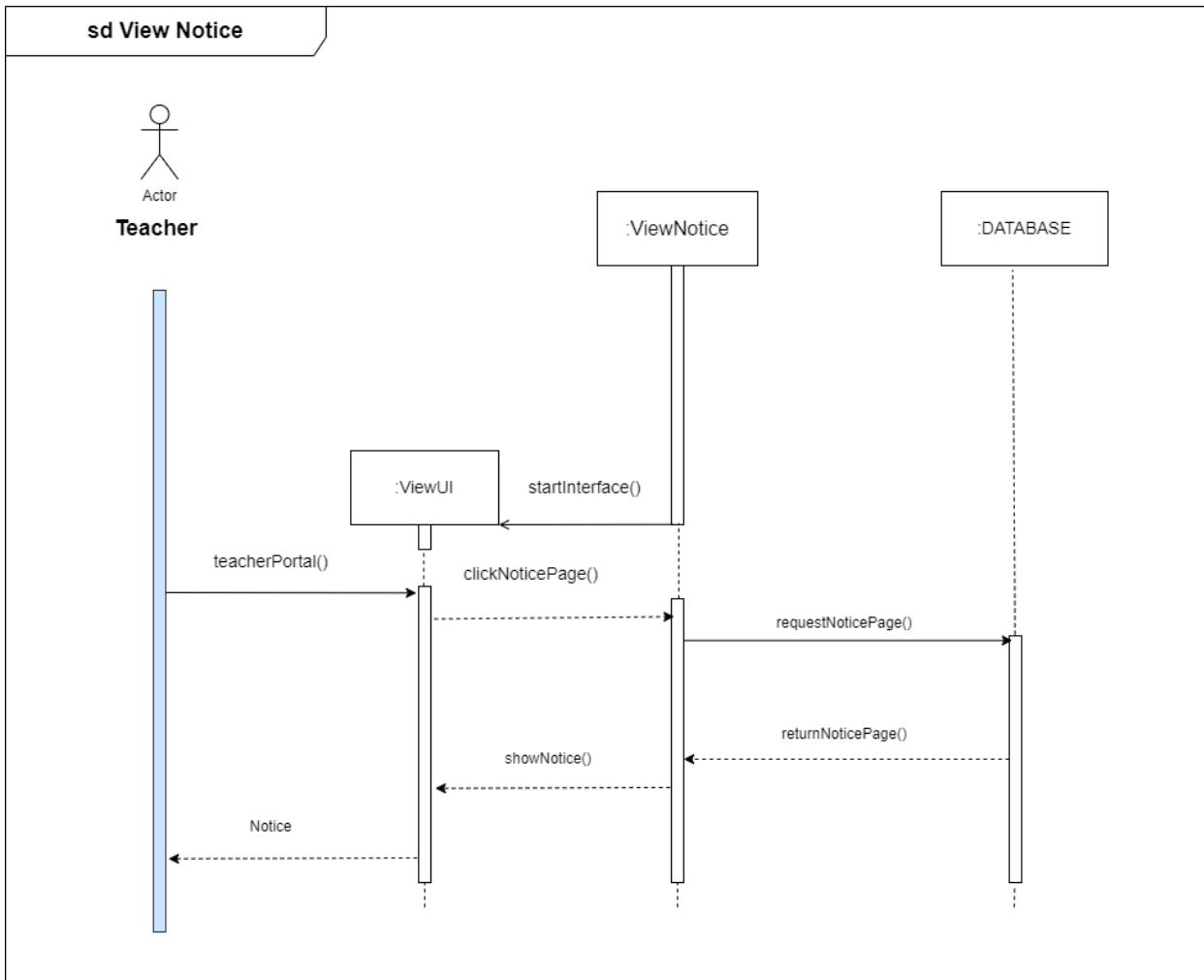


Figure 71 View Notice (Sequence Diagram)

3.5.3.9. Change Password:

3.5.3.9.1. High level use case:

Use Case:	Change Password.
Actors:	Teacher
Description:	Teacher clicks on the Change Password option from settings. Then the password change form is displayed. Teacher enters details to change password (old password, new password and confirm password). If the entered details are valid, system redirects to change password successful confirmation page.

Table 25 Change password (teacher) (High level use case)

3.5.3.9.2. Activity Diagram:

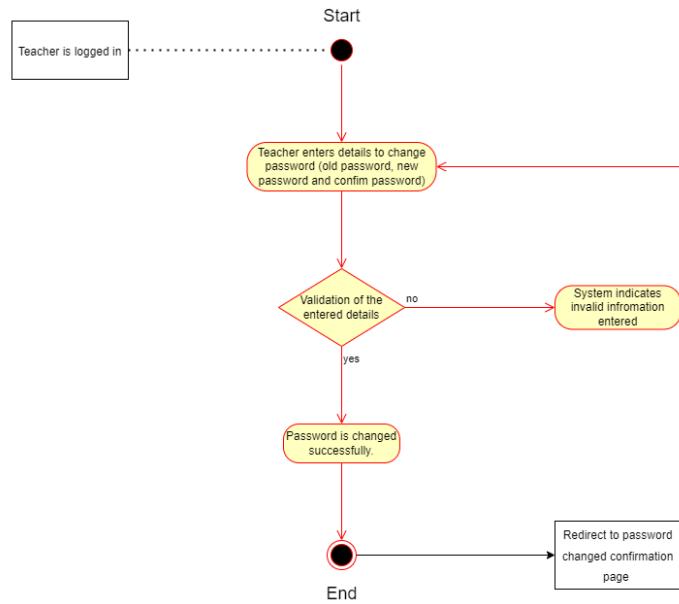


Figure 72 Change password (teacher) (Activity Diagram)

3.5.3.9.3. Sequence Diagram:

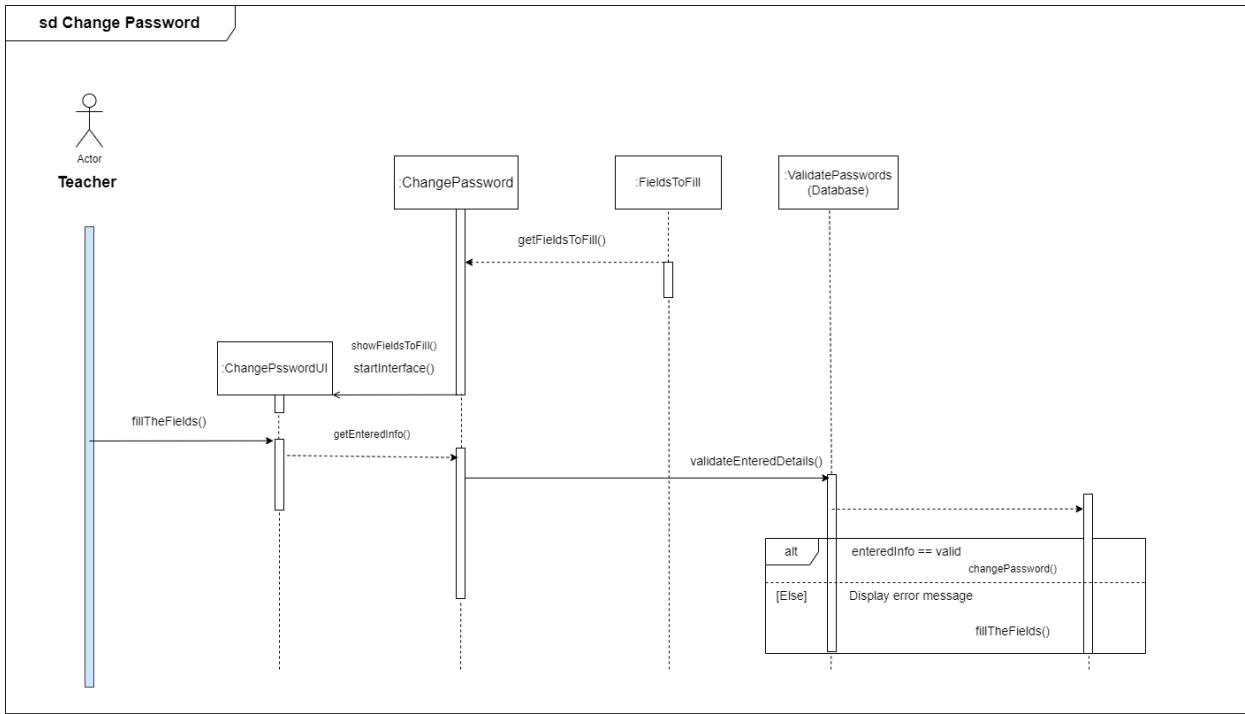


Figure 73 Change password (teacher) (Sequence Diagram)

3.5.4. For Student:

3.5.4.1. Use Case diagram:

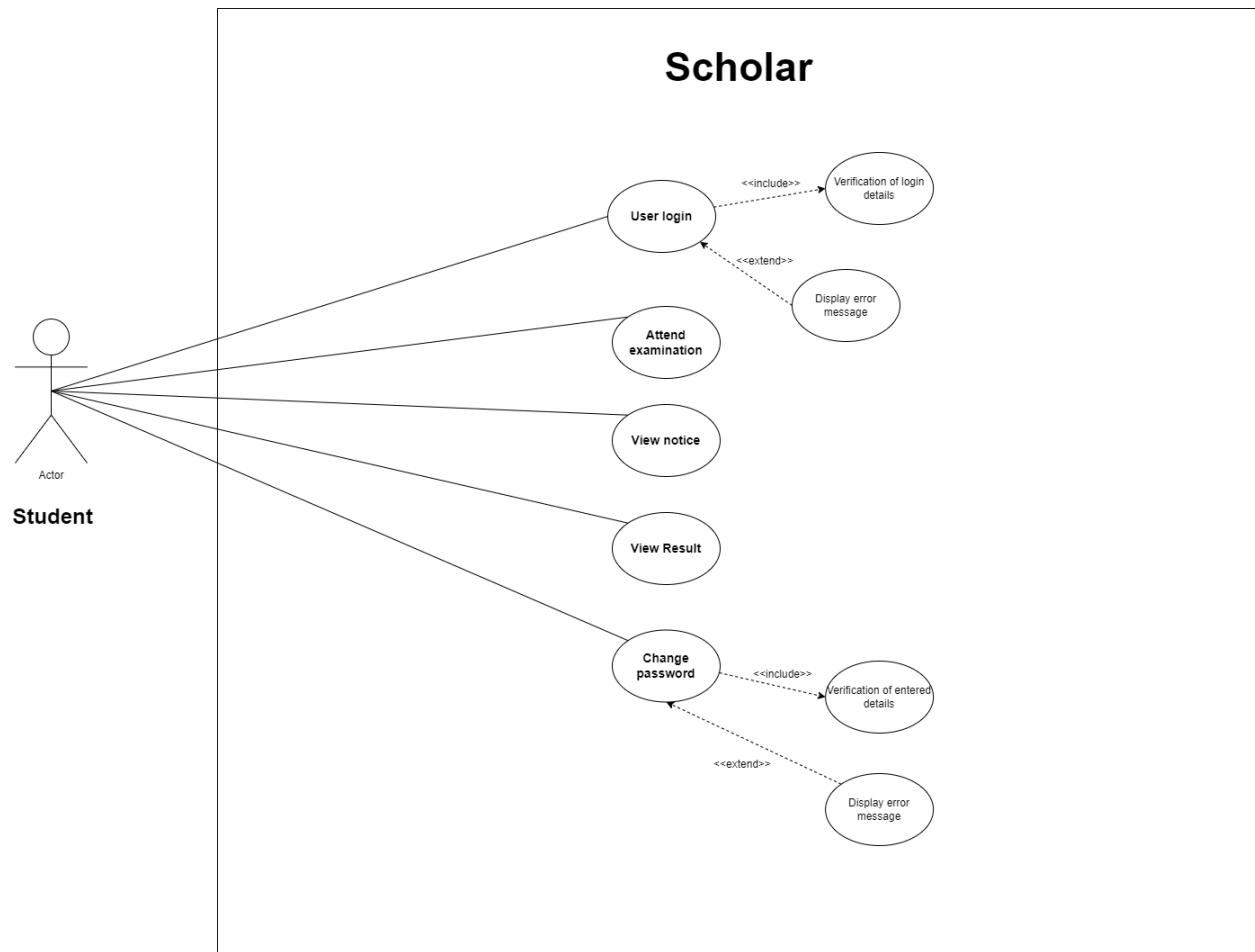


Figure 74 Student (Use Case diagram)

3.5.4.2. Student Login:

3.5.4.2.1. High level use case:

Use Case:	Student Login
Actors:	Student
Description:	After the registration of the student, he/she is provided with the login username and password through email. The student can log in to the system using the correct username and password.

Table 26 Student Login (High level use case)

3.5.4.2.2. Wireframe:

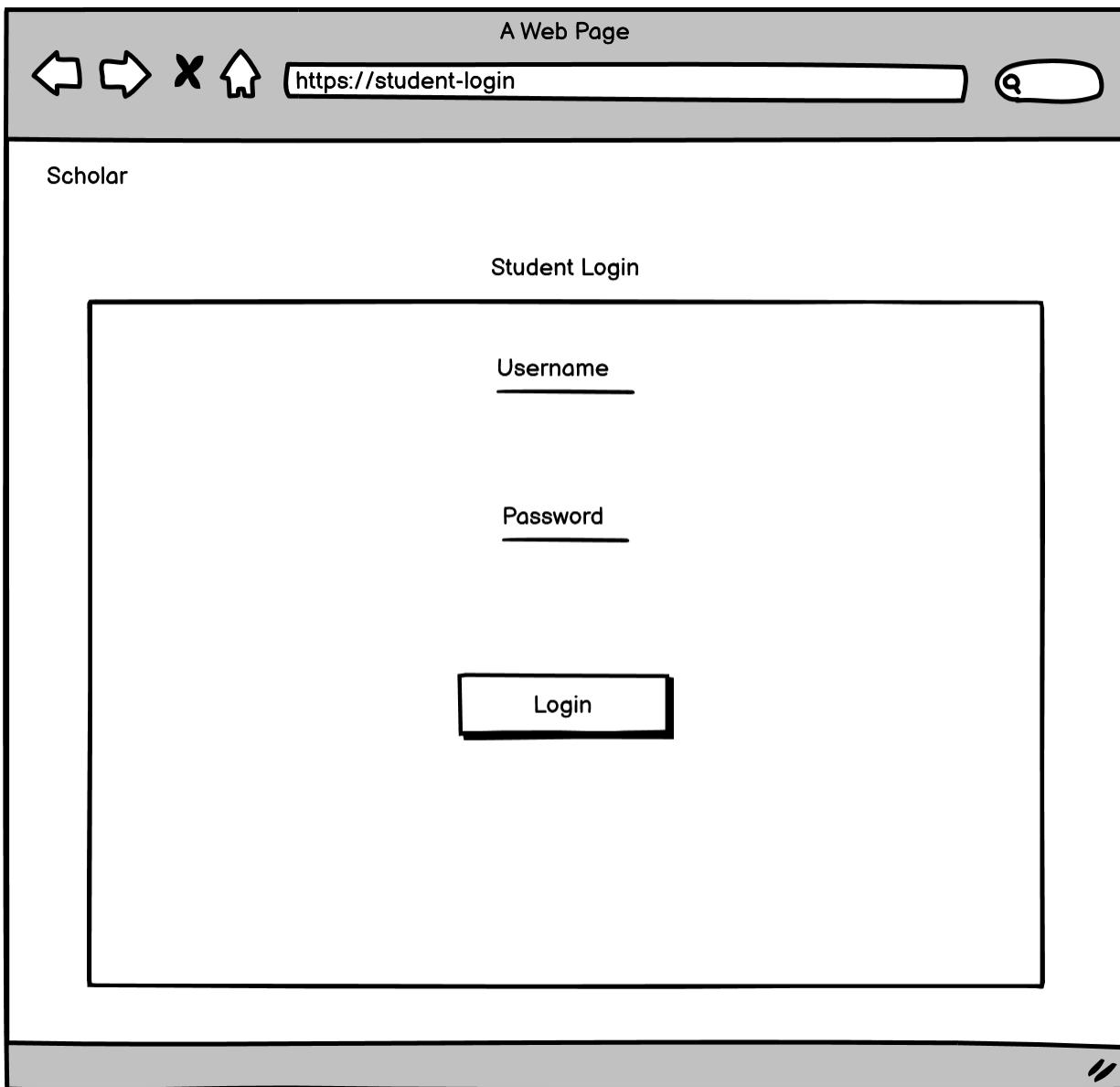


Figure 75 Student Login (Wireframe)

3.5.4.2.3. Prototype Design:

Scholar

Student Login

Username _____

Password _____ 

LOGIN

Figure 76 Student Login (Prototype Design)

3.5.4.2.4. Activity Diagram:

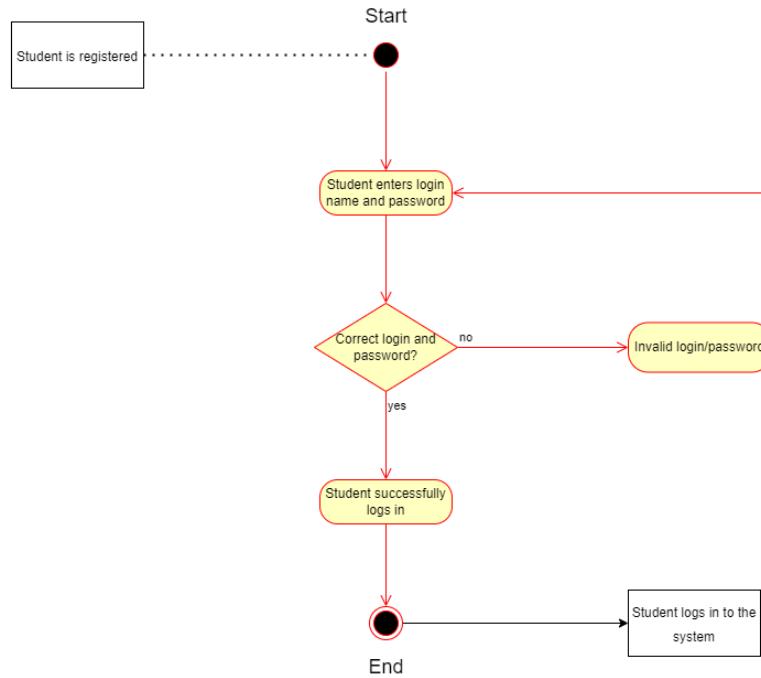


Figure 77 Student Login (Activity Diagram)

3.5.4.2.5. Sequence Diagram:

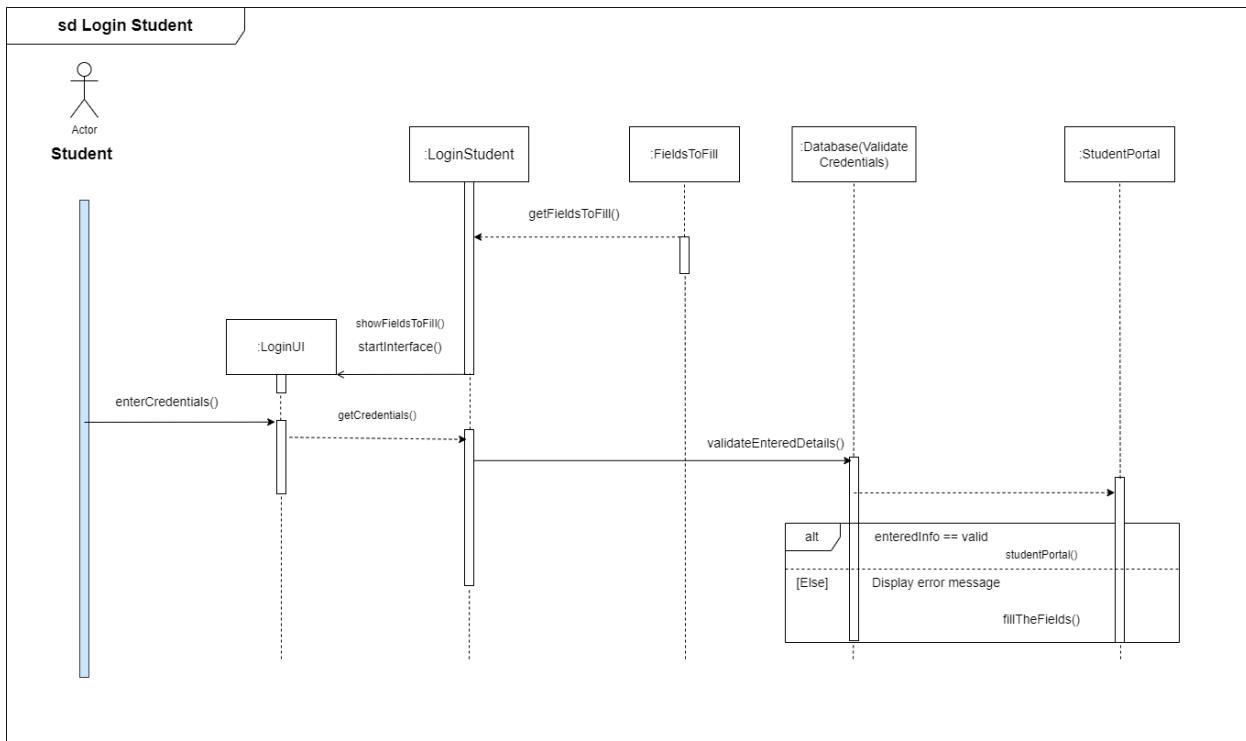


Figure 78 Student Login (Sequence Diagram)

3.5.4.3. Attend Examination:

3.5.4.3.1. High level use case:

Use Case:	Attend Examination
Actors:	Student
Description:	Student selects the exam from available exam. Then students answer to the respective question. Student submits the paper. Exam is then attended successfully.

Table 27 Attend Examination (High level use case)

3.5.4.3.2. Wireframe:

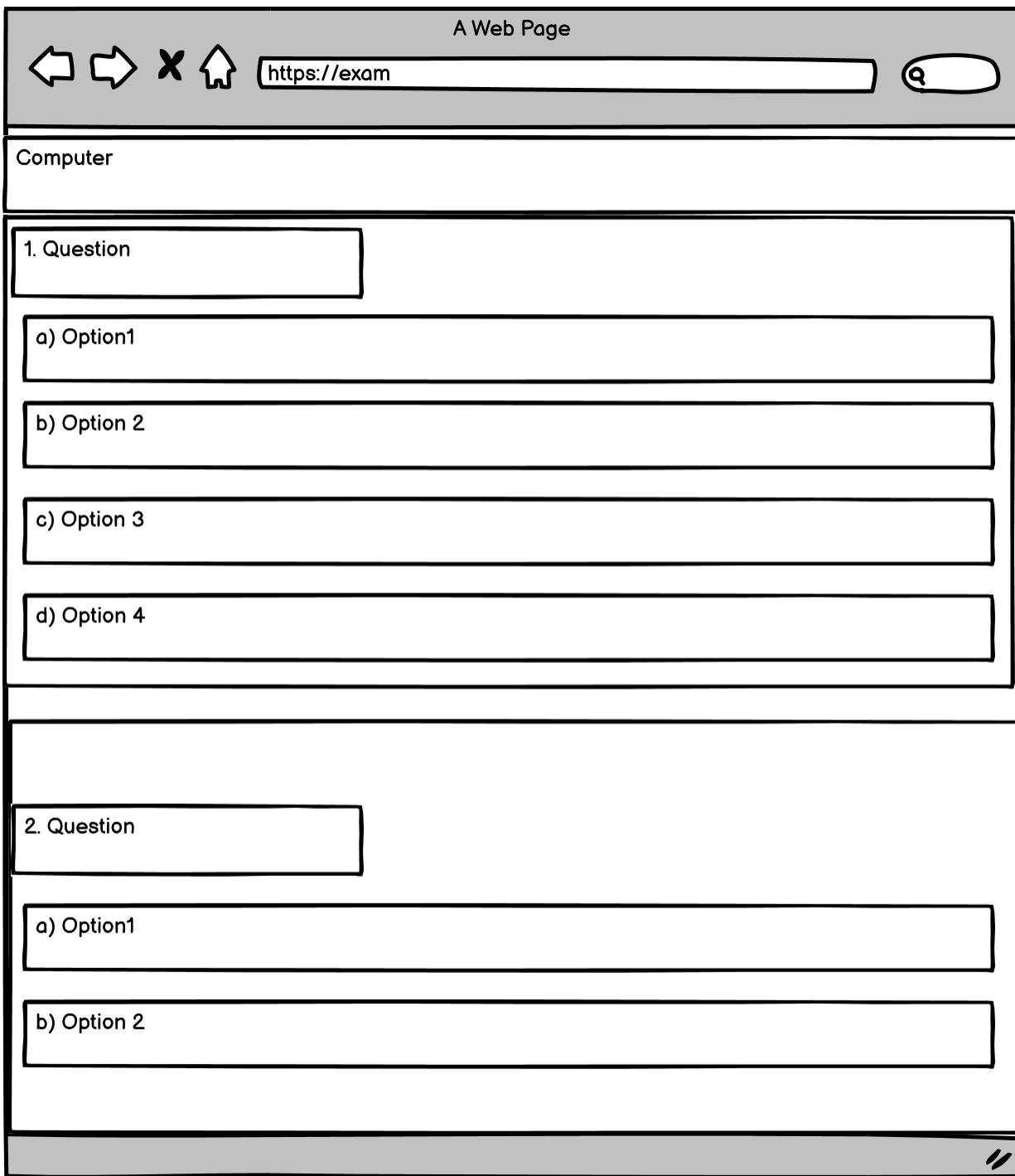


Figure 79 Attend Examination (Wireframe)

3.5.4.3.3. Prototype:

Computer

1. Who was the father of computer

Michael Faraday

Charles Babbage

Adam Smith

Philo Taylor Farnsworth II

2. When was the first computer made??

1945 AD

1500 AD

1290 AD

None of the above

Finish Test

Figure 80 Attend Examination (Prototype)

3.5.4.3.4. Activity Diagram:

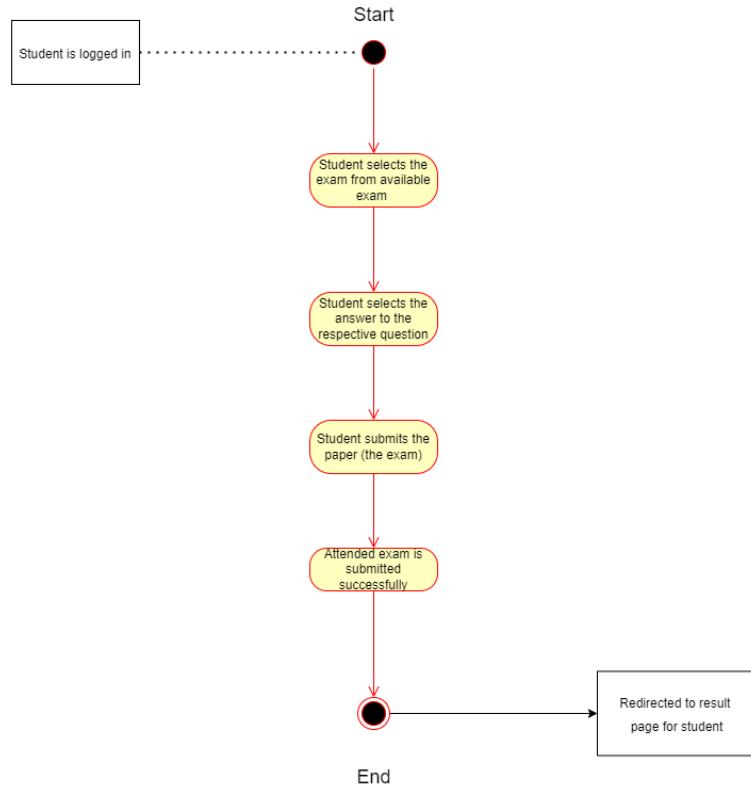


Figure 81 Attend Examination (Activity Diagram)

3.5.4.3.5. Sequence Diagram:

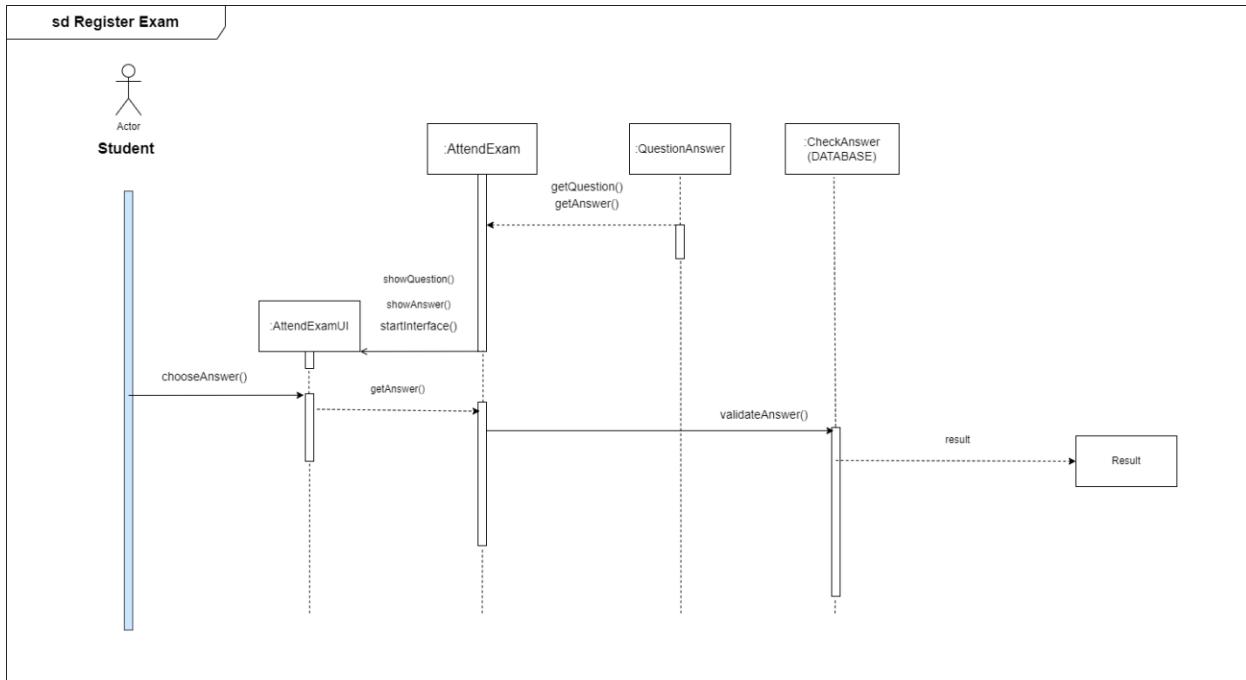


Figure 82 Attend Exam (Sequence Diagram)

3.5.4.4. View Result:

3.5.4.4.1. High level use case:

Use Case:	View Result
Actors:	Student
Description:	Student selects the exam from available exam. Then students answer to the respective question. Student submits the paper. After the exam is attended successfully, the student is redirected to result page.

Table 28 View Result (Student) (High level use case)

3.5.4.4.2. Wireframe:

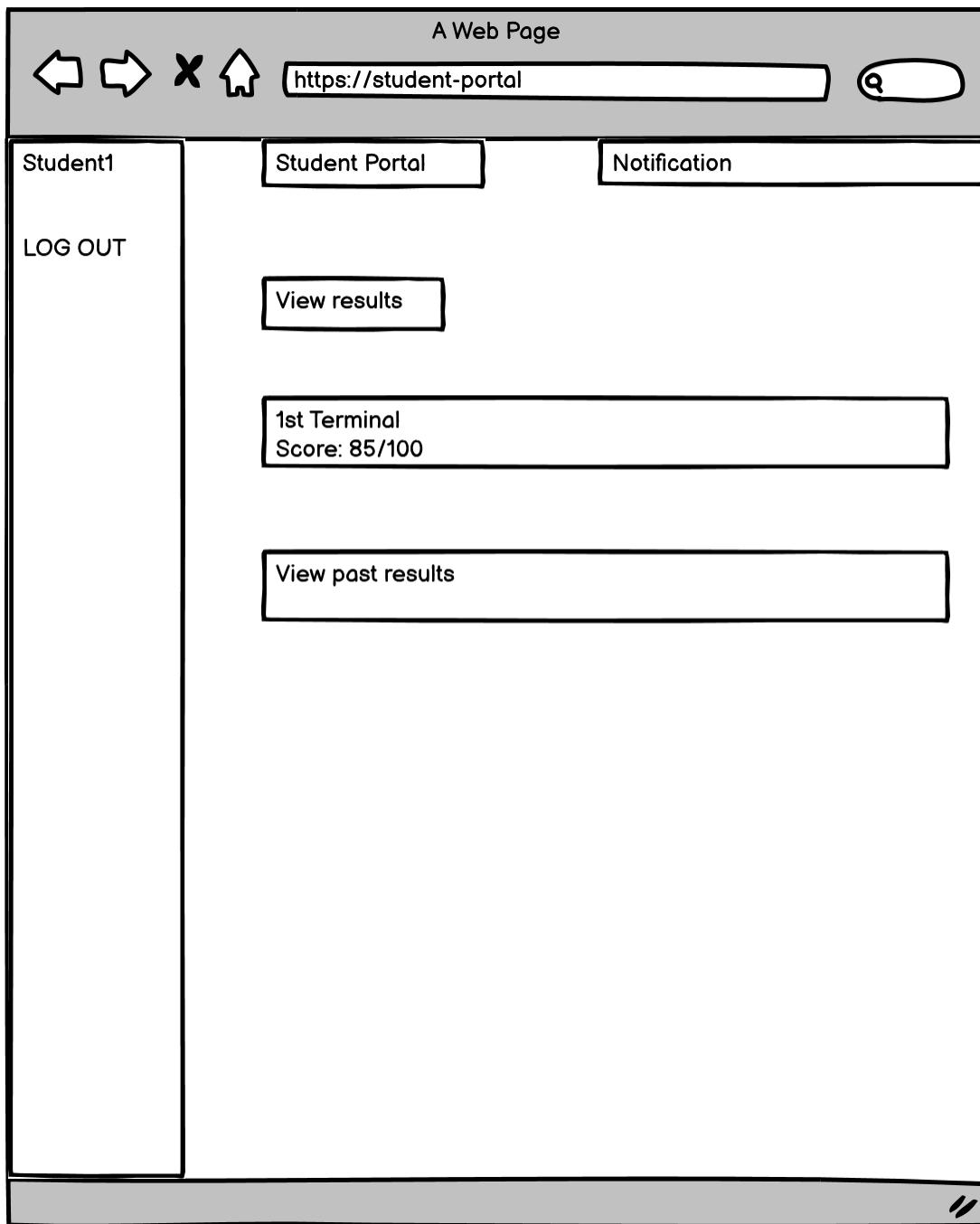


Figure 83 View Result (Wireframe)

3.5.4.4.3. Prototype:

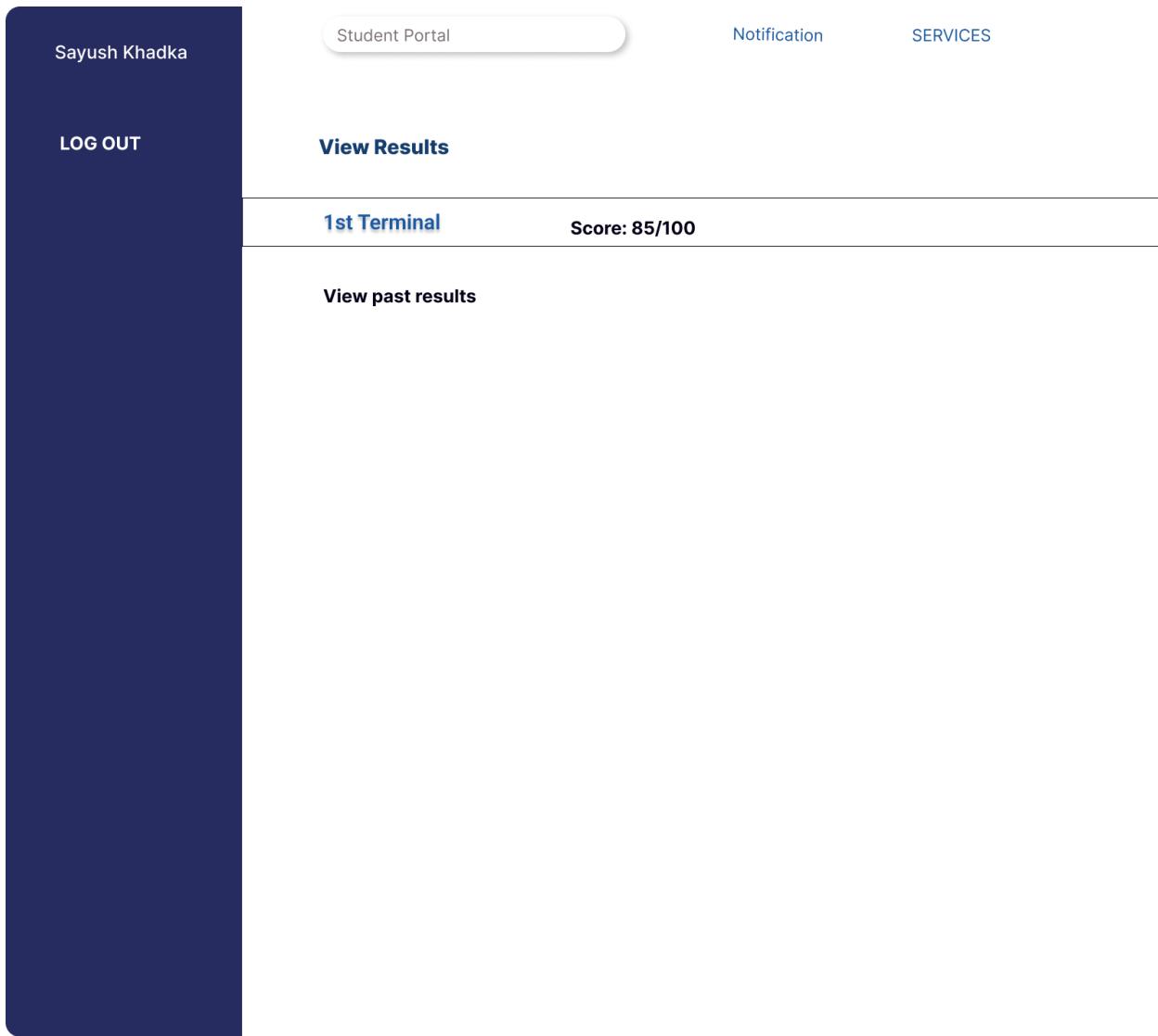


Figure 84 View Result (Prototype)

3.5.4.4.4. Activity Diagram:

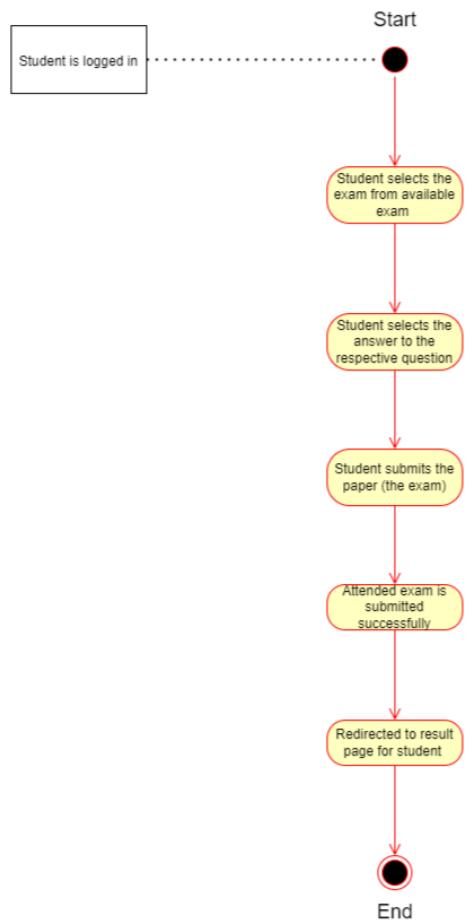


Figure 85 View Result (Activity Diagram)

3.5.4.4.5. Sequence Diagram:

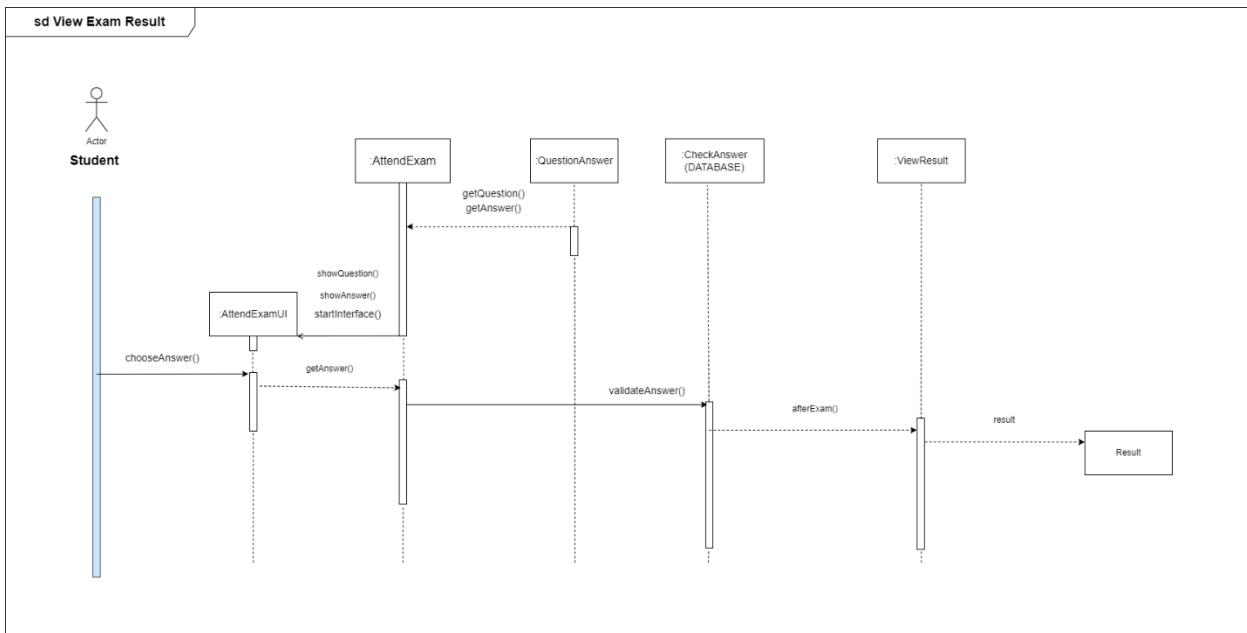


Figure 86 View Result (Sequence Diagram)

3.5.4.5. View Notice:

3.5.4.5.1. High level use case:

Use Case:	View Notice
Actors:	Student
Description:	Student clicks on the View Notice option from navigation bar. Then the Student Notice View Page is displayed.

Table 29 View notice (High level use case)

3.5.4.5.3. Activity Diagram:

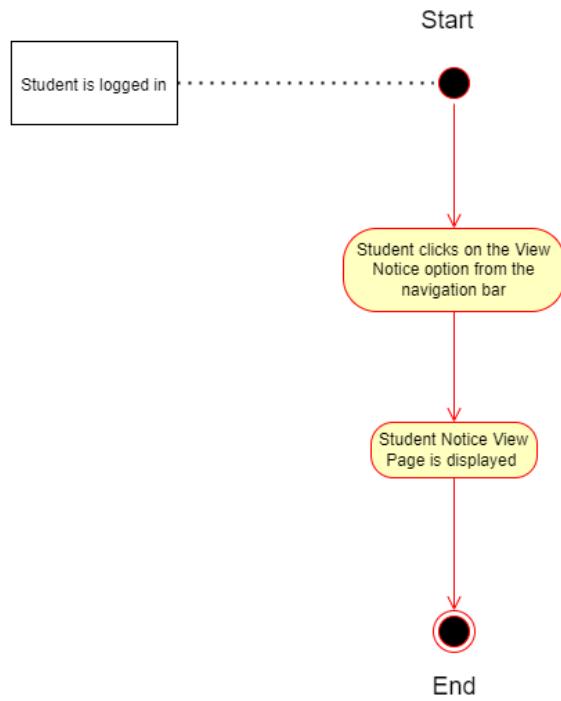
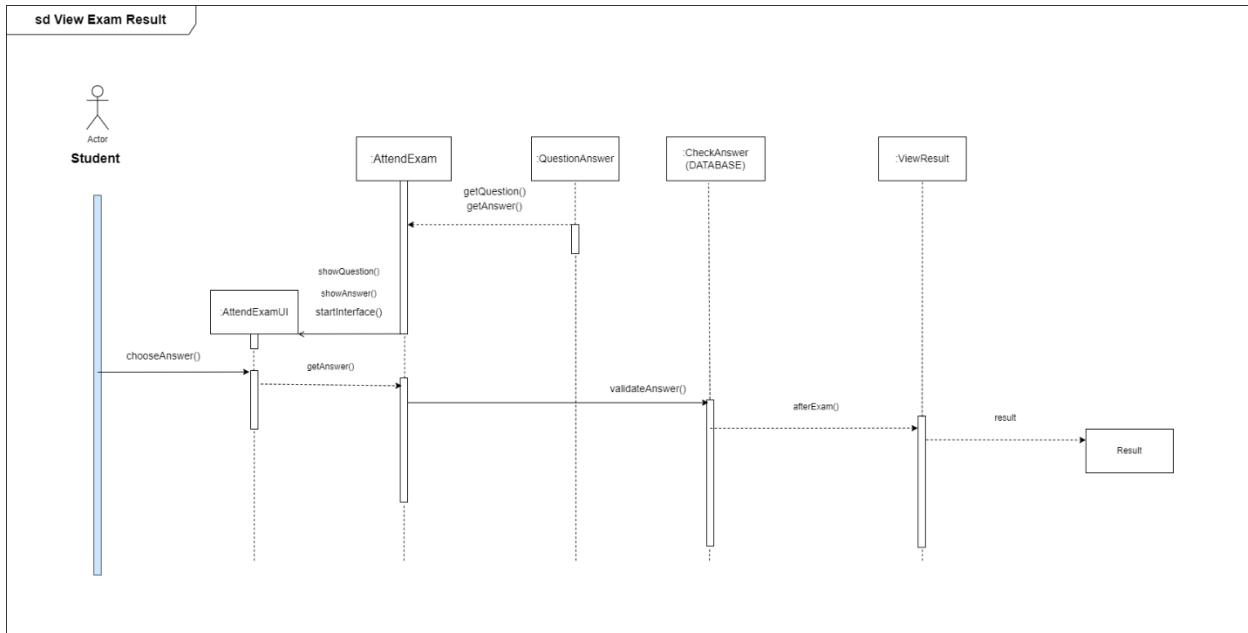


Figure 87 View Notice (Activity Diagram)

3.5.4.5.4. Sequence Diagram:



3.5.4.6. Change Password:

3.5.4.6.1. High level use case:

Use Case:	Change Password.
Actors:	Student
Description:	Student clicks on the Change Password option from settings. Then the password change form is displayed. Student enters details to change password (old password, new password and confirm password). If the entered details are valid, system redirects to change password successful confirmation page.

Table 30 Change password (student) (High level use case)

3.5.4.6.2. Activity Diagram:

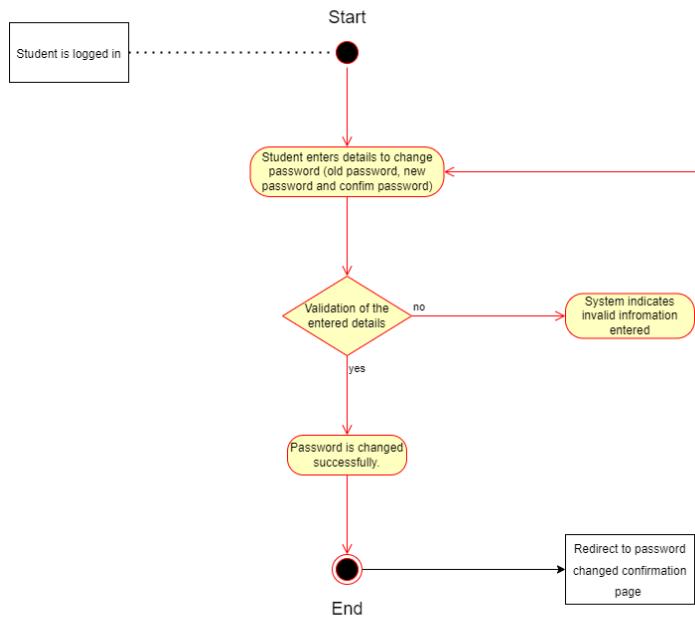


Figure 88 Change password (student) (Activity Diagram)

3.5.4.6.3. Sequence Diagram:

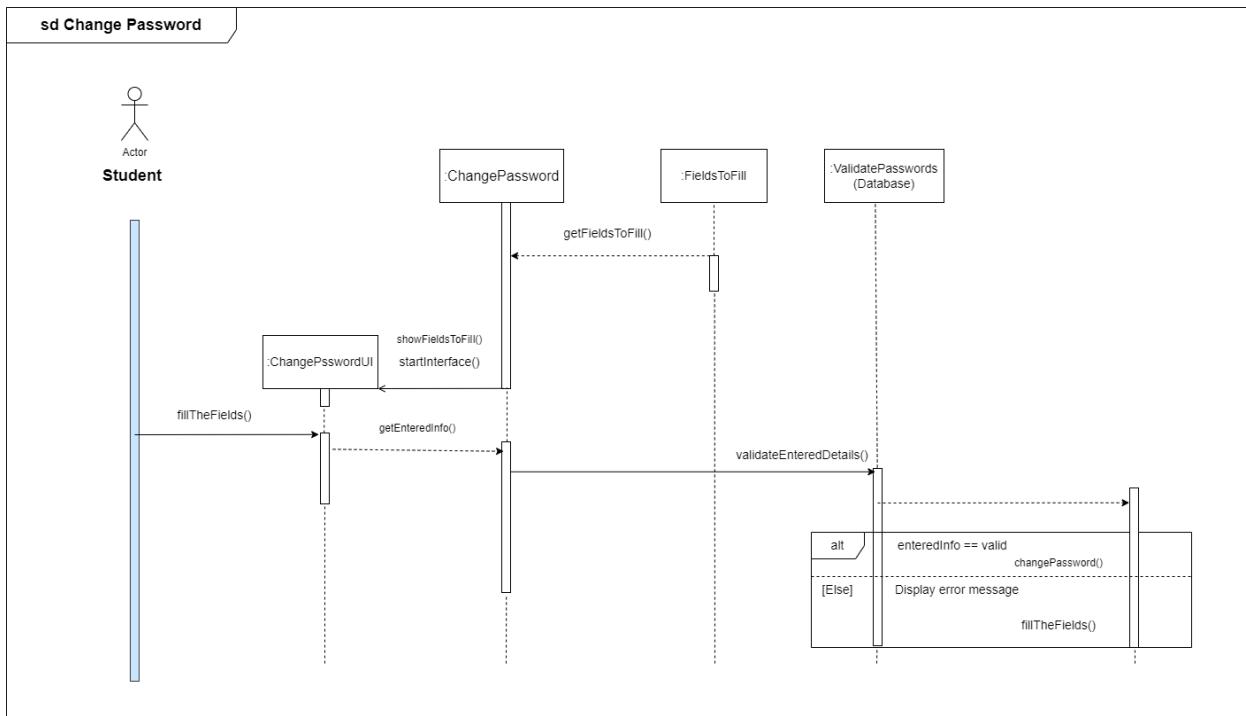


Figure 89 Change password (student) (Sequence Diagram)

3.6. Design:

3.6.1. System Design:

3.6.1.1. Use Case Diagram:

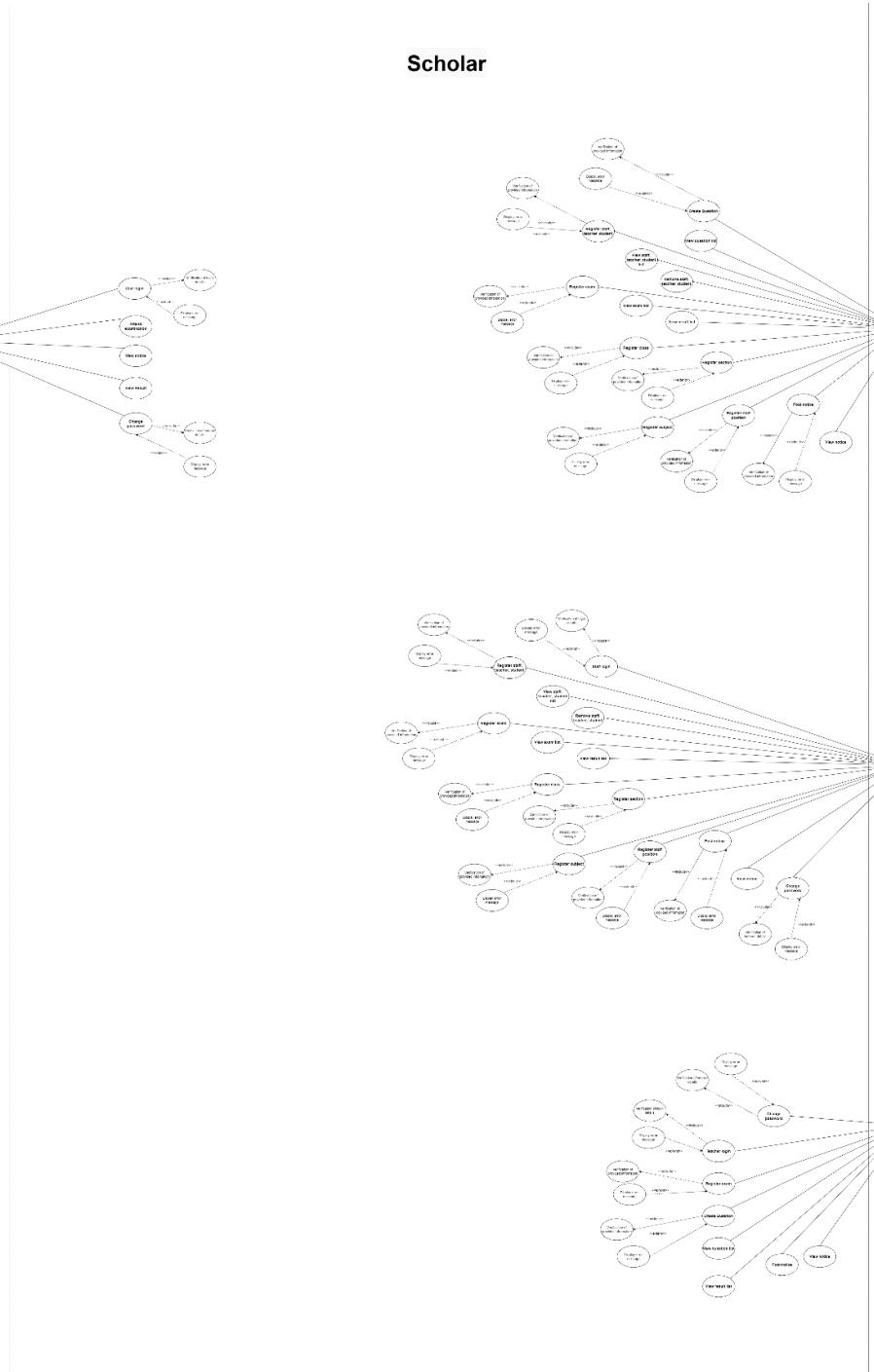


Figure 90 Use Case Diagram

3.6.1.2. System Architecture:

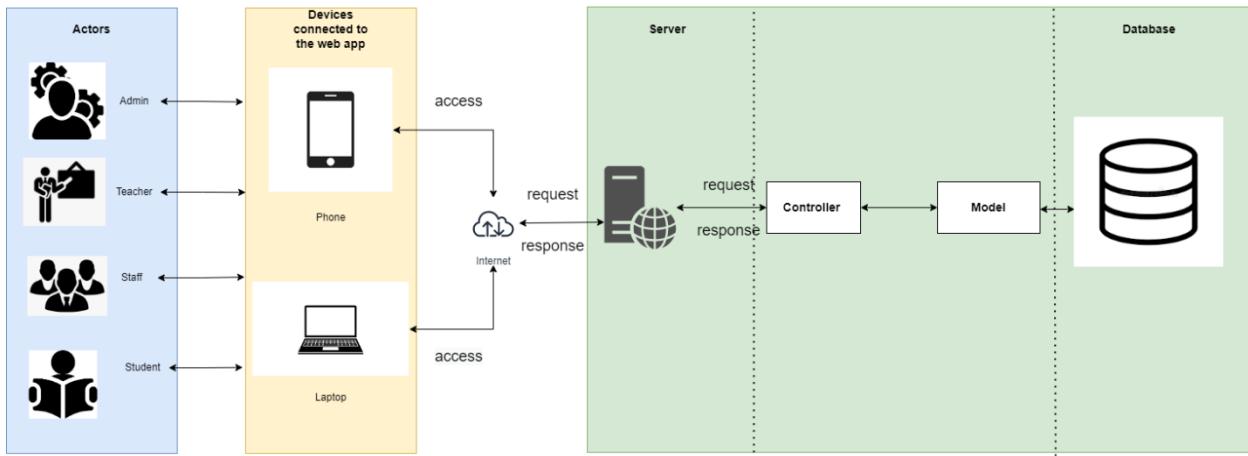


Figure 91 Main system architecture

The actors use the web application from their respective devices which are connected to the internet. The server manages the requests and responses from the system. The controller follows, where it manages the incoming requests as well as the view and database by using the model.

The browser first sends a request to the Controller. The Controller then exchanges data with the Model via communication. To render the data, the Controller next engages with the View. To the View, the information's presentation is what matters; the final presentation is irrelevant. It will be an HTML file that dynamically renders data based on input from the Controller. The Controller will then receive the View's final presentation and communicate the data to the user output (Kumar, 2021).

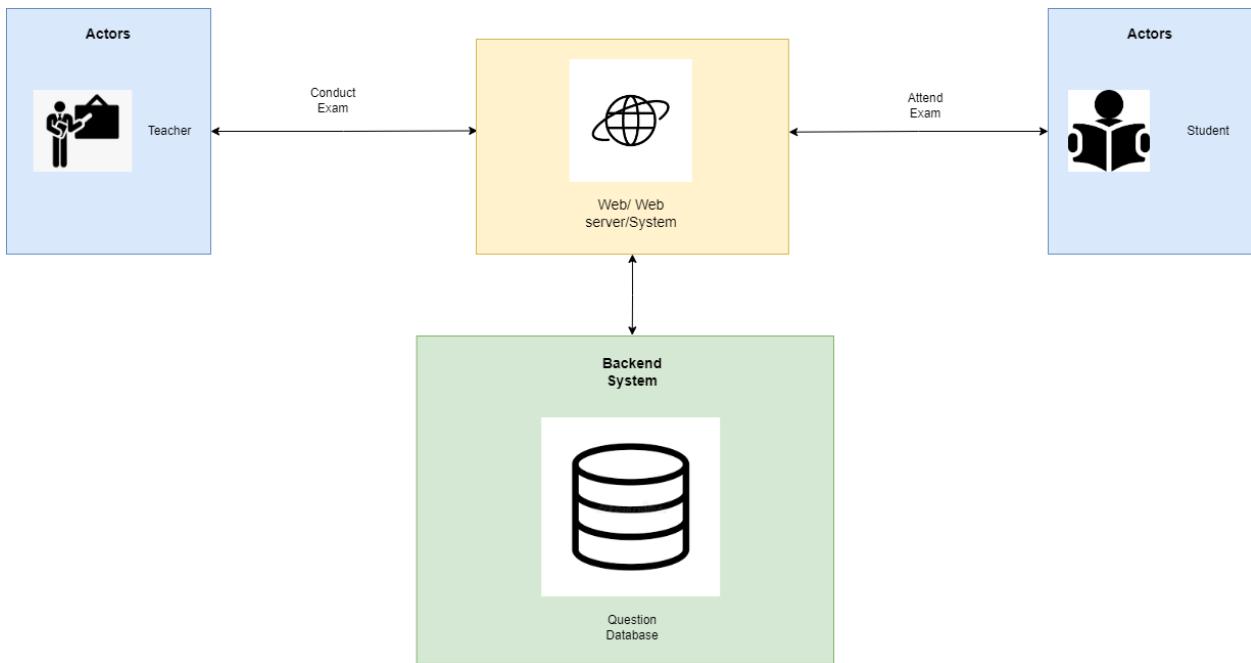


Figure 92 System architecture for examination

This is a basic system architecture for examination where the teacher conducts an exam which is requested by the student. The server manages requests and responses from both actors accordingly. The web server returns responses for the asked pages from the web application asked by the student.

3.6.1.3. ERD:

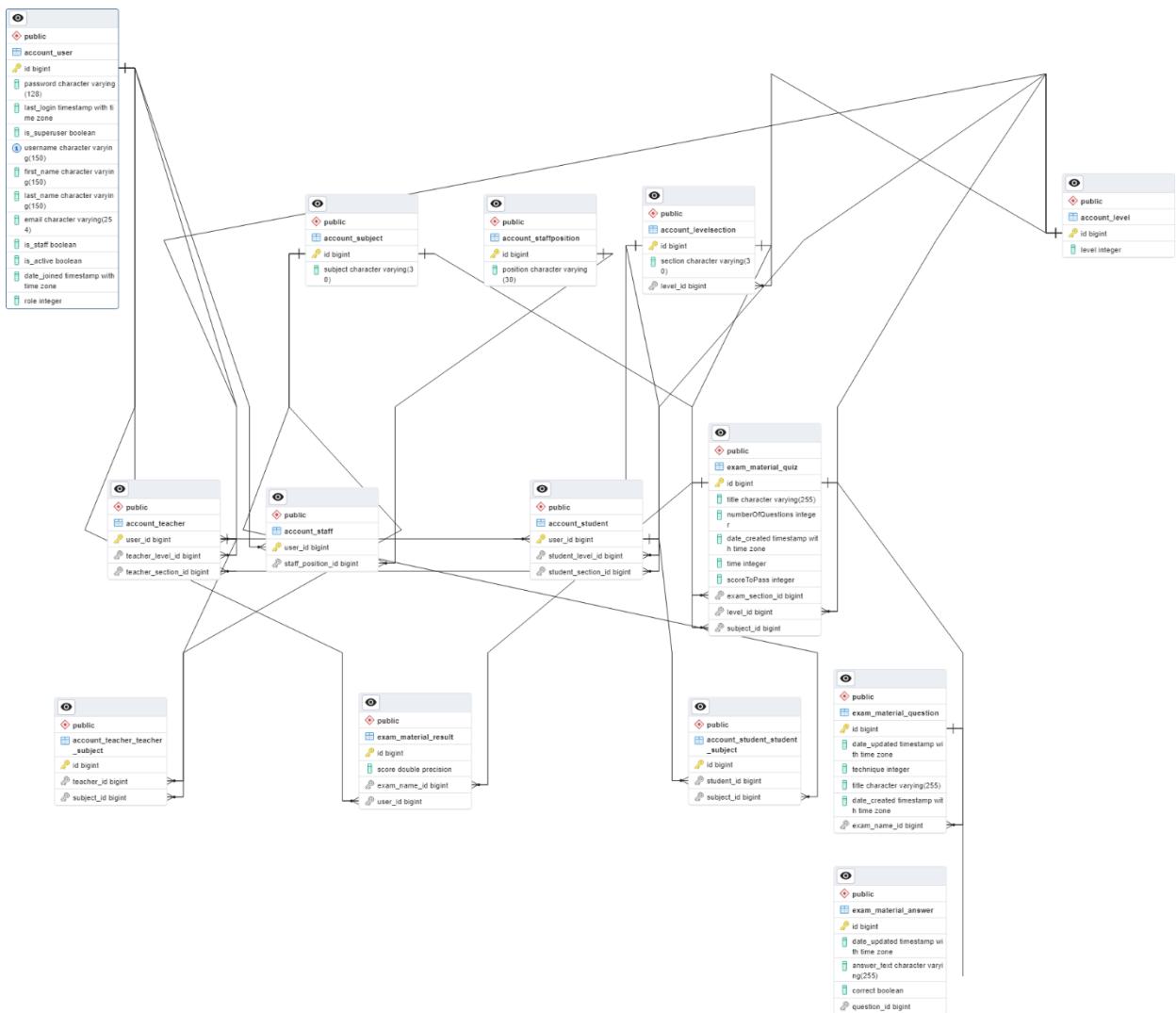


Figure 93 Entity Relationship Diagram

Other system designs and features have already been mentioned in 3.5. Requirement analysis: , including the UML diagrams.

3.7. Implementation:

The code-based implementation part is carried out according to the methodology (evolutionary prototyping) from building a prototype.

Django follows MVT software pattern (Model View Template) for developing web application in it.

**React has been eliminated from the project for the frontend as the task were efficiently carried out with core JavaScript and jQuery.*

*Therefore, core JavaScript and jQuery has been implemented in place of React. **

3.7.1. Register apps:

```
INSTALLED_APPS = [
    'jazzmin',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    #new app
    'account',
    'main',
    'exam_material',
    'rest_framework',
    'django.contrib.admin',
    'django.contrib.auth',
]
```

Figure 94 Register apps

3.7.2. Model Design (Database design):

3.7.2.1. Model for account app:

```
account > ✎ models.py > ...
1   from django.contrib.auth.models import AbstractUser
2
3   from django.db import models
4
5   #for choosing roles(student, teacher, staff, admin)
6   class Role:
7       Admin = 1
8       Student = 2
9       Teacher = 3
10      Staff = 4
11      NotSelected = 5
12
13      role_choices = (
14          (Admin, 'admin'),
15          (Student, 'student'),
16          (Teacher, 'teacher'),
17          (Staff, 'staff'),
18          (NotSelected, 'notSelected')
19      )
20
21
22  class User(AbstractUser):
23
24      # first_name = models.CharField(max_length=30, blank=False, null=False, default=None)
25      # last_name = models.CharField(max_length=30, blank=False, null=False, default=None)
26      # email = models.EmailField(max_length=250, blank=False, null=False, default=None)
27
28      base_role = Role.NotSelected
29
30      role = models.IntegerField('Role', choices=Role.role_choices, null=False, blank=False, default=base_role)
31
32
33      @property
34      def is_student(self):
35          return self.role == Role.Student
```

Figure 95 account app (model)

```

37     @property
38     def is_teacher(self):
39         return self.role == Role.Teacher
40
41     @property
42     def is_staff1(self):
43         return self.role == Role.Staff
44
45     def __str__(self) -> str:
46         return f"[{self.username}]"
47
48
49     class Level(models.Model):
50         level = models.PositiveIntegerField(blank=False, null=False, default=0)
51
52     def __str__(self) -> str:
53         return f"[{self.level}]"
54
55     class Subject(models.Model):
56         subject = models.CharField(max_length=30, blank=True)
57         # Level = models.ForeignKey(Level, on_delete=models.CASCADE)
58
59     def __str__(self) -> str:
60         return self.subject
61
62     class LevelSection(models.Model):
63         level = models.ForeignKey(Level, on_delete=models.CASCADE)
64         section = models.CharField(max_length=30, help_text='Please use capital letters for section.')
65
66     def __str__(self) -> str:
67         return self.section.upper()
68
69     class StaffPosition(models.Model):
70         position = models.CharField(max_length=30)

```

Figure 96 account app (model)(continued1)

```

account > ✘ models.py ...
0 / |   return self.section.upper()
68
69     class StaffPosition(models.Model):
70         position = models.CharField(max_length=30)
71
72     def __str__(self) -> str:
73         return self.position
74
75
76     class Student(models.Model):
77         base_role = Role.Student
78
79         user = models.OneToOneField(User, on_delete=models.CASCADE, primary_key=True, related_name='student')
80         student_subject = models.ManyToManyField(Subject, related_name='student_subject')
81         student_level = models.ForeignKey(Level, on_delete=models.CASCADE, related_name='student_level', null=True)
82         student_section = models.ForeignKey(LevelSection, on_delete=models.CASCADE, related_name='student_level_section', default=0)
83         #role = models.IntegerField(choices=User.Role.role_choices, null=False, blank=False, default=User.Role.Student)
84
85         # @property
86         # def student_level(self):
87         #     return self.student_section.level
88
89     def __str__(self) -> str:
90         return self.user.username
91
92

```

Figure 97 account app (model)(continued 2)

```
account > models.py > ...
90 |     return self.user.username
91 |
92 |
93 |
94 class Teacher(models.Model):
95     base_role = Role.Teacher
96
97     user = models.OneToOneField(User, on_delete=models.CASCADE, primary_key=True, related_name="teacher")
98     teacher_subject = models.ManyToManyField(Subject, related_name='teacher_subject')
99     teacher_level = models.ForeignKey(Level, on_delete=models.CASCADE, related_name='teacher_level', null=True)
100    teacher_section = models.ForeignKey(LevelSection, on_delete=models.CASCADE, related_name='teacher_level_section', default=0)
101    #role = models.IntegerField(choices=Role.role_choices, default=base_role)
102
103    def __str__(self) -> str:
104        return self.user.username
105
106 class Staff(models.Model):
107     base_role = Role.Staff
108
109     user = models.OneToOneField(User, on_delete=models.CASCADE, primary_key=True, related_name='staff')
110     staff_position = models.ForeignKey(StaffPosition, on_delete=models.CASCADE, related_name='staff_position', default='staff')
111     #role = models.IntegerField(choices=Role.role_choices, default=base_role)
112
113     def __str__(self) -> str:
114         return self.user.username
115
116
117
```

Figure 98 account app (model) (continued 3)

3.7.2.2. Model for exam material app:

```
exam_material > 🏷 models.py > ...
1  from django.db import models
2  from account.models import User
3  from account.models import Subject, LevelSection, Level
4  #for translation
5  # from django.utils.translation import gettext_lazy as _
6
7  # class Category(models.Model):
8  #     name = models.CharField(max_length=255)
9
10 #     def __str__(self) -> str:
11 #         return self.name
12
13 SCALE = (
14     ('fundamental', ('Fundamental')),
15     ('beginner', ('Beginner')),
16     ('intermediate', ('Intermediate')),
17     ('advanced', ('Advanced')),
18     ('expert', ('Expert'))
19 )
20
21 #verbose_name allows to name the name of the field in the admin as you want
22 class Quiz(models.Model):
23     title = models.CharField(max_length=255, default='New Exam', verbose_name='Exam Title')
24     subject = models.ForeignKey(Subject, on_delete=models.CASCADE, default='Subject not selected')
25     level = models.ForeignKey(Level, on_delete=models.CASCADE, related_name='exam_level', null=True)
26     exam_section = models.ForeignKey(LevelSection, on_delete=models.CASCADE, related_name='exam_level_section', null=True)
27     numberOfQuestions = models.PositiveIntegerField()
28     date_created = models.DateTimeField(auto_now_add=True)
29     time = models.PositiveIntegerField(help_text="duration of the exam")
30     scoreToPass = models.PositiveIntegerField()
31     # difficulty = models.CharField(max_length=70, choices=SCALE, default='fundamental', verbose_name="Difficulty")
32
33     def __str__(self) -> str:
34         return f"{self.subject}: {self.title}"
```

Figure 99 exam material app (model)

```

exam_material > 🏷 models.py ...
31     # difficulty = models.IntegerField(max_length=70, choices=SCALE, default='fundamental', verbose_name=("Difficulty"))
32
33     def __str__(self) -> str:
34         return f'{self.subject}: {self.title}'
35
36     def get_questions(self):
37         return self.question_set.all()[:self.numberOfQuestions]
38
39     class Meta:
40         verbose_name = ('Quiz')
41         verbose_name_plural = ('Quizzes')
42         ordering = ['id'] #orders according to the id
43
44
45     class Updated(models.Model):
46         date_updated = models.DateTimeField(verbose_name=('Last Updated'), auto_now=True )
47
48         class Meta:
49             abstract = True
50
51
52     class Question(Updated):
53
54         TYPE = (
55             (0, ('Multiple Choice')),
56         )
57
58         exam_name = models.ForeignKey(Quiz, on_delete=models.CASCADE, verbose_name=("Exam name"))
59         technique = models.IntegerField(choices=TYPE, default=0, verbose_name=("Type of Question"))
60         title = models.CharField(max_length=255, verbose_name=("Question"))
61         # difficulty = models.IntegerField(choices=SCALE, default=0, verbose_name=_("Difficulty"))
62         date_created = models.DateTimeField(auto_now_add=True, verbose_name=("Date Created"))
63         # is_active = models.BooleanField(default=False, verbose_name=_("Active Status"))
64
65         def __str__(self):
66             return str(self.title)

```

Figure 100 exam material app (model) (continued 1)

```

exam_material > 🏷 models.py ...
64
65     def __str__(self):
66         return str(self.title)
67
68     def get_answers(self):
69         return self.answer_set.all()
70
71     class Meta:
72         verbose_name = ('Question')
73         verbose_name_plural = ('Questions')
74         ordering = ['id']
75
76
77     class Answer(Updated):
78         question = models.ForeignKey(Question, on_delete=models.CASCADE)
79         answer_text = models.CharField(max_length=255, verbose_name=("Answer"))
80         correct = models.BooleanField(default=False)
81
82         def __str__(self):
83             return f'question: {self.question}, answer: {self.answer_text}, correct: {self.correct}'
84
85     class Meta:
86         verbose_name = ("Answer")
87         verbose_name_plural = ("Answers")
88         ordering = ['id']
89
90
91     class Result(models.Model):
92         user = models.ForeignKey(User, on_delete=models.CASCADE, default=None)
93         exam_name = models.ForeignKey(Quiz, on_delete=models.CASCADE)
94         score = models.FloatField()
95
96
97         def __str__(self) -> str:
98             return str(self.user)

```

Figure 101 exam material app (model) (continued 2)

3.7.2.3. Model for main app:

```
main > models.py > ...
1  from django.db import models
2
3
4
5  class Forum(models.Model):
6      name=models.CharField(max_length=200,default="anonymous" )
7      email=models.CharField(max_length=200,null=True)
8      topic= models.CharField(max_length=300)
9      description = models.CharField(max_length=1000,blank=True)
10     date_created=models.DateTimeField(auto_now_add=True,null=True)
11
12     def __str__(self):
13         return str(self.topic)
14
15
16  class Discussion(models.Model):
17      forum = models.ForeignKey(Forum,blank=True,on_delete=models.CASCADE)
18      discuss = models.CharField(max_length=1000)
19
20      def __str__(self):
21          return str(self.forum)
22
23
```

Figure 102 main app (model)

3.7.3. Views:

3.7.3.1. View for account app:

```
account > views.py > TeacherRegistration
54
55 #teacher registration
56 class TeacherRegistration(generic.CreateView):
57     template_name = 'account/teacher_registration_form.html'
58     form_class = TeacherRegistrationForm
59     model = Teacher
60     success_url = reverse_lazy('account:teacher-registration')
61
62     def form_valid(self, form) -> HttpResponseRedirect:
63         self.object=form.save()
64         user = self.object.user
65         # username = form.cleaned_data.get('username')
66         password = form.cleaned_data.get('password1')
67
68         subjects = list(self.object.teacher_subject.values_list('subject', flat=True))
69
70
71         subject = f'Login credentials, {user.get_role_display()}.'
72         messageFromEmail = f'Hello {user.get_full_name()}. Welcome to Scholar.\nYour login credentials are as follows:\nUsername: {user.username}\n\n'
73         fromEmail = settings.EMAIL_HOST_USER
74         toEmail = [user.email]
75         send_mail(subject, messageFromEmail, fromEmail, toEmail, fail_silently=True)
76
77         messages.success(self.request, f'{user.username} was registered successfully. The login credentials have been delivered to {user.username}.')
78         return HttpResponseRedirect(self.get_success_url())
79
80     def load_level_section(request):
81         level_id = request.GET.get('level_id')
82         sections = LevelSection.objects.filter(level_id=level_id)
83         return render(request, 'account/level_section.html', {'sections': sections})
84
85
86
```

Figure 103 View for account app (Registration)

```
221 #teacher login
222 class TeacherLogin(generic.FormView):
223     template_name = 'account/teacher_login.html'
224     form_class = TeacherLoginForm
225     context_object_name = 'user_teacher_username'
226     success_url = reverse_lazy('main:teacher-page')
227
228     def form_valid(self, form) -> HttpResponseRedirect:
229
230         username = form.cleaned_data.get('username')
231         password = form.cleaned_data.get('password')
232
233         user = authenticate(username=username, password=password)
234
235         if user:
236             if user.is_teacher:
237                 login(self.request, user)
238                 return super().form_valid(form)
239
240         return self.form_invalid(form)
241
242     def form_invalid(self, form):
243         context = self.get_context_data(form=form)
244         context.update({
245             'error': "Either username or password is invalid."
246         })
247         return self.render_to_response(context)
248
```

Figure 104 View for account app (Login)

3.7.3.2. View for exam material app:

```
170 def questionAnswer(request):
171     QuestionAnswerFormSet = modelformset_factory(Question, fields=('exam_name','title'), extra=4)
172
173     if request.method == 'POST':
174         form = QuestionAnswerFormSet(request.POST)
175         # instances = form.save(commit=False)
176
177         # for instance in instances:
178         #     instance.save()
179
180         instances = form.save()
181
182     form = QuestionAnswerFormSet(queryset=Question.objects.none())
183     return render(request, 'exam_material/questionAnswer.html', {'form': form})
184
185
186
187 class QuestionInline():
188     form_class = QuestionForm
189     model = Question
190     template_name = 'exam_material/question_create_or_update.html'
191
```

Figure 105 View for exam material app

3.7.3.4. View for main app:

```
main > views.py > Index
1  from django.views import generic
2  from account.forms import TeacherRegistrationForm
3  from django.urls import reverse_lazy
4  from django.shortcuts import render, redirect
5  from django.views import generic
6  from django.http import HttpResponseRedirect, HttpResponseRedirect
7  from django.contrib import messages
8
9  from account.models import Student, Teacher, Staff
10 from account.forms import StudentRegistrationForm, TeacherRegistrationForm
11 from main.forms import TeacherUpdateForm, StaffUpdateForm, StudentUpdateForm
12
13 from .forms import *
14
15 class Index(generic.TemplateView):
16     template_name = 'main/index.html'
17
18 class StudentPage(generic.TemplateView):
19     template_name = 'main/student_page.html'
20
21 class TeacherPage(generic.TemplateView):
22     template_name = 'main/teacher_page.html'
23
24 class StaffPage(generic.TemplateView):
25     template_name = 'main/staff_page.html'
26
27 # class AfterTeacherLogin(generic.FormView):
28 #     template_name = 'account/teacher_login.html'
29 #     form_class = TeacherRegistrationForm
30 #     context_object_name = 'user_teacher_username'
31 #     success_url = reverse_lazy('main:after_teacher_login')
32
33 def staffNoticeBoard(request):
34     forums=Forum.objects.all()
35     count=forums.count()
36     discussions=
```

Figure 106 View for main ap

3.7.4. Template:

3.7.4.1. Template for account app:

Figure 107 Template for account app

3.7.4.2. Template for exam material app:

```
exam_material > templates > exam_material > question_list.html > div.pcoded-wrapper > div.pcoded-content > div.pcoded-inner-content > div.col-xl-12 > div.card > div.card-block
44          </ol>
45
46      <div class="card-block table-border-style">
47          <div class="table-responsive">
48              <table class="table table-hover">
49                  <thead>
50                      <tr>
51                          <th>S.no</th>
52                          <th>Question</th>
53                          <th>Exam Name</th>
54                          <th>Subject</th>
55                          <th>Update</th>
56                          <th>Delete</th>
57
58                  </tr>
59              </thead>
60              <tbody>
61                  {% for item in questions %}
62                      <tr>
63
64                          <td scope="row">{{ forloop.counter }}</td>
65
66                          <td>
67                              <a href="{% url 'exam_material:question-detail' item.pk %}">{{ item }}</a>
68                          </td>
69
70                          <td>{{ item.exam_name }}</td>
71                          <td>{{ item.exam_name.subject }}</td>
72
73
74                          <td> <a href="{% url 'exam_material:question-update' item.id %}"> update {{ item }} </a>
75                          </td>
76
77                          <td>
78                              <form method="POST" action="{% url 'exam_material:question-delete' item.pk %}">
79                                  {% csrf_token %}
```

Figure 108 Template for exam material app

3.7.4.3. Template for main app:

```

navbar-light > #div.navbar-collapse.navbar-collapse > #ul.navbar-nav.ml-auto > #li > #div.dropdown.drp-user > #div.dropdown-menu.dropdown-menu-right.profile-notification > #ul.pro-body > #li
127   </header>
128   <!-- [ Header ] end -->
129
130   <!-- [ Main Content ] start -->
131   <div class="pcoded-main-container">
132     <div class="pcoded-wrapper">
133       <div class="pcoded-content">
134         <div class="pcoded-inner-content">
135           <!-- [ breadcrumb ] start -->
136
137           <!-- [ breadcrumb ] end -->
138           <div class="main-body">
139             <div class="page-wrapper">
140               <!-- [ Main Content ] start -->
141               <div class="row">
142
143               <!--[ year sales section ] starts-->
144               <div class="col-md-12 col-xl-4">
145                 <div class="card yearly-sales">
146                   <div class="card-block">
147                     <h3 class="mb-4">Teacher Portal</h3>
148                     <h6 class="mb-4">You are using the official teacher portal experience of Scholar
149                     |   <b>{{ user.first_name }} {{ user.last_name }}</b>.</h6>
150
151                   </div>
152                 </div>
153               <!--[ year sales section ] end-->
154
155
156
157
158
159
160               <!--[ Recent Users ] start-->
161               <div class="col-xl-8 col-md-6">
162                 <div class="card Recent-Users">

```

Figure 109 Template for main app

3.7.5. Forms:

3.7.5.1. Forms for account app:

```
account > forms.py > StudentRegistrationForm > save
● 77
78 #teacher registration form
79 class TeacherRegistrationForm(UserCreationForm):
80
81     first_name = forms.CharField(max_length=30, widget=forms.TextInput(attrs={'class':'form-control', 'placeholder': 'First Name'}))
82     last_name = forms.CharField(max_length=30, widget=forms.TextInput(attrs={'class':'form-control', 'placeholder': 'Last Name'}))
83     email = forms.EmailField(widget=forms.TextInput(attrs={'class':'form-control', 'placeholder': 'Email'}))
84
85     subjects = forms.ModelMultipleChoiceField(
86         queryset=Subject.objects.all(),
87         widget=forms.SelectMultiple(attrs={'class': 'form-control'}),
88         required = True
89     )
90
91     level = forms.ModelChoiceField(
92         queryset=Level.objects.all(),
93         widget = forms.Select(attrs={'class':'form-control', 'placeholder': 'Class'}),
94         required = True
95     )
96
97     levelSection = forms.ModelChoiceField(
98         queryset=LevelSection.objects.all(),
99         widget = forms.Select(attrs={'class':'form-control', 'placeholder': 'Section'}),
100        required = True
101    )
102
103     username = forms.CharField(widget=forms.TextInput(attrs={'class':'form-control', 'placeholder': 'Username'}))
104     password1 = forms.CharField(widget=forms.PasswordInput(attrs={'class':'form-control', 'placeholder': 'Password'}))
105     password2 = forms.CharField(widget=forms.PasswordInput(attrs={'class':'form-control', 'placeholder': 'Password Confirmation'}))
106
107     role = forms.CharField(label='Role',
108         widget=forms.Select(choices=Role.role_choices, attrs={'class':'form-control', 'placeholder': 'Role'}), disabled=True, initial=Role.Teacher)
109
110 class Meta(UserCreationForm.Meta):
111     model = User
112     fields = ['first_name', 'last_name', 'email', 'username', 'password1', 'password2', 'role', 'subjects', 'level', 'levelSection']
```

Figure 110 Form for account app

3.7.5.2. Forms for exam material app:

```
exam_material > forms.py > ExamRegistrationForm
57     #     widget = forms.Select(attrs={ 'class' : 'form-control', 'placeholder': 'type of question '}),
58     # required = True
59   # )
60   title = forms.CharField(max_length=30, widget=forms.TextInput(attrs={'class':'form-control', 'placeholder': 'Question'}))
61 class Meta:
62     model = Question
63     fields = '__all__'
64     widgets = {
65         'technique': forms.ModelChoiceField(
66             queryset=Question.objects.all(),
67             # attrs={
68             #     'class': 'form-control'
69             # },
70             ),
71     }
72
73 class AnswerForm(forms.ModelForm):
74     class Meta:
75         model = Answer
76         fields = '__all__'
77         widgets = {
78             'answer_text': forms.TextInput(
79                 attrs={
80                     'class': 'form-control'
81                 }
82             ),
83             'correct': forms.TextInput(
84                 attrs={
85                     'class': 'form-control'
86                 }
87             ),
88         }
89
90     AnswerFormSet = inlineformset_factory(
91         Question, Answer, form = AnswerForm, extra = 1, can_delete=True, can_delete_extra=True
92 )
```

Figure 111 Forms for exam material app

3.7.5.3. Forms for main app:

```
main > forms.py > ...
33
34
35
36 #teacher update form
37 class TeacherUpdateForm(ModelForm):
38
39     first_name = forms.CharField(max_length=30, widget=forms.TextInput(attrs={'class':'form-control', 'placeholder': 'First Name'}))
40     last_name = forms.CharField(max_length=30, widget=forms.TextInput(attrs={'class':'form-control', 'placeholder': 'Last Name'}))
41     email = forms.EmailField(widget=forms.TextInput(attrs={'class':'form-control', 'placeholder': 'Email'}))
42     # username = forms.CharField(widget=forms.TextInput(attrs={'class': 'form-control', 'placeholder': 'Username'}))
43
44
45     role = forms.CharField(label='Role',
46         widget=forms.Select(choices=Role.role_choices, attrs={'class': 'form-control', 'placeholder': 'Role'}), disabled=True, initial=Role.Teacher)
47
48     class Meta:
49         model = User
50         fields = [ 'first_name','last_name','email','role']
51
52
53 class StudentUpdateForm(ModelForm):
54
55     first_name = forms.CharField(max_length=30, widget=forms.TextInput(attrs={'class':'form-control', 'placeholder': 'First Name'}))
56     last_name = forms.CharField(max_length=30, widget=forms.TextInput(attrs={'class':'form-control', 'placeholder': 'Last Name'}))
57     email = forms.EmailField(widget=forms.TextInput(attrs={'class': 'form-control', 'placeholder': 'Email'}))
58     # username = forms.CharField(widget=forms.TextInput(attrs={'class': 'form-control', 'placeholder': 'Username'}))
59
60
61     role = forms.CharField(label='Role',
62         widget=forms.Select(choices=Role.role_choices, attrs={'class': 'form-control', 'placeholder': 'Role'}), disabled=True, initial=Role.Student)
63
64     class Meta:
65         model = User
66         fields = [ 'first_name','last_name','email','role']
```

Figure 112 Forms for main app

3.7.6. URL:

```
exam > ⚡ urls.py > ...
11 |     2. Add a URL to urlpatterns: path('', Home.as_view(), name='home')
12 | Including another URLconf
13 |     1. Import the include() function: from django.urls import include, path
14 |     2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))
15 |
16 from django.conf.urls import url
17 from django.contrib.auth import views as auth_views
18 from django.contrib import admin
19 from django.urls import path,include
20
21 urlpatterns = [
22     path('admin/', admin.site.urls),
23     path('account/', include('account.urls')),
24     path('exam_material/', include('exam_material.urls')),
25     path('', include('main.urls')),
26
27     #for forgot password
28     path('password_reset/', auth_views.PasswordResetView.as_view(), name='password_reset'),
29     path('password_reset_done/', auth_views.PasswordResetDoneView.as_view(), name='password_reset_done'),
30     path('reset///', auth_views.PasswordResetConfirmView.as_view(), name='password_reset_confirm'),
31     path('reset/done/', auth_views.PasswordResetCompleteView.as_view(template_name='main/password_reset_complete.html'), name='password_reset_complete')
32     # url(r'^', include('django.contrib.auth.urls'))
33
34     #for change password
35     path('password_change/', auth_views.PasswordChangeView.as_view(template_name='main/password_change.html'), name='password_change'),
36     path('password_change_done/', auth_views.PasswordChangeDoneView.as_view(template_name='main/password_change_done.html'), name='password_change_done')
37 ]
38
```

Figure 113 Main URL of the project

3.7.6.1. URL (account app):

```
account > ⚡ urls.py > ...
1 from django.urls import path,include
2 from django.contrib.auth import views as auth_views
3 from django.conf.urls import url
4 from .views import AfterTeacherLogin,StaffList, StaffLogin, StaffRegistration, StudentLogin, SubjectRegistration, TeacherLogin, TeacherRegistration,
5
6 app_name = 'account'
7
8 urlpatterns = [
9     path('student-registration/', StudentRegistration.as_view(), name='student-registration'),
10    path('teacher-registration/', TeacherRegistration.as_view(), name='teacher-registration'),
11    path('staff-registration/', StaffRegistration.as_view(), name='staff-registration'),
12    path('student-list/', StudentList.as_view(), name='student-list'),
13    path('teacher-list/', TeacherList.as_view(), name='teacher-list'),
14    path('staff-list/', StaffList.as_view(), name='staff-list'),
15    path('subject-registration/', SubjectRegistration.as_view(), name='subject-registration'),
16    path('student-login/', StudentLogin.as_view(), name='student-login'),
17    path('teacher-login/', TeacherLogin.as_view(), name='teacher-login'),
18    path('staff-login/', StaffLogin.as_view(), name='staff-login'),
19    path('after-teacher-login/', AfterTeacherLogin.as_view(), name='after-teacher-login'),
20    path('level-registration/', LevelRegistration.as_view(), name='level-registration'),
21    path('level-section-registration/', LevelSectionRegistration.as_view(), name='level-section-registration'),
22    path('staff-position-registration/', StaffPositionRegistration.as_view(), name='staff-position-registration'),
23    path('user_logout/', UserLogOut.as_view(), name='user_logout'),
24
25
26    # url('^', include('django.contrib.auth.urls')),
27
28    path('load-section/', load_level_section, name='load-section'),
29    # path('multiple-subjects/', multiple_subjects, name='multiple-subjects'),
30
31    # path('Load-subject/', Load_level_subject, name='Load-subject'),
32 ]
```

Figure 114 URL (account app)

3.7.6.2. URL (exam material app):

```
exam_material > ✎ urls.py > ...
1  from django.urls import path, include
2  from .views import QuizListView, quiz_view, quiz_data_view, save_quiz_view, QuestionList, QuestionCreate, QuestionUpdate, delete_answer, ExamRegistrationStaff, ExamRegistrationTeacher, ExamList, ExamDetail, ExamUpdate, ExamDelete, questionAnswer, ResultListTeacher, ResultListStaff
3
4  app_name = 'exam_material'
5
6  urlpatterns = [
7      # path('quiz-api/', QuizAPI.as_view(), name='quiz-api'),
8      # path('r/<str:topic>', RandomQuestion.as_view(), name='random'),
9      # path('question/<str:topic>', QuizQuestion.as_view(), name='question-api'),
10     # path('answer/<str:topic>', QuizAnswer.as_view(), name='answer-api'),
11
12     path('quiz-main/', QuizListView.as_view(), name='quiz-main'),
13     path('quiz-main/<pk>', quiz_view, name='quiz-view'),
14     path('quiz-main/<pk>/save/', save_quiz_view, name='save-quiz-view'),
15     path('quiz-main/<pk>/data/', quiz_data_view, name='quiz-data-view'),
16
17     path('question-create/', QuestionCreate.as_view(), name='question-create'),
18     path('question-list/', QuestionList.as_view(), name='question-list'),
19     path('question-detail/<int:pk>', QuestionDetail.as_view(), name='question-detail'),
20     path('question-update/<int:pk>', QuestionUpdate.as_view(), name='question-update'),
21     path('question-delete/<int:pk>', QuestionDelete.as_view(), name='question-delete'),
22
23     path('answer-delete/<int:pk>', delete_answer, name='answer-delete'),
24
25     path('exam-registration-staff/', ExamRegistrationStaff.as_view(), name='exam-registration-staff'),
26     path('exam-registration-teacher/', ExamRegistrationTeacher.as_view(), name='exam-registration-teacher'),
27     path('exam-list/', ExamList.as_view(), name='exam-list'),
28     path('exam-detail/<int:pk>', ExamDetail.as_view(), name='exam-detail'),
29     path('exam-update/<int:pk>', ExamUpdate.as_view(), name='exam-update'),
30     path('exam-delete/<int:pk>', ExamDelete.as_view(), name='exam-delete'),
31
32     path('question-answer/', questionAnswer, name='question-answer'),
33
34     path('exam-result-list-teacher/', ResultListTeacher.as_view(), name='exam-result-list-teacher'),
35     path('exam-result-list-staff/', ResultListStaff.as_view(), name='exam-result-list-staff'),
36 ]
```

Figure 115 URL (exam material app)

3.7.6.3. URL (main app):

```
main > ✘ urls.py > ...
  1  from django.urls import path
  2  from django.contrib.auth import views as auth_views
  3  from .views import Index, StudentPage, TeacherPage, StaffPage, staffNoticeBoard, commentNotice, staffCreateNotice, studentNoticeBoard, staffNoticeVi
  4
  5  app_name = 'main'
  6
  7  urlpatterns = [
  8      #pages
  9      path('', Index.as_view(), name='index'),
10      path('student-page/', StudentPage.as_view(), name='student-page'),
11      path('teacher-page/', TeacherPage.as_view(), name='teacher-page'),
12      path('staff-page/', StaffPage.as_view(), name='staff-page'),
13
14      # Notice Board
15      path('staff-notice-board/', staffNoticeBoard, name='staff-notice-board'),
16      path('student-notice-board/', studentNoticeBoard, name='student-notice-board'),
17      path('staff-notice-view/', staffNoticeView, name='staff-notice-view'),
18      path('teacher-notice-view/', teacherNoticeView, name='teacher-notice-view'),
19      path('staff-create-notice/', staffCreateNotice, name='staff-create-notice'),
20      path('teacher-create-notice/', teacherCreateNotice, name='teacher-create-notice'),
21      path('comment-notcie/', commentNotice, name='comment-notice'),
22      path('teacher-notice-board/', teacherNoticeBoard, name='teacher-notice-board'),
23
24      #Student account crud
25      path('student-list/', StudentList.as_view(), name='student-list'),
26      path('student-detail/<int:pk>', StudentDetail.as_view(), name='student-detail'),
27      path('student-update/<int:pk>', StudentUpdate.as_view(), name='student-update'),
28      path('studnet-delete/<int:pk>', StudentDelete.as_view(), name='student-delete'),
29
30      #Teacher account crud
31      path('teacher-list/', TeacherList.as_view(), name='teacher-list'),
32      path('teacher-detail/<int:pk>', TeacherDetail.as_view(), name='teacher-detail'),
33      path('teacher-update/<int:pk>', TeacherUpdate.as_view(), name='teacher-update'),
34      path('teacher-delete/<int:pk>', TeacherDelete.as_view(), name='teacher-delete'),
35
```

Figure 116 URL (main app)

```

main > urls.py > ...
14     # Notice Board
15     path('staff-notice-board/', staffNoticeBoard, name='staff-notice-board'),
16     path('student-notice-board/', studentNoticeBoard, name='student-notice-board'),
17     path('staff-notice-view/', staffNoticeView, name='staff-notice-view'),
18     path('teacher-notice-view/', teacherNoticeView, name='teacher-notice-view'),
19     path('staff-create-notice/', staffCreateNotice, name='staff-create-notice'),
20     path('teacher-create-notice/', teacherCreateNotice, name='teacher-create-notice'),
21     path('comment-notice/', commentNotice, name='comment-notice'),
22     path('teacher-notice-board/', teacherNoticeBoard, name='teacher-notice-board'),
23
24     #Student account crud
25     path('student-list/', StudentList.as_view(), name='student-list'),
26     path('student-detail/int:pk//', StudentDetail.as_view(), name='student-detail'),
27     path('student-update/int:pk//', StudentUpdate.as_view(), name='student-update'),
28     path('student-delete/int:pk//', StudentDelete.as_view(), name='student-delete'),
29
30     #Teacher account crud
31     path('teacher-list/', TeacherList.as_view(), name='teacher-list'),
32     path('teacher-detail/int:pk//', TeacherDetail.as_view(), name='teacher-detail'),
33     path('teacher-update/int:pk//', TeacherUpdate.as_view(), name='teacher-update'),
34     path('teacher-delete/int:pk//', TeacherDelete.as_view(), name='teacher-delete'),
35
36     #Staff account crud
37     path('staff-list/', StaffList.as_view(), name='staff-list'),
38     path('staff-detail/int:pk//', StaffDetail.as_view(), name='staff-detail'),
39     path('staff-update/int:pk//', StaffUpdate.as_view(), name='staff-update'),
40     path('staff-delete/int:pk//', StaffDelete.as_view(), name='staff-delete'),
41
42
43     #path('after-teacher-Login/', AfterTeacherLogin.as_view(), name='after-teacher-login'),
44 ]

```

Figure 117 URL (main app) (continued)

4. Testing and Analysis:

4.1. Test Plan:

Software testing is a technique for determining whether the actual piece of software complies with specifications and is fault-free. It entails manually or automatically executing software or system components in order to evaluate one or more noteworthy characteristics. The purpose of software testing is to identify errors, gaps, or unmet requirements in comparison to the written specifications. To describe the definition of software testing, some people prefer to use the words White Box and Black Box Testing (Hamilton, 2023).

4.1.1. White Box Testing, Unit Testing:

According to the definition, "white box testing" also known as clear, glass or structural testing is a testing technique that evaluates the code and internal organization of the program. White box testing comprises analysing the structure of the code. Once the internal structure of a product is identified, testing can be done to ensure that its internal processes operate in line with standards. Additionally, the appropriate workout has been given to each internal component (Help, 2023).

4.1.2. Black Box Testing, System Testing:

Black box testing is a technique for assessing the performance of software applications without having access to the implementation specifics, internal communication channels, or the underlying code. Black Box Testing is entirely based on software requirements and standards and concentrates on the input and output of software programs commonly referred to as behavioural research (Hamilton, 2023).

4.2. Unit Testing:

4.2.1. Test case: Create Admin (Superuser)

Test No:	
Objective:	To create admin (superuser).
Action:	i) Admin creation command is given in the terminal> ii) Admin username is given. iii) Admin password is given.
Expected result:	Admin would be created.
Actual result:	Admin was created.
Conclusion:	The test is successful.

Table 31 To create admin (superuser).

```
(venv) C:\Users\lillo\Desktop\n_django\exam>python manage.py createsuperuser
Username: admin
Email address:
Password:
Password (again):
The password is too similar to the username.
This password is too short. It must contain at least 8 characters.
This password is too common.
Bypass password validation and create user anyway? [y/N]: y
Superuser created successfully.
```

Figure 118 To create admin (superuser).

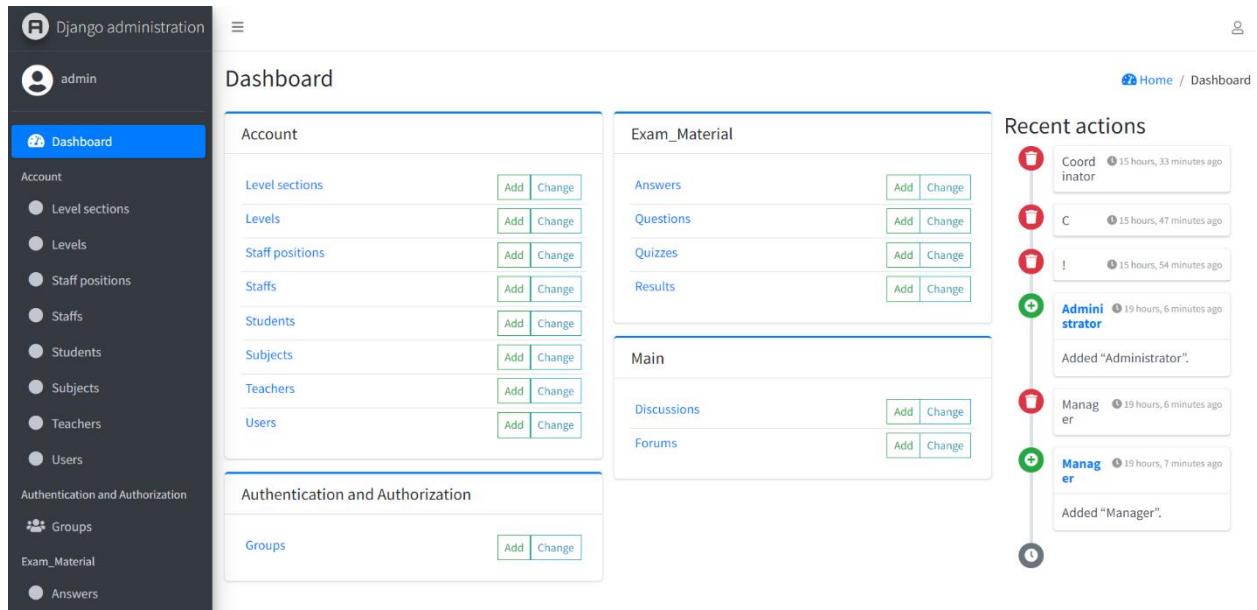


Figure 119 To create admin (superuser). (Test successful)

4.2.2. Test Case: To test User Registration:

Test No: 2	
Objective:	To test user registration (valid user information is given).
Action:	i) Valid user information is given.
Expected result:	i) The user would be registered successfully(teacher). ii) Registered user would be shown in database.
Actual result:	i) The user was registered successfully(teacher). ii) The registered user was shown in database.
Conclusion:	The test is successful.

Table 32 To test user registration (valid user information is given).

john was registered successfully. The login credentials have been delivered to john.

Scholar

Teacher Registration

First Name	Role
<input type="text"/>	teacher
Last Name	Class
<input type="text"/>	-----
Email	Section
<input type="text"/>	-----
Username	Subjects('ctrl' and 'click' to select multiple subjects)
<input type="text"/>	Maths English
Password	
Password Confirmation	

Register

Get back to staff page: [Staff Page](#)

Figure 120 To test user registration.

The screenshot shows the pgAdmin 4 interface with a query editor window. The query is:

```
1 SELECT * FROM public.account_user
2 ORDER BY id ASC
```

The results table displays the following data:

	public.account_user/exam/postgres@PostgreSQL 15					
				No limit		
	Query	Query History			Scratch Pad	
	1	SELECT * FROM public.account_user				
	2	ORDER BY id ASC				
	i	password	last_login	is_superuser	username	
	PK	bigint	character varying (128)	timestamp with time zone	boolean	character varying (150)
1	1	pbkdf2_sha256\$260000\$GbeTxpstMLza89ttBsBhr\$nymHlZe55KL1kMz+Lpn0NEYssXJWaYY247Sy2t9...	2023-04-17 19:23:11.253208+05:45	true	admin	
2	3	pbkdf2_sha256\$260000\$ZabLATTpjzTgQlulqRS01V\$vi/U7+rUDhl2406/07mDEVQoyPjf1v+f+Ljhy2h=	2023-04-17 05:55:09.785924+05:45	false	eddie	
3	4	pbkdf2_sha256\$260000\$PKepRRFA0xNQps8CY3Qad8\$T4wCS+5tB+v/VCHVSc9WA8BzXIDZCwZePuTo...	2023-04-17 05:00:50.789352+05:45	false	sayush	
4	5	pbkdf2_sha256\$260000\$kyY201PfXGytx61T9dBTbH\$5sVvwHnC9rVMPYJL10LV4KMVmtpbXLZ27P2r...	2023-04-17 04:07:27.412274+05:45	false	john	
5	7	pbkdf2_sha256\$260000\$45bYXlJogemLfirUDmF6F0\$/brUV8Y+0M ZwZXRoQgR0VgqYwVoTe2HL325r...	2023-04-17 05:54:46.556307+05:45	false	messi	
6	8	pbkdf2_sha256\$260000\$hsKQ8TzYEeAOTwal07qHcl\$fPLd+x/efFATPCXijotu0/p3tUMQ6Qtfc06LsVcZl...	2023-04-17 05:54:24.269349+05:45	false	jimmy	

Figure 121 To test user registration. (Test successful)

The screenshot shows the pgAdmin 4 interface. At the top, there's a toolbar with various icons for database management. Below the toolbar, the title bar displays the connection information: public.account_teacher/exam/postgres@PostgreSQL 15. The main area is divided into two panes: 'Query' on the left and 'Scratch Pad' on the right. The 'Query' pane contains the following SQL code:

```
1 SELECT * FROM public.account_teacher
2 ORDER BY user_id ASC
```

The 'Data Output' tab is selected at the bottom, showing the results of the query in a grid format:

	user_id [PK] bigint	teacher_level_id bigint	teacher_section_id bigint
1	5	2	2

Figure 122 To test user registration (continued). (Test successful)

4.2.3. Test Case: View User List in Database:

Test No: 3	
Objective:	To view user list in database.
Action:	Select query to view user is given.
Expected result:	List of existing users would be shown in database.
Actual result:	List of existing users was shown in database.
Conclusion:	The test is successful.

Table 33 To view user list in database.

The screenshot shows the pgAdmin 4 interface. At the top, there's a toolbar with various icons for file operations, search, and preferences. Below the toolbar, the title bar displays the connection information: public.account_user/exam/postgres@PostgreSQL 15. The main area has two tabs: 'Query' and 'Scratch Pad'. The 'Query' tab contains the following SQL code:

```

1 SELECT * FROM public.account_user
2 ORDER BY id ASC

```

The 'Data Output' tab is selected, displaying the results of the query as a table. The table has columns: id [PK] bigint, password character varying (128), last_login timestamp with time zone, is_superuser boolean, and username character varying (150). There are six rows of data, each representing a user account.

	id [PK] bigint	password character varying (128)	last_login timestamp with time zone	is_superuser boolean	username character varying (150)
1	1	pbkdf2_sha256\$260000\$GbeTxpstIMLza89ttBsBhr\$myMHZeS5KL1kMz+LpnoNEYssXJWaYY247Sy2t9...	2023-04-17 19:23:11.253208+05:45	true	admin
2	3	pbkdf2_sha256\$260000\$ZabLATTpjzTgQlulqRS01V\$vi/U7+I7rUDhl2406/07mDEVQoyPjfjp1vf+Ljvy2hl=	2023-04-17 05:55:09.785924+05:45	false	eddie
3	4	pbkdf2_sha256\$260000\$PKepRRFA0xNOp8CY3Qad8\$T4wCS+5B+V/VCHVSc9WA8BzXIDZCwZePuTo...	2023-04-17 05:00:50.789352+05:45	false	sayush
4	5	pbkdf2_sha256\$260000\$kyY201fPfXGytx61T9dBT\$5sVvwHnC9rVMPYJL10LV4KMVmtpbXLIZ27P2rp...	2023-04-17 04:07:27.412274+05:45	false	john
5	7	pbkdf2_sha256\$260000\$4SbYXIJogemLfIRUDmF6F0\$/brUV8Y+0MZwZXRoQgROVgyqYwVoTe2HL3Z5r...	2023-04-17 05:54:46.556307+05:45	false	messi
6	8	pbkdf2_sha256\$260000\$hsKQ8TzYEeAOTwal07qHc\$fpLd+x/eFATPCXijotuO/p3tUMQ6QTfcf06LsVcZl...	2023-04-17 05:54:24.269349+05:45	false	jimmy

Figure 123 To view user list in database. (Test successful)

4.2.4. Test Case: Exam Registration:

Test No: 4	
Objective:	To register exam (valid exam information is given).
Action:	i) Valid exam information is given.
Expected result:	i) The exam would be registered successfully. ii) Registered exam would be shown in database.
Actual result:	i) The exam was registered successfully. ii) The registered exam was shown in database.
Conclusion:	The test is successful.

Table 34 Exam Registration.

Exam name: 'First Term' was registered successfully. Subject: 'Maths'

Scholar

Exam Registration

Class

Section

Subject

Exam name

No of Questions(*Cannot be changed later*)

Time (In minutes)

Score To Pass

Figure 124 Exam Registration

The screenshot shows a PostgreSQL database interface with the following details:

- Connection:** public.exam_material_quiz/exam/postgres@PostgreSQL 15
- Query Tab:** SELECT * FROM public.exam_material_quiz ORDER BY id ASC
- Data Output:** A table showing the results of the query:

	id [PK] bigint	title character varying (255)	numberOfQuestions integer	date_created timestamp with time zone	time integer	scoreToPass integer	exam_section_id bigint	level_id bigint	subject_id bigint
1	1	First Term	4	2023-04-17 02:42:35.429683+05:45	1	40	2	2	1
2	2	Class Test 1	5	2023-04-17 03:08:46.501098+05:45	10	45	1	1	2
3	3	Class Test 2	5	2023-04-17 03:09:20.837624+05:45	15	45	3	1	1

Figure 125 Exam Registration (Test successful)

4.2.5. Test Case: View Exam Result List in Database:

Test No: 5	
Objective:	To view exam result list in database.
Action:	Select query to view exam list.
Expected result:	List of existing results of the exams given by the students would be shown.
Actual result:	List of existing results of the exams given by the students was shown.
Conclusion:	The test is successful.

Table 35 To view exam result list in database.

The screenshot shows the pgAdmin 4 interface. At the top, there's a toolbar with various icons for database management. Below the toolbar, the title bar indicates the connection is to 'public.exam_material_result/exam/postgres@PostgreSQL 15'. The main area has two tabs: 'Query' (selected) and 'Scratch Pad'. The 'Query' tab contains the following SQL code:

```
1 SELECT * FROM public.exam_material_result
2 ORDER BY id ASC
```

Below the code, the results are displayed in a table format. The table has four columns: 'id' (PK bigint), 'score' (double precision), 'exam_name_id' (bigint), and 'user_id' (bigint). There are 9 rows of data:

	id [PK] bigint	score double precision	exam_name_id bigint	user_id bigint
1	1	0	1	4
2	2	50	1	4
3	3	50	1	4
4	4	25	1	4
5	5	25	1	4
6	6	25	1	4
7	7	0	1	4
8	8	50	1	8
9	9	75	1	7

Figure 126 To view exam result list in database. (Test successful)

4.2.6. Test Case: Create Question (Question to be registered in Database):

Test No: 6	
Objective:	To create question (question to be registered in database).
Action:	<ul style="list-style-type: none">i) Question name is given.ii) Answer to the question is provided.iii) Question List is viewed in database.
Expected result:	<ul style="list-style-type: none">i) The question would be created successfully.ii) The question would be shown in database.
Actual result:	<ul style="list-style-type: none">i) The question was created successfully.ii) The question was shown in database.
Conclusion:	The test is successful.

Table 36 To create question (question to be registered in database).

Add Questions

[Return to your portal](#)

Exam name *

Maths: First Term

Type of Question * **Multiple Choice**

Title *

1+1?

Add Answers

Answer *	Correct(Write true for the correct answer) *	Delete?
3	False	<input type="checkbox"/>

Figure 127 To create question (question to be registered in database).

Title *

1+1?

Add Answers

Answer *	Correct(Write true for the correct answer) *	Delete?
3	False	<input type="checkbox"/>
2	true	<input type="checkbox"/>
4	False	<input type="checkbox"/>
5	False	<input type="checkbox"/>

Add answers

Submit

Figure 128 To create question (question to be registered in database). (continued)

The screenshot shows the pgAdmin 4 interface. At the top, there's a toolbar with various icons for database management. Below the toolbar, the title bar displays the connection information: public.exam_material_question/exam/postgres@PostgreSQL 15. The main area has two tabs: 'Query' (selected) and 'Scratch Pad'. The 'Query' tab contains the following SQL code:

```

1 SELECT * FROM public.exam_material_question
2 ORDER BY id ASC

```

Below the code, the results are displayed in a table format. The table has columns: id, date_updated, technique, title, date_created, and exam_name_id. There are four rows of data, each corresponding to one of the five questions listed in the code. The 'date_updated' column shows the timestamp of the last update, and the 'date_created' column shows the timestamp when the question was first added.

	id [PK] bigint	date_updated timestamp with time zone	technique integer	title character varying (255)	date_created timestamp with time zone	exam_name_id bigint
1	1	2023-04-17 04:36:28.771067+05:45	0	1+1?	2023-04-17 04:32:15.98206+05:45	1
2	2	2023-04-17 04:39:11.586072+05:45	0	2+2?	2023-04-17 04:39:11.586072+05:45	1
3	3	2023-04-17 04:39:37.570634+05:45	0	2*2?	2023-04-17 04:39:37.570634+05:45	1
4	5	2023-04-17 05:32:24.521302+05:45	0	5-4?	2023-04-17 05:32:24.521302+05:45	1

Figure 129 To create question (question to be registered in database). (Test successful)

4.2.7. Test Case: Add Answers (Answers to be Registered in Database):

Test No: 7	
Objective:	To create answers (answer to be registered in database).
Action:	<ul style="list-style-type: none">i) Question name is given.ii) Answer to the question is provided.iii) Answer List is viewed in database.
Expected result:	<ul style="list-style-type: none">i) The question would be created successfully.ii) The answer would be shown in database.
Actual result:	<ul style="list-style-type: none">i) The question was created successfully.ii) The answer was shown in database.
Conclusion:	The test is successful.

Table 37 To create answers (answer to be registered in database).

Title *

1+1?

Add Answers

Answer *	Correct(Write true for the correct answer) *	Delete?
3	False	<input type="checkbox"/>
2	true	<input type="checkbox"/>
4	False	<input type="checkbox"/>
5	False	<input type="checkbox"/>

Add answers

Submit

Figure 130 To create answers (answer to be registered in database).

```
public.exam_material_answer/exam/postgres@PostgreSQL 15
```

Query History

```
1 SELECT * FROM public.exam_material_answer
2 ORDER BY id ASC
```

Scratch Pad

Data Output

	id [PK] bigint	date_updated timestamp with time zone	answer_text character varying (255)	correct boolean	question_id bigint
1	2	2023-04-17 04:32:15.988056+05:45	2	true	1
2	3	2023-04-17 04:32:15.988056+05:45	4	false	1
3	4	2023-04-17 04:32:15.989056+05:45	5	false	1
4	5	2023-04-17 04:36:28.772067+05:45	3	false	1
5	6	2023-04-17 04:39:11.589071+05:45	5	false	2
6	7	2023-04-17 04:39:11.591071+05:45	7	false	2
7	8	2023-04-17 04:39:11.592073+05:45	4	true	2
8	9	2023-04-17 04:39:11.593074+05:45	3	false	2
9	10	2023-04-17 04:39:37.572634+05:45	5	false	3
10	11	2023-04-17 04:39:37.574632+05:45	4	true	3

Figure 131 To create answers (answer to be registered in database). (Test successful)

4.2.8. Test Case: Calling ‘set_password’ using other models than Django User Model:

Test No:	8
Objective:	Calling ‘set_password’ using other models than Django User Model.
Action:	‘set_password’ is called for Teacher model.
Expected result:	Traceback would be shown.
Actual result:	Traceback was shown.
Conclusion:	The test is successful.

Table 38 Calling ‘set_password’ using other models than Django User Model.

```
AttributeError at /teacher-update/11/
'Teacher' object has no attribute 'set_password'

Request Method: POST
Request URL: http://127.0.0.1:8000/teacher-update/11/
Django Version: 3.2
Exception Type: AttributeError
Exception Value: 'Teacher' object has no attribute 'set_password'
Exception Location: C:\Users\lillo\Desktop\in_django\exam\venv\lib\site-packages\django\contrib\auth\forms.py, line 129, in save
Python Executable: C:\Users\lillo\Desktop\in_django\exam\venv\Scripts\python.exe
Python Version: 3.9.6
Python Path: ['C:\\Users\\lillo\\Desktop\\in_django\\exam',
 'C:\\Users\\lillo\\AppData\\Local\\Programs\\Python\\Python39\\python39.zip',
 'C:\\Users\\lillo\\AppData\\Local\\Programs\\Python\\Python39\\DLLs',
 'C:\\Users\\lillo\\AppData\\Local\\Programs\\Python\\Python39\\lib',
 'C:\\Users\\lillo\\Desktop\\in_django\\exam\\venv',
 'C:\\Users\\lillo\\Desktop\\in_django\\exam\\venv\\lib\\site-packages']
Server time: Thu, 23 Mar 2023 06:54:31 +0000

Traceback Switch to copy-and-paste view
C:\Users\lillo\Desktop\in_django\exam\venv\lib\site-packages\django\core\handlers\exception.py, line 47, in inner
    47.         response = get_response(request)
   ▶ Local vars
C:\Users\lillo\Desktop\in_django\exam\venv\lib\site-packages\django\core\handlers\base.py, line 181, in _get_response
    181.     response = wrapped_callback(request, *callback_args, **callback_kwargs)
   ▶ Local vars
C:\Users\lillo\Desktop\in_django\exam\venv\lib\site-packages\django\views\generic\base.py, line 70, in view
    70.         return self.dispatch(request, *args, **kwargs)
   ▶ Local vars
C:\Users\lillo\Desktop\in_django\exam\venv\lib\site-packages\django\views\generic\base.py, line 98, in dispatch
```

Figure 132 Calling ‘set_password’ using other models than Django User Model.

4.2.9. Test Case: To enter staff page directly without creating a single user at the very beginning:

Test No: 9	
Objective:	To enter staff page through URL directly without creating a single user at the very beginning.
Action:	Direct URL to get to staff page is entered.
Expected result:	Traceback would be shown.
Actual result:	Traceback was shown.
Conclusion:	The test is successful.

Table 39 To enter staff page through URL directly without creating a single user at the very beginning.

← → C ① 127.0.0.1:8000/staff-page/

NoReverseMatch at /staff-page/

Reverse for 'staff-update' with arguments '(None,)' not found. 1 pattern(s) tried: ['staff\\-update/(?P<pk>[0-9]+)/\$']

Request Method: GET
 Request URL: http://127.0.0.1:8000/staff-page/
 Django Version: 3.2
 Exception Type: NoReverseMatch
 Exception Value: Reverse for 'staff-update' with arguments '(None,)' not found. 1 pattern(s) tried: ['staff\\-update/(?P<pk>[0-9]+)/\$']
 Exception Location: C:\Users\lillo\Desktop\in_django\exam\venv\lib\site-packages\django\urls\resolvers.py, line 694, in _reverse_with_prefix
 Python Executable: C:\Users\lillo\Desktop\in_django\exam\venv\Scripts\python.exe
 Python Version: 3.9.6
 Python Path: ['C:\Users\lillo\Desktop\in_django\exam', 'C:\Users\lillo\AppData\Local\Programs\Python\Python39\python39.zip', 'C:\Users\lillo\AppData\Local\Programs\Python\Python39\DLLs', 'C:\Users\lillo\AppData\Local\Programs\Python\Python39\lib', 'C:\Users\lillo\AppData\Local\Programs\Python\Python39', 'C:\Users\lillo\Desktop\in_django\exam\venv', 'C:\Users\lillo\Desktop\in_django\exam\venv\lib\site-packages']
 Server time: Sun, 16 Apr 2023 18:25:03 +0000

Error during template rendering

In template C:\Users\lillo\Desktop\in_django\exam\main\templates\main\base.html, error at line 0

Reverse for 'staff-update' with arguments '(None,)' not found. 1 pattern(s) tried: ['staff\\-update/(?P<pk>[0-9]+)/\$']

```

1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <title>{% block title %}{% endblock %}</title>
6   <!-- HTML5 Shim and Respond.js IE11 support of HTML5 elements and media queries -->
7   <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
8   <!--[if lt IE 11]>
9     <script src="https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js"></script>
10    <script src="https://oss.maxcdn.com/libs/respond.js/1.4.2/respond.min.js"></script>
```

Traceback [Switch to copy-and-paste view](#)

C:\Users\lillo\Desktop\in_django\exam\venv\lib\site-packages\django\core\handlers\exception.py, line 47, in inner

Figure 133 To enter staff page through URL directly without creating a single user at the very beginning. (Test successful)

4.2.10. Test Case: To add existing field name in Django User Admin:

Test No: 10	
Objective:	To add existing field name in Django User Admin.
Action:	Existing field name is added to user model (user admin).
Expected result:	Error message would be shown
Actual result:	Error message was shown.
Conclusion:	The test is successful.

Table 40 To add existing field name in Django User Admin.

```
6  from .models import User, Student, Subject, Teacher, Staff, Level
7
8 # Register your models here.
9
10 class UserAdmin(UserAdmin):
11     model = User
12
13     fieldsets = (
14         *UserAdmin.fieldsets,
15         ('User role', {'fields': ('role')}),
16         ('Personal info', {'fields': ('first_name', 'last_name', 'email')}),
17     )
18
19
20 admin.site.register(User, UserAdmin)
21 admin.site.register(Student)
22 admin.site.register(Subject)
23 admin.site.register(Level)
24 admin.site.register(Teacher)
25 admin.site.register(Staff)
```

PROBLEMS 18 OUTPUT TERMINAL DEBUG CONSOLE

```
self.check(display_num_errors=True)
File "C:\Users\lillo\Desktop\n_django\exam\venv\lib\site-packages\django\core\management\base.py", line 469, in check
    raise SystemCheckError(msg)
django.core.management.base.SystemCheckError: System check identified some issues:

ERRORS:
<class 'account.admin.UserAdmin'>: (admin.E012) There are duplicate field(s) in 'fieldsets[5][1]'.

System check identified 1 issue (0 silenced).
```

Figure 134 To add existing field name in Django User Admin. (Test successful)

4.3. System Testing:

System Testing carried out has been kept in appendix section [*7.11. System Testing:*](#)

4.4. Critical Analysis:

4.4.1. Test material:

Django and JavaScript were used to construct Scholar, an efficient web-based system for online testing that allows for remote test and assessment administration. The system combines two well-known technologies, the Python-based web framework Django, and the adaptable client-side scripting language JavaScript, to produce a robust and scalable platform that can accommodate the needs of businesses and educational institutions of various sizes. Django's use makes a number of features possible, including user authentication, database management, and administrative capabilities. The test cases were carried out to this project to ensure the proper working of the mentioned functionalities and fixation of bugs and errors if any.

4.4.2. Evaluation of testing:

4.4.2.1. Unit testing:

- i) Unit testing included test case like a user being registered in database and question being registered in the database.
- ii) Unit testing entailed assessing various system components, such as the login page or the question submission form and showing that the data to such particular form was being entered.
- iii) To ensure that each piece of code functions as intended, frequent unit testing was conducted throughout the development process.
- iv) Unit testing assisted in the early detection of issues and flaws, making resolution easier.

4.4.2.2. System testing:

- i) System testing comprised of testing the complete system from the user's perspective, including the login process, question submission, and grading.

- ii) This testing was a result to ensure proper user interface and interaction.
- iii) System testing included test cases like to view list of users, exams, questions, results, and answers allowing the end users to properly view the data fetched from the database.
- iv) System testing helped to identify any issues with the overall functionality of the system, such as weaknesses in the fetching the data or even some validation error.

4.4.3. Conclusion (Test Analysis):

System and unit testing were crucial steps in evaluating the efficiency and integrity of an online exam system. System testing examined the system as a whole to make sure it satisfies its requirements and functions properly from the perspective of the user, whereas unit testing examined individual system components to aid in identifying flaws and problems at an early stage. By utilizing both testing options, Scholar could make sure that the online examination system is dependable, safe, readily accessible, and accurately assesses student performance. A rigorous testing procedure may highlight areas for improvement and ensure that the system is performing at its best, which ultimately enhances user experience and produces a more accurate assessment of students' knowledge and skills.

Moreover, tests have been followed according to the methodology where testing has been done while building the prototype as well as during the refinement of the prototype. Further, testing has been carried out after refinement of the prototype as well.

5. Conclusion:

5.1. Legal, Social and Ethical Issues:

5.1.1. Legal Issues:

5.1.1.1. No location tracking through the system:

The application does not require any location tracking as no such feature has been integrated inside it. This secures the location of the users. And no one can share other individual's location without their consent.

5.1.1.2. Privacy Policy:

Scholar as a project is very much focused on user privacy. The user privacy is maintained. Even the user password is encrypted for the admin maintaining user privacy within the system. No one can see each other's information except for the administration and people assigned by the administration.

The functionalities in the system have been carried out abiding by the laws and no legal boundaries has been breached while development of the product and implementation as well.

5.1.1.3. System limits data collection:

Scholar only requires data necessary for conducting and attending examinations. This causes data minimization as well limits the information about the user (only asks for required information) protecting user privacy.

5.1.1.4. Data transparency:

User data is not being used for any other purpose other than integrating it to this particular application.

So, there no such legal issues associated with this application.

5.1.2. Social Issues:

5.1.2.1. Unequal access to technology and internet:

This application is only functioning when the user is connected to a compatible device and internet connection. Though we live in this fast-changing technological world, there are still places where people do not have access to the modern technology that includes smartphones and computer systems. People also don't have access to internet connection. For this reason, this application can be a social drawback.

The only way to overcome this social issue is rapid development of technological aspects required.

5.1.2.2. Online forum discussion:

Online forum discussion has been implemented, where students can comment on the notices posted by staff and teacher increasing the social interaction even being an examination system.

5.1.2.3. Personal results:

Students only have access to their personal results. A student cannot view any other user's results, maintaining student privacy and respecting your social dignity.

The only major social issue the users have to face is in lack of technological devices or access to internet.

5.1.3. Ethical Issues:

5.1.3.1. Maintaining integrity:

The system is in plan to make it 0 cheating online evaluation. This plan has been discussed in the future work section.

5.1.3.2. Impartial checking:

As answers are pre-selected or pre-defined, no student has to worry about the evaluation being partial or biased.

5.1.3.3. Maintaining privacy and confidentiality:

Scholar will be using secure servers and encryption methods to protect personal information.

5.1.3.4. Demonstration:

Scholar ensures to demonstrate the ethical principles followed to maintain transparency and integrity of online evaluation.

With Scholar being developed and implemented with full integrity there no ethical issues associated with the application.

5.2. Advantages:

Scholar is a huge boost to modern era in the field of education. This is justified below:

- i) Cost-effective: By removing the need for paper-based exams, printing costs, and shipping costs, this application can help institutions and students save money. Furthermore, grading and giving tests are less required, which can free up staff time and resources for other activities.
- ii) Flexibility: Scholar program eliminates the requirement for students to physically visit to a testing centre by enabling them to take exams from any location with internet access. This makes it more convenient for students who might find it difficult to travel to a testing site due to other obligations, such as family obligations or career commitments.
- iii) Individualized Assessment: The online exam platform can provide assessments that are tailored to the needs and learning styles of each student. This supports a flexible, individualized approach to learning and assessment.
- iv) Real-time Feedback: Your online exam application allows students to examine their scores immediately after completing the test, providing real-time feedback. They can then identify their prospective growth areas and adjust their learning strategies accordingly.
- v) Saving time: This program can help both students and instructors by automating the grading process. Due to the elimination of manual exam marking, instructors now have more time to devote to organizing lessons and conducting research.
- vi) Global Reach: A bigger audience, regardless of location, can access the online examination application. Allowing students from various countries and regions to take the same exam, promotes greater variety and a larger range of viewpoints.
- vii) Scalability: The online exam application is incredibly scalable, enabling several users to sign in at once. This implies that it may just need to be scaled up or down to account for varying student counts.

- viii) Reduced Environmental Impact: You can lessen the harm that paper tests cause to the environment by using an online testing solution. It contributes to resource conservation and waste reduction by doing away with the need for paper and printing.
- ix) Analytics: Analytics and data insights that enable educators to monitor students' progress, identify patterns, and evaluate the effectiveness of their teaching strategies can be carried out. This can enhance results and offer data for deciding on future educational options.

5.3. Limitations:

- i) Technical issues: Technology problems, such as network connectivity failures, server outages, or software defects, could affect online exam systems. These issues can disrupt the exam process and annoy both students and instructors.
- ii) Security concern: Systems used for online exams are prone to security flaws such as identity theft, plagiarism, and hacking. Keeping students from cooperating or sharing resources throughout the exam might be challenging.
- iii) System Compatibility: Some hardware or operating systems may not work with online examination systems, which may restrict some students' access to the tests.
- iv) Cost: Building a system for online exams may need a sizable up-front investment in software development, technology infrastructure, and instructor and student training. Maintaining this software needed frequent costings as well.
- v) Test Design: This application does not allow all exam question types, such as essays, oral exams, or lab practical. This could limit the variety of exams that can be administered using the technology.
- vi) Accessibility: Some students won't have access to the internet, or the equipment required for online tests. This could lead to unequal access to the exam, affecting students who are unable to take it due to technological issues.
- vii) Time constraint: Time restrictions on online tests have the potential to push students and lower their performance. The regular prolongation of the permitted time due to technical difficulties or internet connectivity concerns causes stress and worry.

5.4. Future work:

5.4.1. Unification with Learning Management System:

5.4.2. Mobile compatibility:

5.4.3. Integrating multiple languages:

5.4.4. Adaptive testing:

5.4.5. Accessibility:

5.4.6. Integration of AI:

5.4.7. Integration of data science:

5.4.8. Allowing the test-takers to conduct examination with different answer types:

5.4.9. Blockchain Integration:

5.4.10. Interaction Tools:

5.4.11. User personalisation features:

Future Work elaboration has been kept in appendix section [*7.7.1. Future Work \(Elaborated\)*](#):

6. Bibliography:

Bibliography

Admin, T. E., 2022. *Think Exam*. [Online]

Available at: <https://thinkexam.com/blog/introduction-online-examination-system-how-does-it-work-its-pros-cons/>

[Accessed 19 December 2022].

Al-Shuaibi, A., 2014. *ResearchGate*. [Online]

Available at:

https://www.researchgate.net/publication/260075970_The_Importance_of_Education

[Accessed 22 November 2022].

Bugajenko, O., 2021. *study.com*. [Online]

Available at: <https://study.com/academy/lesson/what-is-the-rational-unified-process-methodology-tools-examples.html>

[Accessed 26 December 2022].

Christensson, P., 2014. *TechTerms*. [Online]

Available at: <https://techterms.com/definition/javascript>

[Accessed 26 December 2022].

ClassMarker, n.d. *ClassMarker*. [Online]

Available at: <https://www.classmarker.com/online-testing/>

[Accessed 25 December 2022].

Deepankar Vishwas Kotwal, S. R. B. A. S. G. P. B., 2016. ONLINE EXAMINATION SYSTEM. *International Research Journal of Engineering and Technology (IRJET)*, 03(01).

Eklavya, n.d. *Eklavya*. [Online]

Available at: <https://www.eklavya.com/about-us/>

[Accessed 25 December 2022].

Exam, S., n.d. *Speed Exam*. [Online]

Available at: <https://admin.speedexam.net/help.aspx>

[Accessed 25 December 2022].

Explorance, 2013. *Explorance*. [Online]

Available at: <https://explorance.com/blog/6-disadvantages-of-traditional-paper-based-course-evaluations-2/>

[Accessed 22 November 2022].

Gomes, O., 2020. *Talview*. [Online]

Available at: <https://blog.talview.com/a-definitive-guide-to-online-exam-proctoring-for->

universities

[Accessed 19 December 2022].

Hamilton, T., 2022. *Guru99*. [Online]

Available at: <https://www.guru99.com/agile-scrum-extreme-testing.html>

[Accessed 27 November 2022].

Hamilton, T., 2023. *Guru99*. [Online]

Available at: <https://www.guru99.com/software-testing-introduction-importance.html>

[Accessed 14 April 2023].

Hamilton, T., 2023. *Guru99*. [Online]

Available at: <https://www.guru99.com/black-box-testing.html>

[Accessed 14 April 2023].

Help, S. T., 2023. *Software Testing Help*. [Online]

Available at: <https://www.softwaretestinghelp.com/white-box-testing-techniques-with-example/>

[Accessed 14 April 2023].

InterQuality, n.d. *InterQuality*. [Online]

Available at: <https://interqualitybg.com/en/resources/scrum-and-agile-resources/agile-methodology>

[Accessed 27 November 2022].

Janse, B., 2019. *toolhero*. [Online]

Available at: <https://www.toolhero.com/information-technology/rational-unified-process-rup/>

[Accessed 26 December 2022].

Kinsta, 2022. *Kinsta*. [Online]

Available at: <https://kinsta.com/knowledgebase/what-is-postgresql/>

[Accessed 26 December 2022].

Kumar, N., 2021. *freeCodeCamp*. [Online]

Available at: <https://www.freecodecamp.org/news/model-view-architecture/#:~:text=First%20the%20browser%20sends%20a,not%20the%20final%20presentation.>

[Accessed 27 December 2022].

Martin, M., 2022. *Guru99*. [Online]

Available at: <https://www.guru99.com/software-engineering-prototyping-model.html>

[Accessed 24 November 2022].

Morris, S., n.d. *skillcrush*. [Online]

Available at: <https://skillcrush.com/blog/what-is-react-js/>

[Accessed 26 December 2022].

Muna R. Hameed, F. A. A., 2017. Online Examination System. *International Advanced Research Journal in Science, Engineering and Technology*, 4(3).

naimishsahu08, 2021. GeeksforGeeks. [Online]

Available at: <https://www.geeksforgeeks.org/software-prototyping-model-and-phases/> [Accessed 24 November 2022].

Pandit, N., 2021. C#Corner. [Online]

Available at: <https://www.c-sharpcorner.com/article/what-and-why-reactjs/> [Accessed 26 December 2022].

pappersurvey, 2021. *pappersurvey.io*. [Online]

Available at: <https://www.pappersurvey.io/blog/paper-based-survey-advantages-and-disadvantages>

[Accessed 22 November 2022].

quintagroup, n.d. *quintagroup*. [Online]

Available at: <https://quintagroup.com/cms/python/django-rest-framework> [Accessed 26 December 2022].

Sherman, R., 2015. *ScienceDirect*. [Online]

Available at: <https://www.sciencedirect.com/topics/computer-science/waterfall-methodology>

[Accessed 27 November 2022].

Sherrell, L., 2013. *Springer Link*. [Online]

Available at: https://link.springer.com/referenceworkentry/10.1007/978-1-4020-8265-8_201039

[Accessed 24 November 2022].

Team, A. C., 2022. *Adobe*. [Online]

Available at: <https://business.adobe.com/blog/basics/waterfall> [Accessed 27 November 2022].

Techopedia, 2012. *Techopedia*. [Online]

Available at: <https://www.techopedia.com/definition/28227/django> [Accessed 26 December 2022].

user4541274, n.d. *Freepik*. [Online]

Available at: https://www.freepik.com/premium-vector/person-pile-papers-with-hand-holding-sign-help-outline-drawing-style-art_31580766.htm

[Accessed 19 December 2022].

Wilson, J., 2022. *techopedia*. [Online]

Available at: <https://www.techopedia.com/definition/3533/python> [Accessed 26 December 2022].

Yuan Zhenming, Z. L. Z. G., 2003. A NOVEL WEB-BASED ONLINE EXAMINATION SYSTEM FOR COMPUTER SCIENCE EDUCATION. *33rd ASEE/IEEE Frontiers in Education Conference*.

7. Appendix:

7.1. Appendix A: Pre-Survey:

7.1.1. Pre-Survey Form:

The screenshot shows a survey form titled "Scholar". At the top, there is a brief description: "This is an online and computer based evaluation system. This application is a suitable step to ensure that technology is used in education, enabling it to advance and improve institutional sectors." Below this, there is a field labeled "Email *". A placeholder text "Valid email address" is visible in the input field. A note below the field states, "This form is collecting email addresses. [Change settings](#)". The main content area contains a question: "Enter your full name". To the right of this question is a dropdown menu set to "Short answer". Below this question is a "Short-answer text" input field. On the far right of the form interface, there is a vertical toolbar with icons for adding fields, deleting fields, changing text style, and other form-related functions. At the bottom right of the form area, there are buttons for "Required" (which is turned on) and a three-dot menu icon.

Figure 135 Pre-Survey Form (fig 1)

Enter your email address

Short-answer text

Have you ever used an online based or computer based examination? *

Yes

No

Do you prefer your examination to be online or computer system based? *

Yes

No

Figure 136 Pre-Survey Form (fig 2)

Online examination is better than the traditional approach of conducting the examination. Why or why not?

Short-answer text

What facility would you like to be added in the project?

Short-answer text

Have you ever come across such application? *

Yes

No

Figure 137 Pre-Survey Form (fig 3)

The image shows a survey form with two questions. The first question is "Can it be the new and better way of evaluating papers? *". It has two radio button options: "Yes" and "No". The second question is "Would you suggest it to be used in educational institutes? *". It also has two radio button options: "Yes" and "No". The background of the form is light purple, and there are navigation icons in the top right corner.

Can it be the new and better way of evaluating papers? *

Yes

No

Would you suggest it to be used in educational institutes? *

Yes

No

Figure 138 Pre-Survey Form (fig 4)

7.1.2. Sample of Filled Pre-Survey Form:

The screenshot shows a filled pre-survey form titled "Scholar". The form includes a brief introduction about the online evaluation system, the user's email address, a note about required fields, and three input fields for email addresses.

This is an online and computer based evaluation system. This application is a suitable step to ensure that technology is used in education, enabling it to advance and improve institutional sectors.

[sayushkhadka777@gmail.com](#) [Switch accounts](#) Draft saved

* Indicates required question

Email *

sayushkhadka777@gmail.com

*

Sayush Khadka

Enter your email address

sayushkhadka777@gmail.com

Figure 139 Sample of Filled Pre-Survey Form (fig 1)

Have you ever used an online based or computer based examination? *

- Yes
 No

Do you prefer your examination to be online or computer system based? *

- Yes
 No

Online examination is better than the traditional approach of conducting the examination. Why or why not?

It is better. Easy and convenient.

What facility would you like to be added in the project?

Countdown Timer

Figure 140 Sample of Filled Pre-Survey Form (fig 2)

Have you ever come across such application? *

Yes
 No

Can it be the new and better way of evaluating papers? *

Yes
 No

Would you suggest it to be used in educational institutes? *

Yes
 No

Submit [Clear form](#)

Never submit passwords through Google Forms.

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

Google Forms

Figure 141 Sample of Filled Pre-Survey Form (fig 3)

7.1.3. Pre-Survey Result:

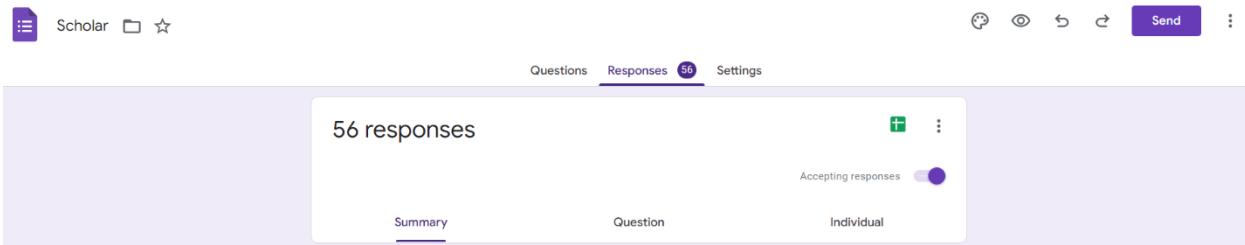


Figure 142 Pre-Survey Result (fig 1)

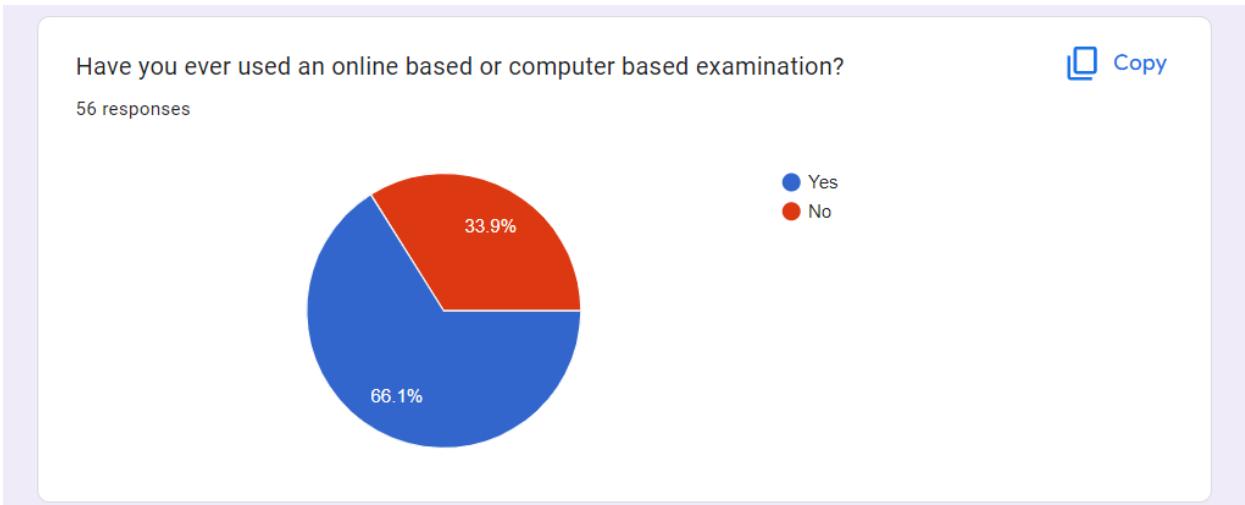


Figure 143 Pre-Survey Result (fig 2)

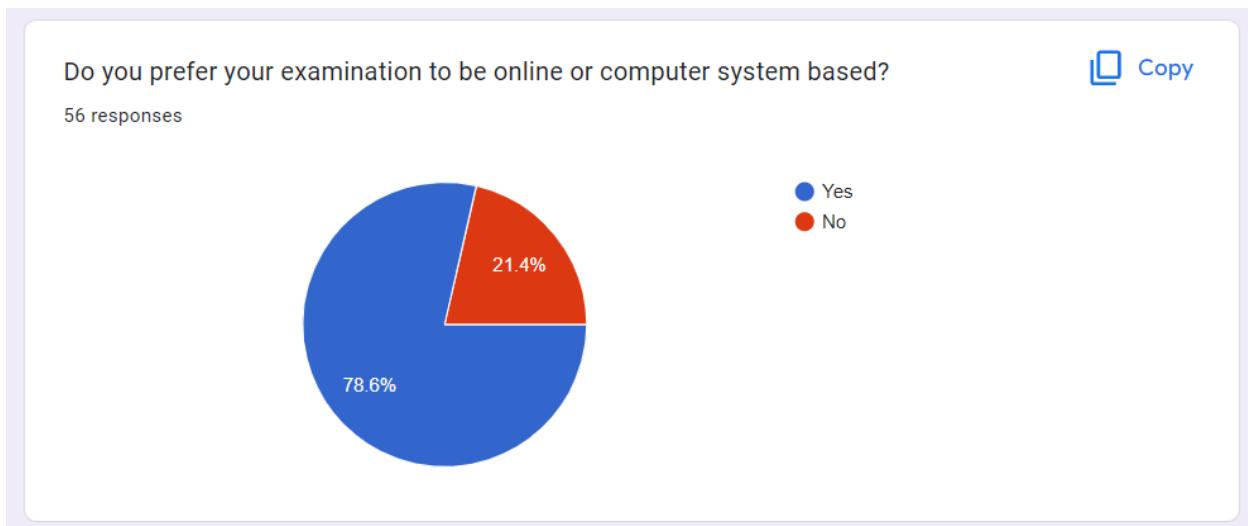


Figure 144 Pre-Survey Result (fig 3)

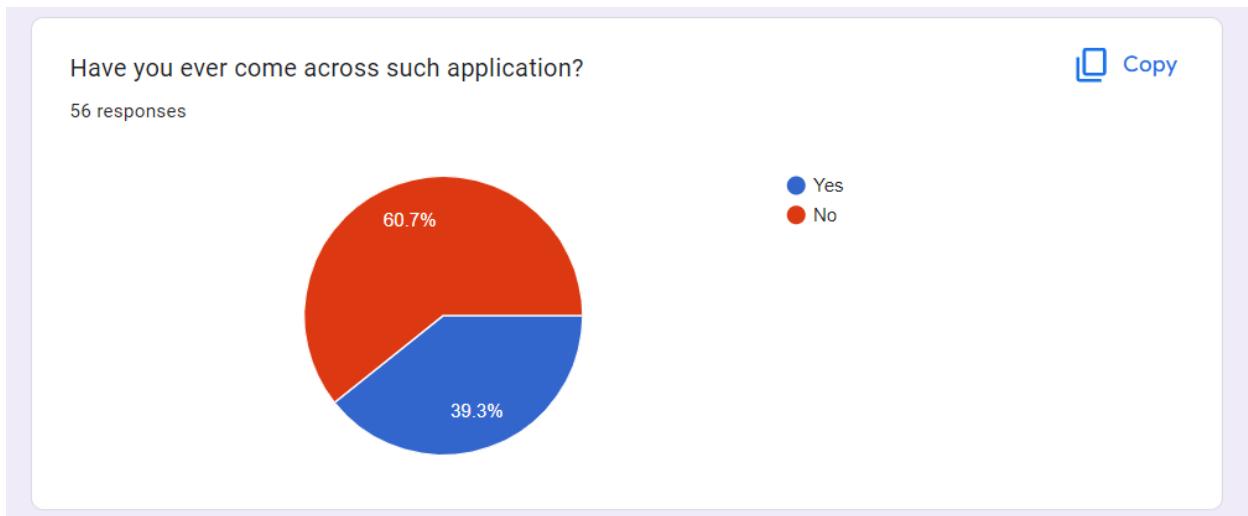


Figure 145 Pre-Survey Result (fig 4)

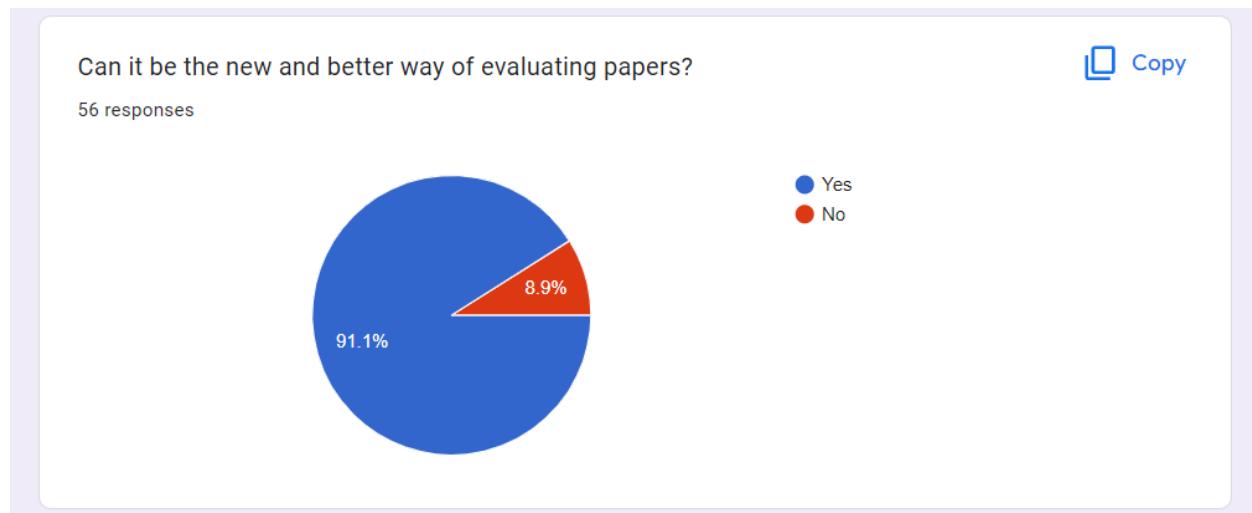


Figure 146 Pre-Survey Result (fig 5)

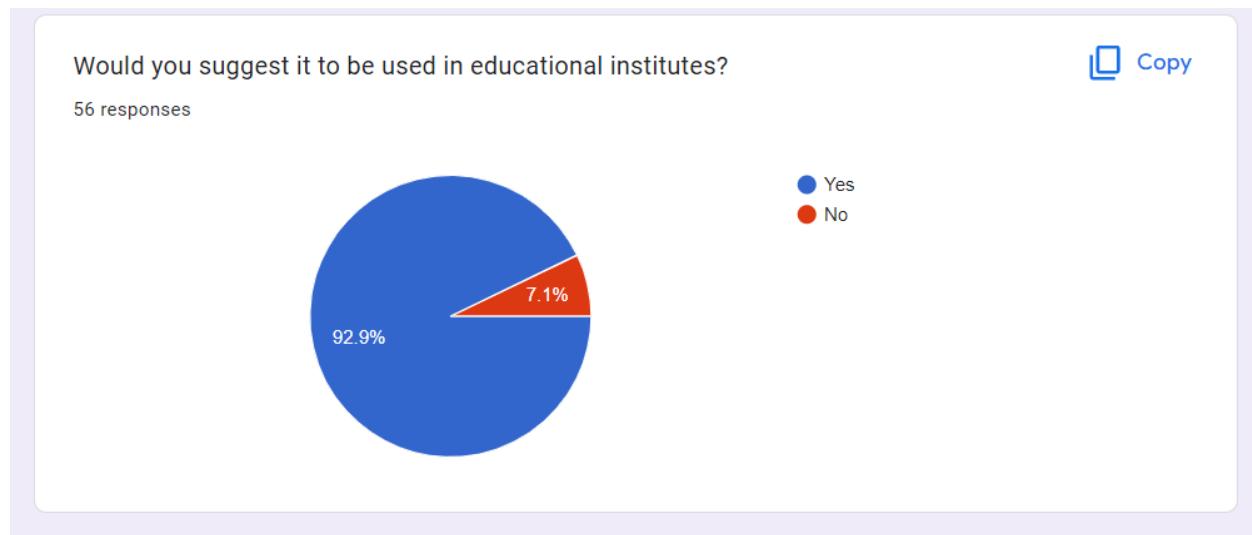


Figure 147 Pre-Survey Result (fig 6)

Online examination is better than the traditional approach of conducting the examination. Why or why not?

38 responses

Online examination is better considering the world we live in today since we have access to technology but regardless traditional approach is also convenient to those area where they lack access to technology a

online examination is better because it is time efficient, environmental friendly and scalable with worldwide reach

Online examination might not be a better way of conductinh exams as it is difficult to monitor and manage time.

No ..due to the different unfortunate circumstance like network connecton, device problem etc.

Yes, because it allows us to analyze the data instead of memorizing it. I believe 21st century is an age of technology and information. You can get info anywhere but its up to us how we use those info. Online exam allows us such analytical approach of evaluating students' capacity.

Because it allows us to write and think freely on our own space.

It's more comfortable

Figure 148 Pre-Survey Result (fig 7)

Online examination is better than the traditional approach of conducting the examination. Why or why not?

38 responses

However, if everything goes well and this problem is solved in future, online exam can be a better option at that time, but not in the current situation of the country.

Very nice

Advantages of Online Exams hlp4othr@gmail.com Online examination is better than the traditional approach of conducting the examination. Why There are several reasons why online examinations may be better than traditional examinations in certain situations: Convenience: Online examinations can be taken from any location with an internet connection, which is more convenient for students who may not be able to travel to a specific location to take an exam. Flexibility: Online exams can be scheduled at a time that is convenient for the student, which may not be possible with traditional exams that are held at specific times and locations. Reduced cheating: Online exams can be designed to minimize the opportunity for cheating, such as by using proctoring software that monitors the student's webcam and screen during the exam. Increased accessibility: Online exams can be made more accessible for students with disabilities by providing accommodations such as text-to-speech or larger font sizes. Cost savings: Online exams can be more cost-effective for institutions to administer, as they do not require the use of physical space or resources like paper and ink. Of course, there are also potential drawbacks to online exams, such as the need for reliable internet access and potential technical issues. It is important to carefully consider the specific needs and goals of a particular examination and determine the best approach based on those factors.

Figure 149 Pre-Survey Result (fig 8)

Online examination is better than the traditional approach of conducting the examination. Why or why not?

38 responses

Not fare

yes it will be more effective

Yes online examination is better than traditional examination because today's world is world of technology so, people want to use technical things .

online examination is better because it save times and it can help to conduct examinations on time . Students can complete their grade on time, which can help to make students future bright.

ok its fast and secured.

Since we have greater flexibility, taking examinations online is preferable

Traditional approach is better because people can't easily cheat.

Because we can cheat efficiently.

Figure 150 Pre-Survey Result (fig 9)

7.2. Appendix B: Post-Survey:

7.2.1. Post-Survey Form:

Questions Responses 18 Settings

Scholar

This is an online and computer based evaluation system. This application is a suitable step to ensure that technology is used in education, enabling it to advance and improve institutional sectors.

Image title



Enter your full name.*

Short-answer text

(+) 

(-) 

T 

A 

Y 

≡ 

Figure 151 Post-Survey Form (fig1)

How would you rate your application experience? *

Excellent

Very Good

Good

Fair

Poor

How was the user interface and user experience of the application? *

Excellent

Very Good

Good

Fair

Poor

The image shows a survey form with two questions. The first question asks about application experience with five rating options: Excellent, Very Good, Good, Fair, and Poor. The second question asks about user interface and user experience with the same five rating options. To the right of the questions is a vertical toolbar with icons for adding, saving, and other document-related functions.

Figure 152 Post-Survey Form (fig2)

Do you think this application would help the purpose of eradicating traditional approach of conducting examination? *

Yes

No

Maybe

Would you recommend educational institute to implement this application? *

Yes

No

Maybe

What additional features do you think the application requires for further development?

Short-answer text

Figure 153 Post-Survey Form (fig3)

Do you think this application is an easier way to conduct and attend examination? *

Yes

No

Was the application easy to use and navigate through different features? *

Yes

No

Suggestion box. We welcome your other feedbacks on this application.

Long-answer text

Figure 154 Post-Survey Form (fig4)

7.2.2. Sample of Filled Post-Survey Form:

Scholar

This is an online and computer based evaluation system. This application is a suitable step to ensure that technology is used in education, enabling it to advance and improve institutional sectors.

[sayushkhadka777@gmail.com](#) [Switch accounts](#)

 Draft saved

 Not shared

* Indicates required question



Enter your full name. *

Sayush Khadka

Figure 155 Sample of Filled Post-Survey Form (fig1)

How would you rate your application experience? *

Excellent

Very Good

Good

Fair

Poor

How was the user interface and user experience of the application? *

Excellent

Very Good

Good

Fair

Poor

Figure 156 Sample of Filled Post-Survey Form (fig 2)

Do you think this application would help the purpose of eradicating traditional approach of conducting examination? *

Yes

No

Maybe

Would you recommend educational institute to implement this application? *

Yes

No

Maybe

What additional features do you think the application requires for further development?

AI Integration

Figure 157 Sample of Filled Post-Survey Form (fig3)

Do you think this application is an easier way to conduct and attend examination? *

Yes
 No

Was the application easy to use and navigate through different features? *

Yes
 No

Suggestion box. We welcome your other feedbacks on this application.

Do implement AI

Submit **Clear form**

Never submit passwords through Google Forms.

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

Google Forms

Figure 158 Sample of Filled Post-Survey Form (fig 5)

7.2.3. Post-Survey Result:

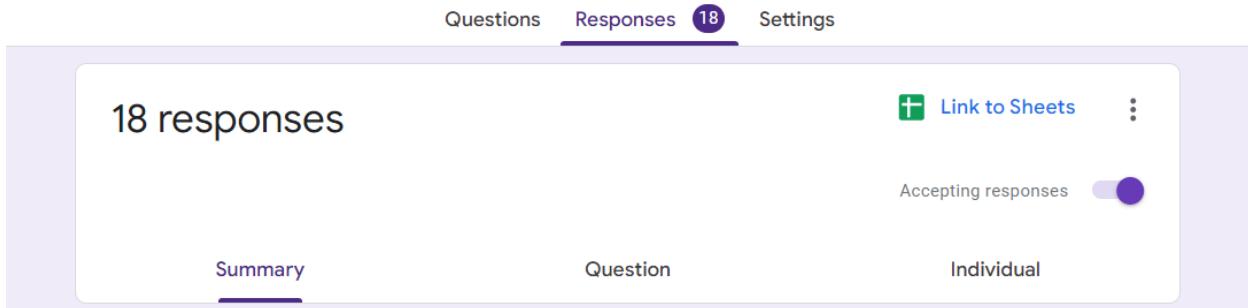


Figure 159 Post-Survey Result (fig 1)



Figure 160 Post-Survey Result (fig 2)

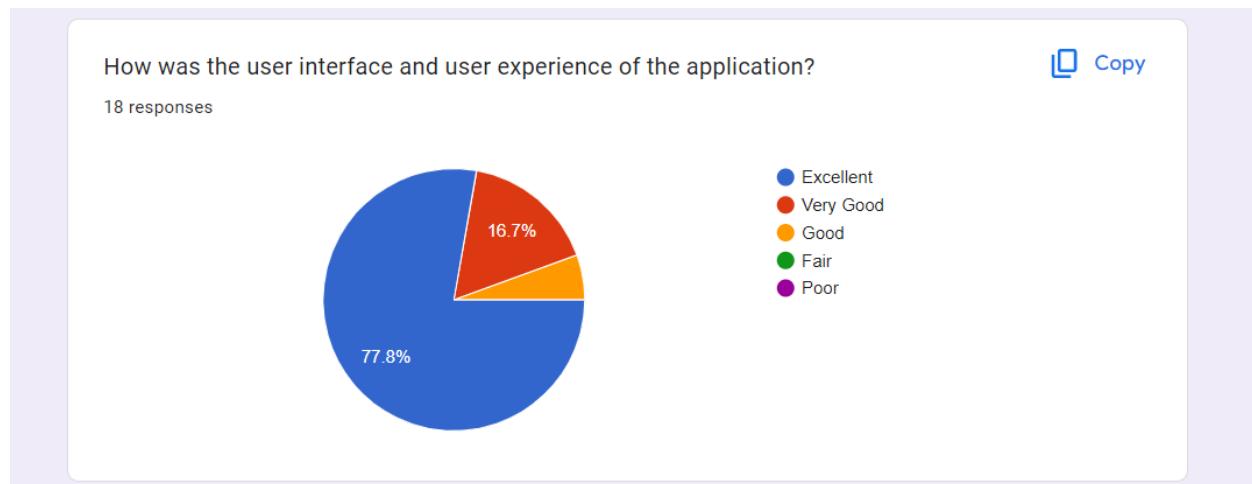


Figure 161 Post-Survey Result (fig 3)

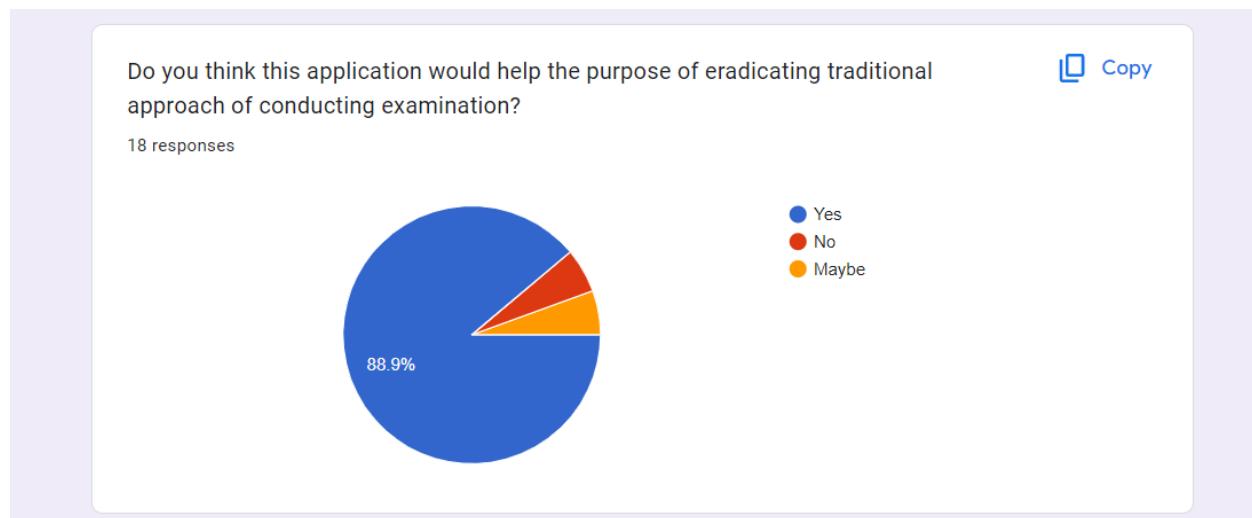


Figure 162 Post-Survey Result (fig 4)

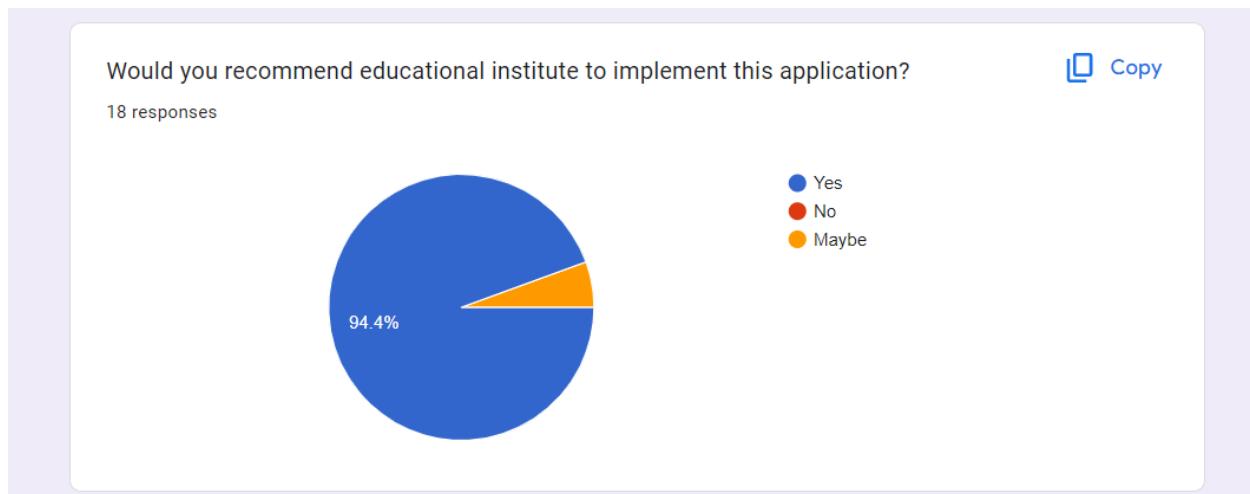


Figure 163 Post-Survey Result (fig 5)

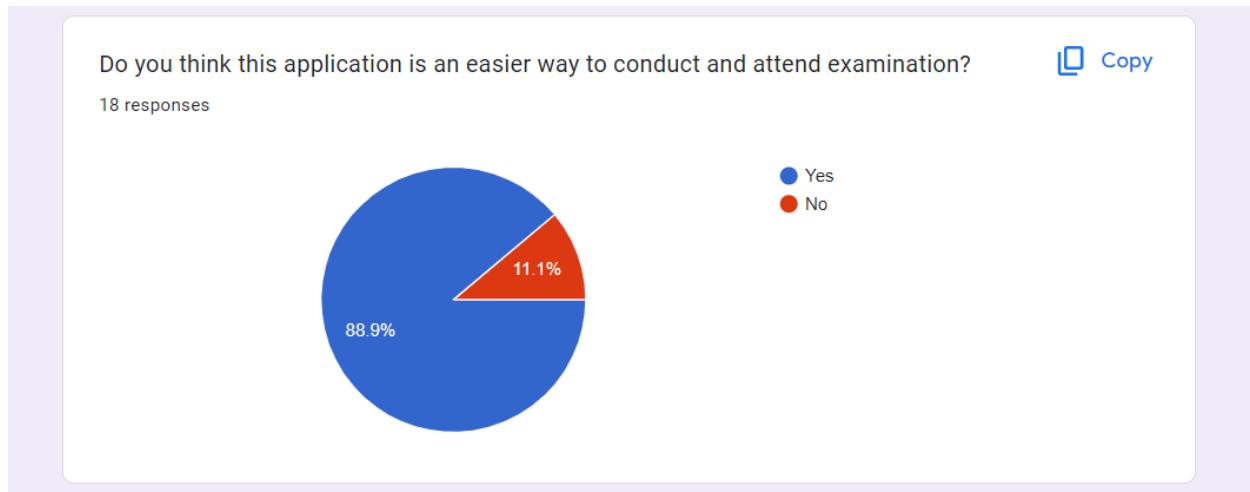


Figure 164 Post-Survey Result (fig 6)

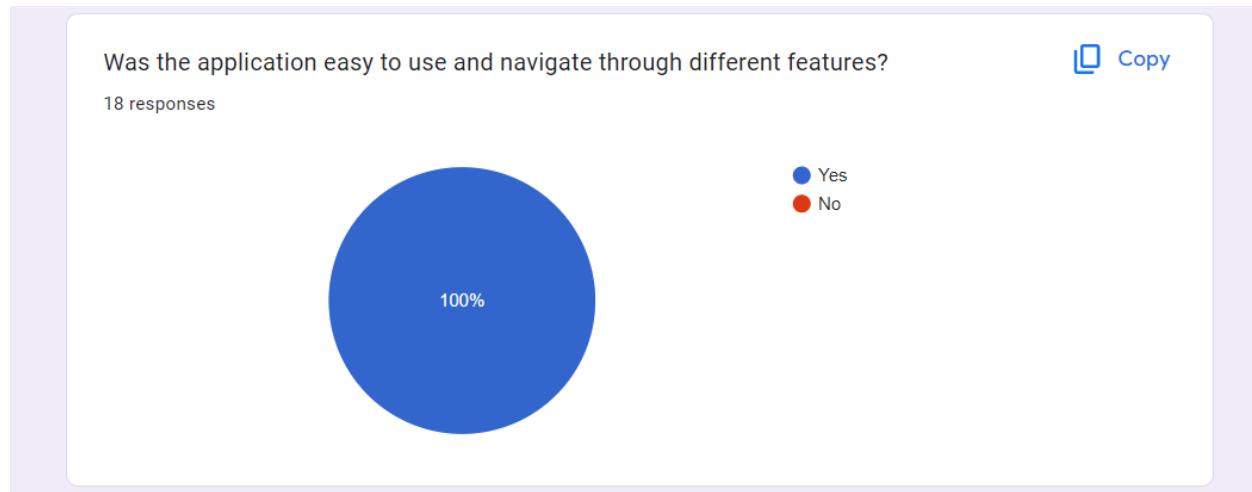


Figure 165 Post-Survey Result (fig 7)

What additional features do you think the application requires for further development?

6 responses

Short answers features can be required

Enhanced UI and voice assistant

Other type of questions

Able to essay as well

Figure 166 Post-Survey Result (fig 8)

7.3. Appendix C: Sample Codes:

7.3.1. Sample Code of the UI:

```
1  {%- load static %}  
2  <!DOCTYPE html>  
3  <html lang="en">  
4  
5  <head>  
6      <title>Scholar</title>  
7      <!-- HTML5 Shim and Respond.js IE11 support of HTML5 elements and media queries --&gt;<br/>8      <!-- WARNING: Respond.js doesn't work if you view the page via file:// --&gt;<br/>9      <!--[if lt IE 11]>  
10         <script src="https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js"></script>  
11         <script src="https://oss.maxcdn.com/libs/respond.js/1.4.2/respond.min.js"></script>  
12     <![endif]-->  
13     <!-- Meta -->  
14     <meta charset="utf-8">  
15     <meta name="viewport" content="width=device-width, initial-scale=1.0, user-scalable=0, minimal-ui">  
16     <meta http-equiv="X-UA-Compatible" content="IE=edge" />  
17     <meta name="description"  
18         content="Free Datta Able Admin Template come up with latest Bootstrap 4 framework with basic components, form elements and lots of pre-made  
19     <meta name="keywords"  
20         content="admin templates, bootstrap admin templates, bootstrap 4, dashboard, dashboard templets, sass admin templets, html admin templates,  
21     <meta name="author" content="CodedThemes" />  
22  
23     <style>  
24         * {  
25             padding: 0;  
26             margin: 0;  
27             text-decoration: none;  
28             list-style: none;  
29             box-sizing: border-box;  
30             background-color: #272C60;  
31         }  
32  
33         .index-body {  
34             height: 100vh;  
35             width: 100%;
```

Figure 167 Index page (UI)

```

account > templates > account > teacher_registration_form.html > ...
1  {% extends 'account/base_register.html' %}
2  {% load static %}
3
4  {% block title %}
5  Teacher Registration Form
6  {% endblock %}
7
8
9  {% block content%}
10
11  <!-- <div class="right-register"> -->
12  <div class="pcoded-main-container">
13    <h4><b><a href="{% url 'main:index' %}">Scholar</a></b></h4>
14  <div class="page-wrapper">
15    <!-- [ Main Content ] start -->
16    <form method="POST" id="levelSectionForm" level-section-url="{% url 'account:load-section' %}">
17      {% csrf_token %}
18    <div class="row">
19      <div class="col-sm-12">
20        <div class="register-card">
21          <!-- <div class="card-header">
22            |   <h4><b>Scholar</b></h4>
23          </div -->
24        <div class="card-body">
25          <h5><b>Teacher Registration</b></h5>
26          <hr>
27
28
29    <div class="row">
30      <div class="col-md-6">
31
32
33
34      <div class="form-group">
35        <!-- <label for="exampleInputEmail1">First Name</Label> -->
36        <!-- <input type="text" class="form-control" placeholder="First Name"> -->
37        {{ form.first_name }}

```

Figure 168 Teacher Registration (UI)

```

account > templates > account > teacher_login.html > div.auth-wrapper-login > div.auth-content > div.card > form > div.card-body.text-center
1  {% extends 'account/base_login.html' %}
2  {% load static %}
3
4  {% block title %}
5  Teacher Login
6  {% endblock %}
7
8  {% block content %}
9
10 <!-- <div class="whole-page-Login"> -->
11 <!-- <div class="right-login"> -->
12 <div class="auth-wrapper-login">
13   <h4><b><a href="{% url 'main:index' %}">Scholar</a></b></h4>
14   
15   <div class="auth-content">
16
17     <!-- <div class="auth-bg">
18       |   <span class="r"></span>
19       |   <span class="r s"></span>
20       |   <span class="r s"></span>
21       |   <span class="r"></span>
22     </div> -->
23
24   <div class="card">
25     <form method="POST">
26       {% csrf_token %}
27       <div class="card-body text-center">
28         <p class="text-danger">{{ error }}</p>
29         <div class="mb-4">
30           |   <!-- <i class="feather icon-unlock auth-icon"></i> -->
31           |   <div>
32             <h4 class="mb-4"><b>Teacher Login</b></h4>
33             <div class="input-group mb-3">
34               |   {{ form.username }}
35             </div>
36             <div class="input-group mb-4">

```

Figure 169 Teacher Login (UI)

```

1  {% extends 'main/base.html' %}
2  {% load static %}
3
4  {% block title %}
5  Teacher Notice Board
6  {% endblock %}
7
8  {% block content %}
9  <h3 class="jumbotron" style="padding: 10px; margin: 10px;">Teacher Notice Board</h3>
10
11 <h6 class="jumbotron" style="padding: 10px; margin: 10px;">
12 | Return back to: <a href="{% url 'main:teacher-page' %}">Teacher Main Page</a>
13 </h6>
14
15 <h5 class="jumbotron" style="padding: 10px; margin: 10px;">
16 | Total number of notice: {{count}}
17 <form method="POST" action="{% url 'main:teacher-create-notice' %}">
18 |     {% csrf_token %}
19 |     <button class="btn btn-primary" style="width:fit-content; padding: 4px; margin:10px;">Add more</button>
20 </form>
21 </h5>
22
23 <div class="card-columns" style="padding: 5px;"></div>
24
25
26 {%for forum in forums %}
27
28 <div class="card box container">
29 |     <br>
30 |     <h5 class="card-title">
31 |         <h3>{{forum.topic}}</h3>
32 |     <div class="card-body container">
33 |         <p>{{forum.description}}</p>
34 </div>
35

```

Figure 170 Teacher Notice Board (UI)

```

5  Staff Page
6  {% endblock %}
7
8  {% block content %}
9  <!-- [ Pre-Loader ] start -->
10 <!-- <div class="loader-bg">
11 |     <div class="loader-track">
12 |         <div class="loader-fill"></div>
13 |     </div>
14 </div> -->
15 <!-- [ Pre-Loader ] End -->
16 <!-- [ navigation menu ] start -->
17 <nav class="pcoded-navbar">
18     <div class="navbar-wrapper">
19         <div class="navbar-brand header-logo">
20             <a href="{% url 'main:index' %}" class="b-brand">
21                 <!-- <div class="b-bg">
22                 |     <i class="feather icon-trending-up"></i>
23                 |     </div> -->
24                 <span class="b-title">Scholar</span>
25             </a>
26
27         </div>
28         <div class="navbar-content scroll-div">
29             <ul class="nav pcoded-inner-navbar">
30                 <li class="nav-item pcoded-menu-caption">
31                     <label style="text-transform: none;">Username: {{ user.username }}</label>
32                 </li>
33
34                 <li data-username="Table bootstrap datatable footable" class="nav-item active">
35                     <a href="{% url 'main:staff-page' %}" class="nav-link "><span class="pcoded-micon"><i
36                         |     class="feather icon-home"></i></span><span class="pcoded-mtext">Staff Main</span></a>
37                 </li>
38

```

Figure 171 Staff Page UI

```

exam_material > templates > exam_material > quiz.html > div.quiz > div.quiz-header > div.row > div#timer-box.col.text-right
  1  {% extends 'exam_material/exam_material_base.html' %} 
  2  {% load static %} 
  3
  4  {% block title %} 
  5    Exam
  6  {% endblock %} 
  7
  8  {% block scripts %} 
  9    <script src="{% static 'js/quiz.js' %}" defer></script>
 10  {% endblock scripts %} 
 11
 12  {% block content %} 
 13
 14
 15
 16  <div class="quiz">
 17    <div class="quiz-header">
 18      <div class="row">
 19        <div class="col">
 20          <b>Subject: {{ obj.subject }}</b>
 21          <br>
 22          <b>Exam title: {{ obj.title }}</b>
 23          <br>
 24          Number of Questions: {{ obj.numberOfQuestions }} 
 25          <p>Score to pass: {{ obj.scoreToPass }}%</p>
 26          <a href="{% url 'exam_material:quiz-main' %}">Return to available exams</a>
 27        </div>
 28
 29      <div class="col text-right" id="timer-box"> </div>
 30
 31    </div>
 32  </div>
 33
 34
 35
 36

```

Figure 172 Quiz page (UI)

```

21  </div> -->
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53  <script src="https://code.jquery.com/jquery-3.3.1.min.js"></script>
54  <script>
55    $("#id_level").change(function () {
56      const url = $("#levelSectionForm").attr("level-section-url"); // get the url of the `load_level_section` view
57      const levelId = $(this).val(); // get the selected level ID from the HTML input
58
59      $.ajax({
60        url: url, // initialize an AJAX request
61        data: { // set the url of the request (= /account/load-section/ )
62          'level_id': levelId // add the level id to the GET parameters
63        },
64        success: function (data) { // `data` is the return of the `load_level_section` view function
65          $("#id_levelSection").html(data); // replace the contents of the section input with the data that came from the server
66          /*
67            Let html_data = '<option value="">-----</option>';
68            data.forEach(function (section) {
69              html_data += '<option value="' + section.id + '">' + section.name + '</option>';
70            });
71            console.log(html_data);
72            $("#id_section").html(html_data);
73            /*
74        }
75      });
76    });
77  </script>
78
79
80
81
82
83  {% endblock %}

```

Figure 173 For Class Section (Template)

```

account > templates > account > level_section.html ...
● 1  <option value="">-----</option>
2  2  {% for section in sections %}
3  3  <option value="{{ section.pk }}">{{ section }}</option>
4  4  {% endfor %}

```

Figure 174 For Class Section(continued) (Template)

```

1  <!-- question_create_or_update.html -->
2
3  {% extends "account/base_register.html" %}
4
5  {% block title %}
6  Set Question
7  {% endblock %}
8
9  {% block content %}
10 <form enctype="multipart/form-data" class="container" method="post" id="question_form">
11   {% csrf_token %}
12   <!-- main form start --- in our case question form -->
13   <div class="card" style="width: 80%; margin-left: 10%;">
14     <div class="card-header card-header-secondary" style="background-color: #272C60;">
15       <h4 class="card-title" style="color: white;><b>Add Questions</b></h4>
16       <li class="breadcrumb-item" style="list-style-type: none;"><a href="{% url 'main:teacher-page' %}"><i
17         | | |
18         | | | class="feather icon-home"></i>Return to your portal</a></li>
19     </div>
20     {% for field in form %}
21       <div class="form-group card-body">
22         <label>{{field.label}}</label>
23         {% if field.field.required %}
24           <span style="color: red;" class="required">*</span>
25         {% endif %}
26
27           {{field}}
28           {% if field.help_text %}
29             <small style="color: grey">{{ field.help_text }}</small>
30           {% endif %}
31           {% for error in field.errors %}
32             <p style="color: red">{{ error }}</p>
33           {% endfor %}
34       </div>
35     {% endfor %}

```

Figure 175 Create Question (Template)

```

16 |     <form method="POST">
17 |         {% csrf_token %}
18 |         <div class="row">
19 |             <div class="col-sm-12">
20 |                 <div class="register-card">
21 |                     </-- <div class="card-header">
22 |                         |                         <h4><b>Scholar</b></h4>
23 |                         |                     </div> -->
24 |                     <div class="card-body">
25 |                         <h5><b>Class Registration</b></h5>
26 |                         <hr>
27 |
28 |
29 |
30 |
31 |
32 |
33 |
34 |                         {{ form.level }}
35 |                         <div class="text-danger">{{ form.level.errors }}</div>
36 |
37 |
38 |                         <p>Note: Enter positive integer number only.</p>
39 |                         <div>
40 |                             <button type="submit" class="btn btn-primary">Register</button>
41 |                         </div>
42 |
43 |
44 |                         </div>
45 |                         <p class="mb-0 text-muted" style="text-align: center;">Get back to staff page: <b><a
46 |                                         style="color: black;" href="{% url 'main:staff-page' %}">Staff Page</a></b></p>
47 |
48 |
49 |                     </div>
50 |                 </div>
51 |             </div>
52 |         </div>
53 |     </form>
54 |
55 | 
```

Figure 176 Class Registration (Template)

7.3.2. Sample Code of the View:

```
main > views.py > teacherNoticeBoard
  1  from django.views import generic
  2  from account.forms import TeacherRegistrationForm
  3  from django.urls import reverse_lazy
  4  from django.shortcuts import render, redirect
  5  from django.views import generic
  6  from django.http import HttpResponseRedirect, HttpResponseRedirect
  7  from django.contrib import messages
  8
  9  from account.models import Student, Teacher, Staff
 10 from account.forms import StudentRegistrationForm, TeacherRegistrationForm
 11 from main.forms import TeacherUpdateForm, StaffUpdateForm, StudentUpdateForm
 12
 13 from .forms import *
 14
 15 class Index(generic.TemplateView):
 16     template_name = 'main/index.html'
 17
 18 class StudentPage(generic.TemplateView):
 19     template_name = 'main/student_page.html'
 20
 21 class TeacherPage(generic.TemplateView):
 22     template_name = 'main/teacher_page.html'
 23
 24 class StaffPage(generic.TemplateView):
 25     template_name = 'main/staff_page.html'
 26 # class AfterTeacherLogin(generic.FormView):
 27 #     template_name = 'account/teacher_Login.html'
 28 #     form_class = TeacherRegistrationForm
 29 #     context_object_name = 'user_teacher_username'
 30 #     success_url = reverse_lazy('main:after_teacher_login')
 31
 32 def staffNoticeBoard(request):
 33     forums=Forum.objects.all()
 34     count=forums.count()
 35     discussions=[]
 36     for i in forums:
 37         discussions.append(i)
```

Figure 177 Index page (View)

```

account > ⚡ views.py > ...
54
55     #teacher registration
56     class TeacherRegistration(generic.CreateView):
57         template_name = 'account/teacher_registration_form.html'
58         form_class = TeacherRegistrationForm
59         model = Teacher
60         success_url = reverse_lazy('account:teacher-registration')
61
62     def form_valid(self, form) -> HttpResponseRedirect:
63         self.object=form.save()
64         user = self.object.user
65         # username = form.cleaned_data.get('username')
66         # password = form.cleaned_data.get('password1')
67
68         subjects = list(self.object.teacher_subject.values_list('subject', flat=True))
69
70         subject = f'Login credentials, {user.get_role_display()}.'
71         messageFromEmail = f'Hello {user.get_full_name()}. Welcome to Scholar.\nYour login credentials are as follows:\nUsername: {user.username}\n\n'
72         fromEmail = settings.EMAIL_HOST_USER
73         toEmail = [user.email]
74         send_mail(subject, messageFromEmail, fromEmail, toEmail, fail_silently=True)
75
76         messages.success(self.request, f'{user.username} was registered successfully. The login credentials have been delivered to {user.username}.')
77         return HttpResponseRedirect(self.get_success_url())
78
79     def load_level_section(request):
80         level_id = request.GET.get('level_id')
81         sections = LevelSection.objects.filter(level_id=level_id)
82         return render(request, 'account/level_section.html', {'sections': sections})
83
84
85

```

Figure 178 Teacher Registration (View)

```

account > ⚡ views.py > ...
219     |     |     return self.render_to_response(context)
220
221     #teacher Login
222     class TeacherLogin(generic.FormView):
223         template_name = 'account/teacher_login.html'
224         form_class = TeacherLoginForm
225         context_object_name = 'user_teacher_username'
226         success_url = reverse_lazy('main:teacher-page')
227
228     def form_valid(self, form) -> HttpResponseRedirect:
229
230         username = form.cleaned_data.get('username')
231         password = form.cleaned_data.get('password')
232
233         user = authenticate(username=username, password=password)
234
235         if user:
236             if user.is_teacher:
237                 login(self.request, user)
238                 return super().form_valid(form)
239
240         return self.form_invalid(form)
241
242     def form_invalid(self, form):
243         context = self.get_context_data(form=form)
244         context.update({
245             'error': "Either username or password is invalid."
246         })
247         return self.render_to_response(context)
248
249

```

Figure 179 Teacher Login (View)

```

49 def quiz_view(request, pk):
50     quiz = Quiz.objects.get(pk=pk)
51     return render(request, 'exam_material/quiz.html', {'obj': quiz})
52
53 def quiz_data_view(request, pk):
54
55     quiz = Quiz.objects.get(pk=pk)
56     questions = []
57
58     for q in quiz.get_questions():
59         answers = []
60
61         for a in q.get_answers():
62             answers.append(a.answer_text)
63
64         questions.append({str(q): answers})
65
66     return JsonResponse({
67         'data': questions,
68         'time': quiz.time,
69     })
70
71 def save_quiz_view(request, pk):
72
73     # print(request.POST)
74     if request.is_ajax():
75         questions = []
76         data = request.POST
77         data = dict(data.lists()) #transforms queryDict to ordinary Dict

```

Figure 180 Quiz Main (View)

```

230     # add this 2 lines, if you have can_delete=True parameter
231     # set in inlineformset_factory func
232     for obj in formset.deleted_objects:
233         obj.delete()
234     for answer in answers:
235         answer.question = self.object
236         answer.save()
237
238
239 class QuestionCreate(QuestionInline, generic.CreateView):
240
241     success_url = reverse_lazy('exam_material:question-create')
242
243     def get_context_data(self, **kwargs):
244         ctx = super(QuestionCreate, self).get_context_data(**kwargs)
245         ctx['named_formsets'] = self.get_named_formsets()
246         return ctx
247
248     def get_named_formsets(self):
249         if self.request.method == "GET":
250             return {
251                 'answers': AnswerFormSet(prefix='answers'),
252             }
253         else:
254             return {
255                 'answers': AnswerFormSet(self.request.POST or None, self.request.FILES or None, prefix='answers'),
256             }
257
258

```

Figure 181 Create Question (View)

```

72     def save_quiz_view(request, pk):
73
74         # print(request.POST)
75         if request.is_ajax():
76             questions = []
77             data = request.POST
78             data_= dict(data.lists()) #transforms QueryDict to ordinary Dict
79
80             data_.pop('csrfmiddlewaretoken')
81
82             for k in data_.keys():
83                 # print('key: ', k)
84                 question = Question.objects.get(title=k)
85                 questions.append(question)
86             # print(questions)
87
88             user = request.user
89             quiz = Quiz.objects.get(pk=pk)
90
91             score = 0
92             multiplier = 100/ quiz.numberOfQuestions
93             results = []
94             correct_answer = None

```

Figure 182 Save Quiz (View)

```

79
80     def load_level_section(request):
81         level_id = request.GET.get('level_id')
82         sections = LevelSection.objects.filter(level_id=level_id)
83         return render(request, 'account/level_section.html', {'sections': sections})
84
85
86

```

Figure 183 Section selection (View)

```

123 #For student
124 class StudentList(generic.ListView):
125     template_name = 'main/student_list.html'
126     model = Student
127     context_object_name = 'student_list'
128     queryset = Student.objects.all()
129
130 class StudentDetail(generic.DetailView):
131     template_name = 'main/student_detail.html'
132     model = Student
133     context_object_name = 'student_detail'
134     queryset = Student.objects.all()
135
136 class StudentUpdate(generic.UpdateView):
137     template_name = 'main/student_update.html'
138     form_class = StudentUpdateForm
139     model = User
140     context_object_name = 'student_update'
141     success_url = reverse_lazy('main:student-page')
142
143
144 class StudentDelete(generic.DeleteView):
145     template_name = 'main/student_delete.html'
146     model = User
147     context_object_name = 'student_delete'
148     success_url = reverse_lazy('main:student-list')
149
150

```

Figure 184 Generic View

```

99     return super().form_valid(form)
100
101 #Class registration
102 class LevelRegistration(generic.CreateView):
103     template_name = 'account/level_registration_form.html'
104     form_class = LevelRegistrationForm
105     model = Level
106     success_url = reverse_lazy('account:level-registration')
107
108     def form_valid(self, form) -> HttpResponseRedirect:
109         self.object = form.save()
110         level = self.object.level
111         messages.success(self.request, f'Class {level} was registered successfully.')
112
113     return super().form_valid(form)
114
115

```

Figure 185 Class Registration (View)

```
287
288
289 < class UserLogout(generic.View):
290
291     def logout_request(self, request):
292         logout(request)
293         # user = self.object.user
294         messages.success(self.request, f' you have been logged out successfully.')
295         return redirect('main:index')
296
297
```

Figure 186 User Logout (View)

7.4. Designs:

7.4.1. Gantt Chart:

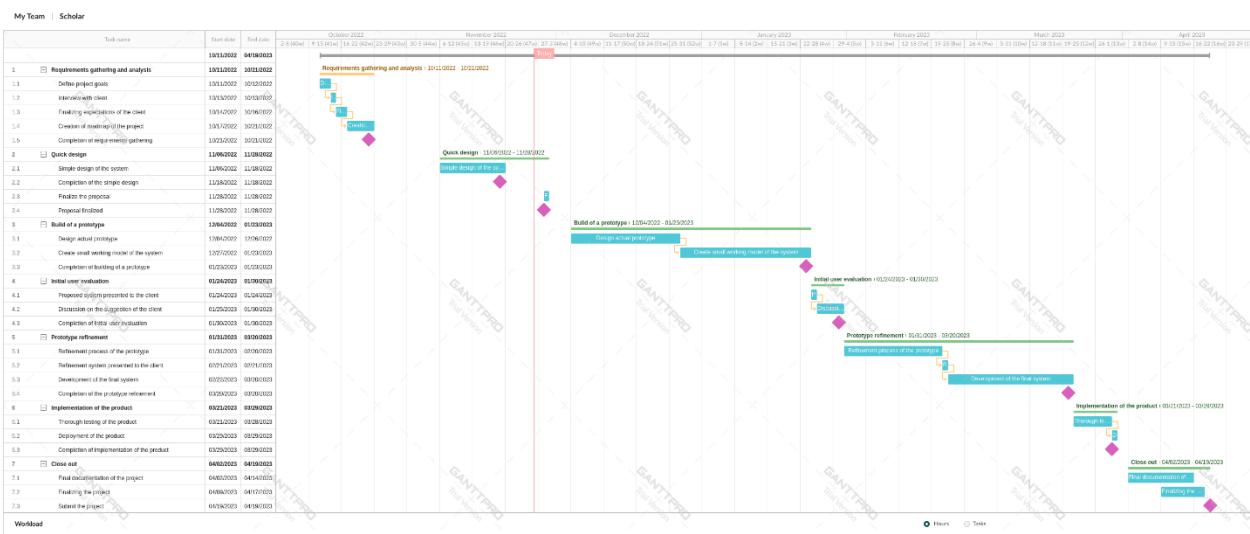


Figure 187 Gantt Chart (full view)

My Team | Scholar

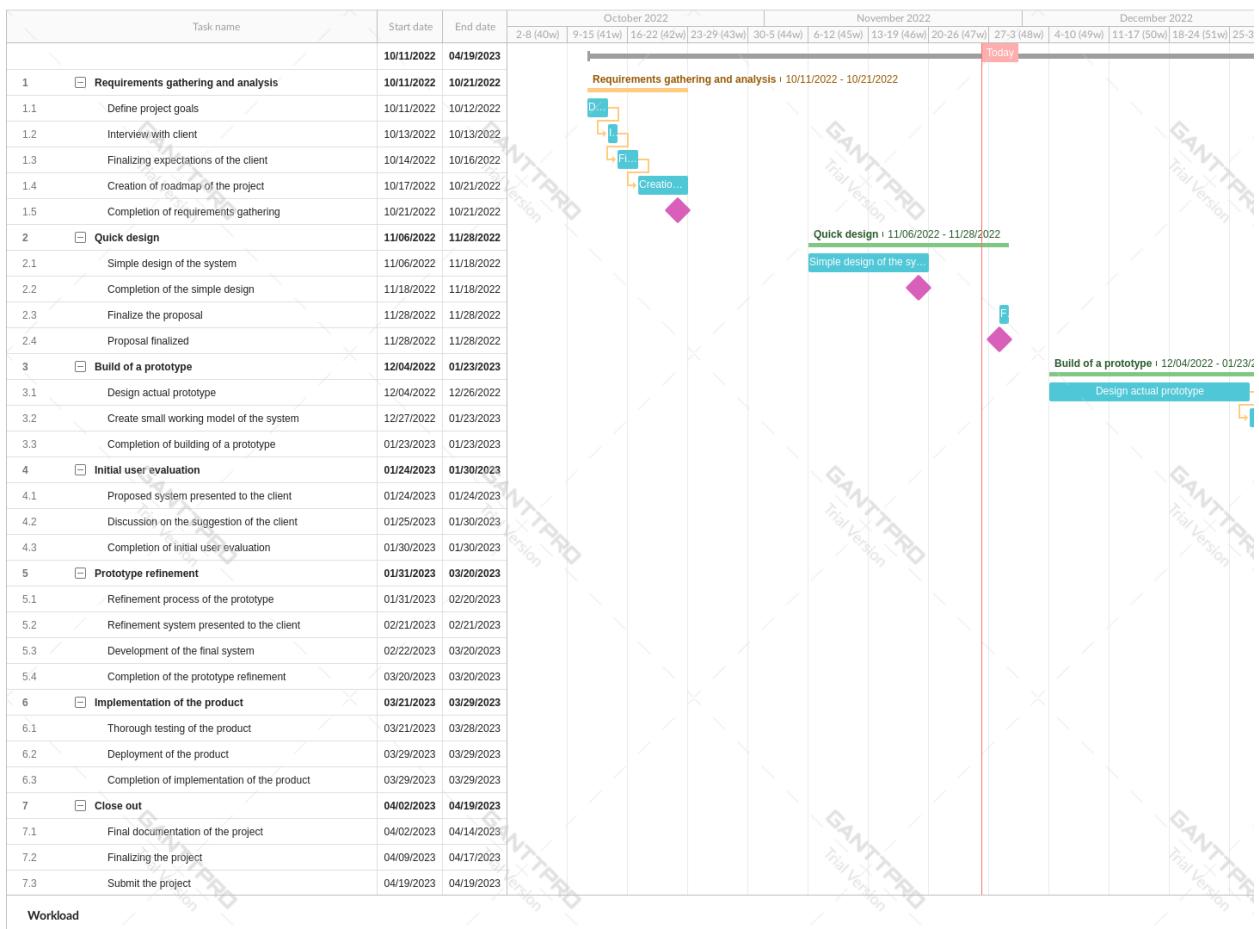


Figure 188 Gantt chart (Larger view 1)

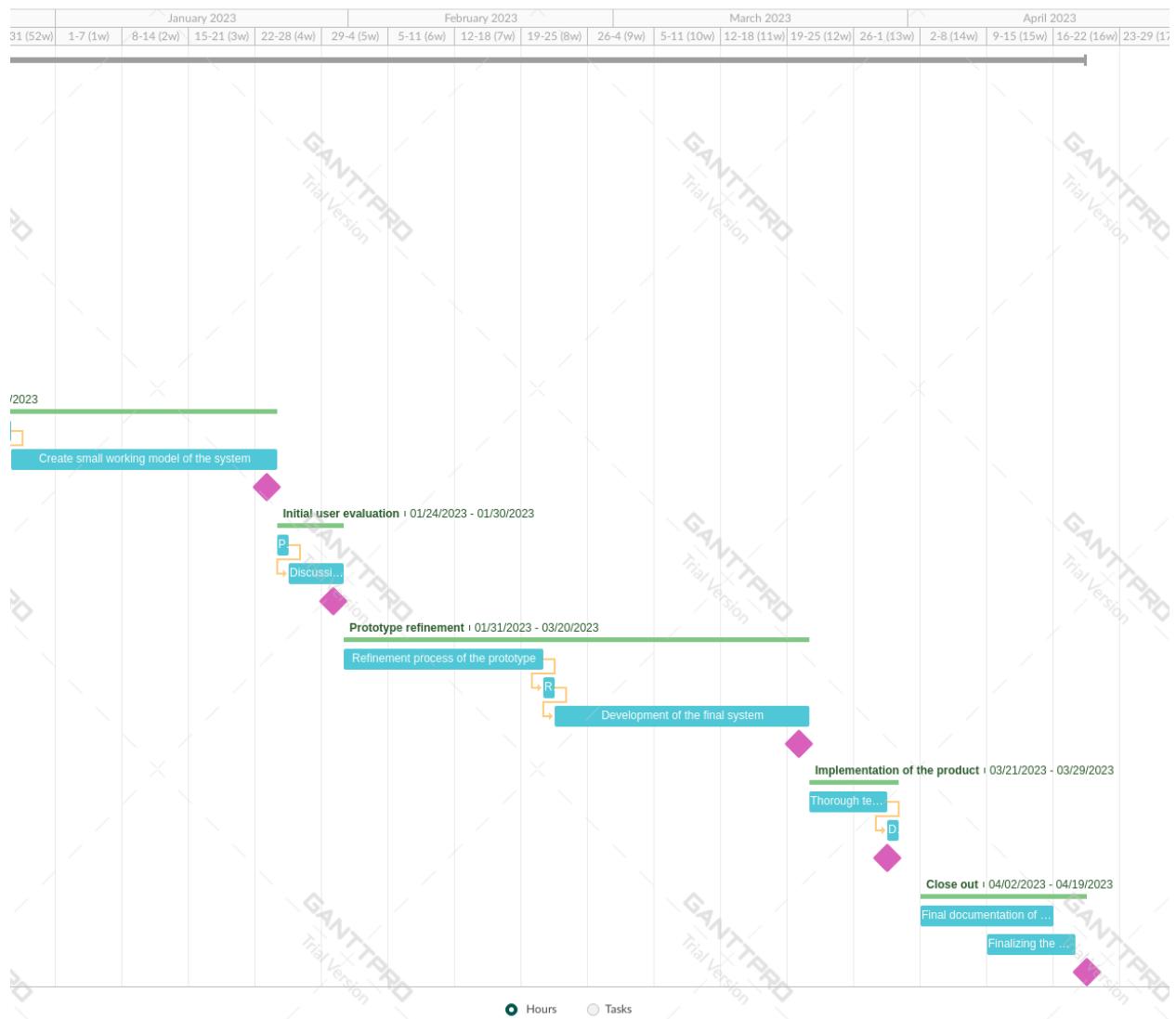


Figure 189 Gantt chart (Larger view 2)

7.4.2. Work Breakdown Structure:

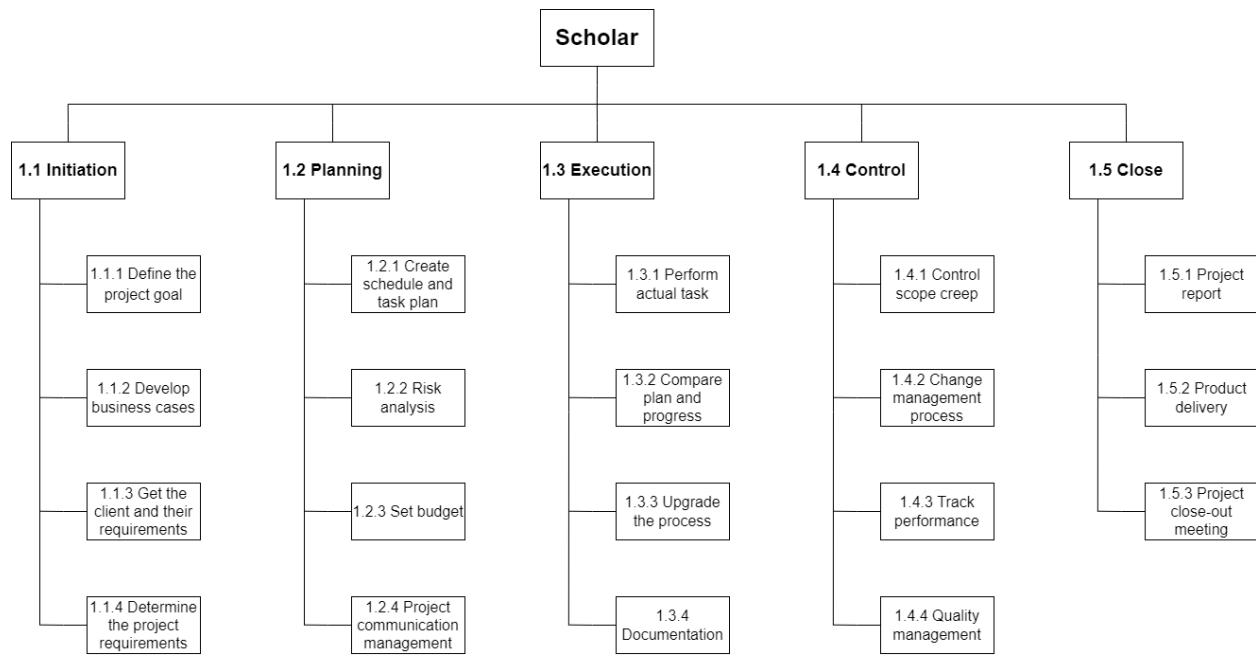


Figure 190 Work Breakdown Structure

7.4.3. Milestone:

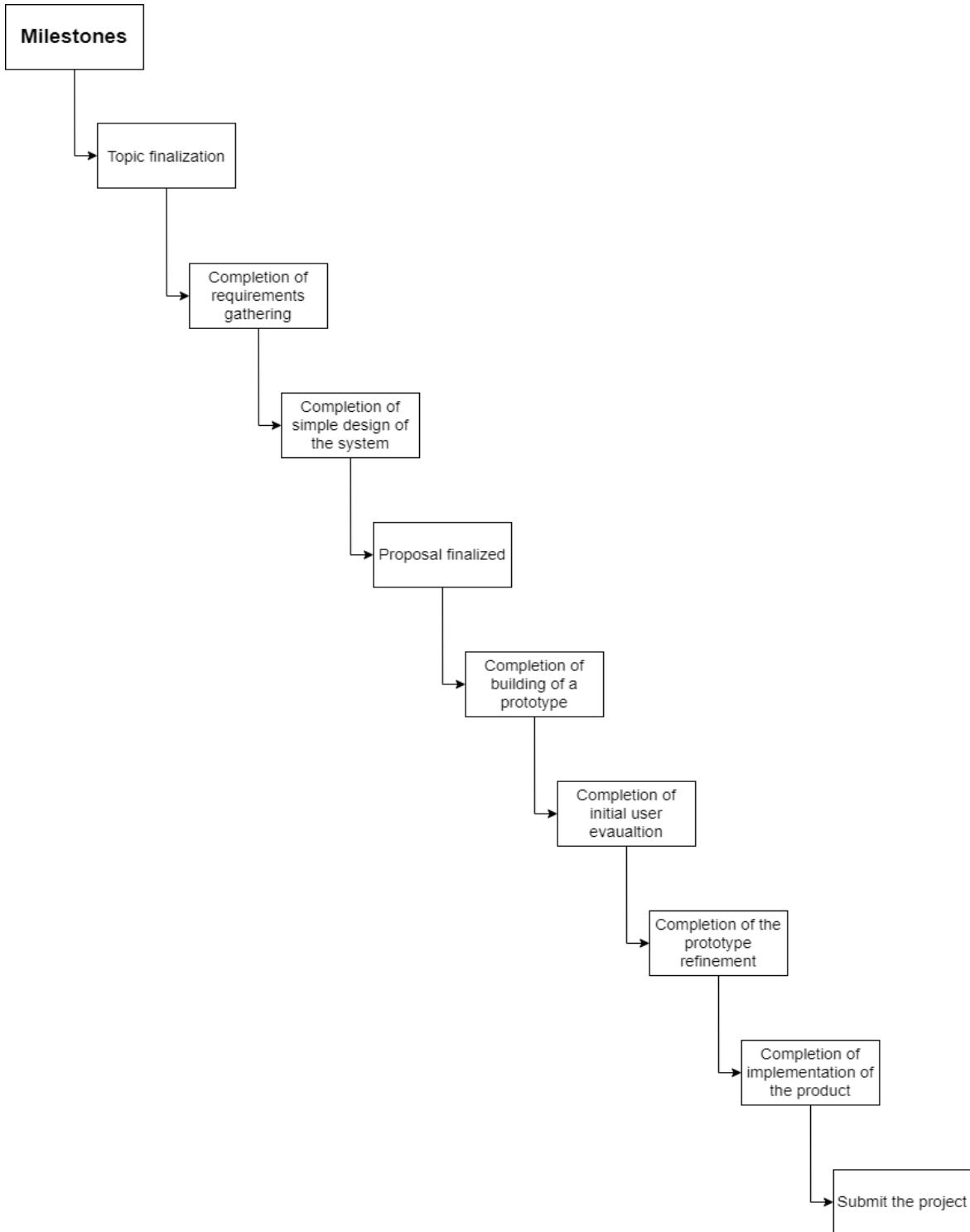


Figure 191 Milestone

7.4.4. Wireframe:

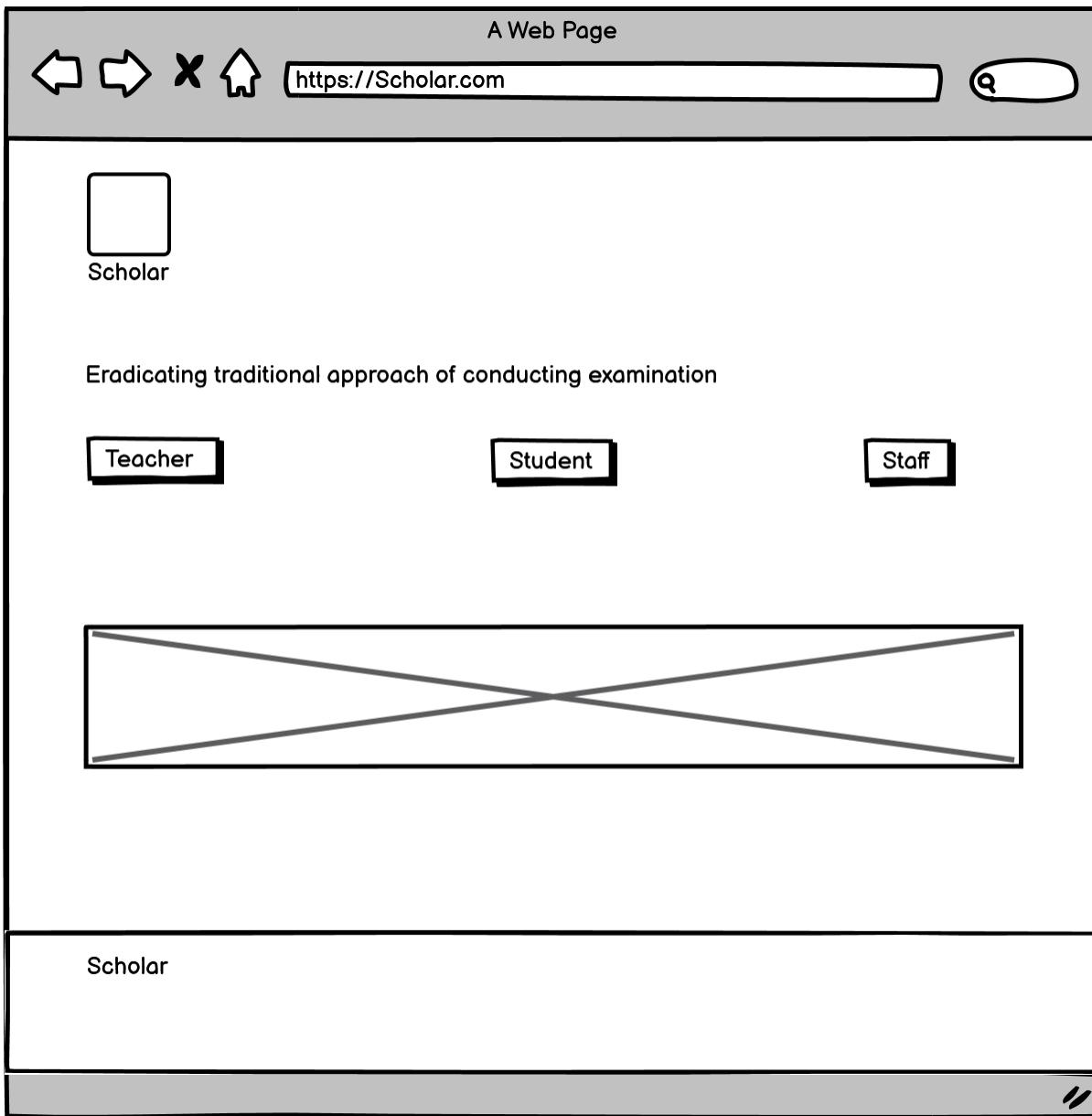


Figure 192 Home page (Wireframe)

A Web Page

https://student-registration

Scholar

Student Registration

First Name _____

Last Name _____

Username _____

Email _____

Password _____

Password Confirmation _____

Student ▾

Subjects ▾

Class ▾

Register

Figure 193 Student Registration (Wireframe)

A Web Page

https://staff-registration

Scholar

Staff Registration

First Name <hr/>	Last Name <hr/>
Username <hr/>	Email <hr/>
Password <hr/>	Password Confirmation <hr/>
Student ▼	Position ▼

Register

A wireframe diagram of a web browser window titled "A Web Page". The address bar shows the URL "https://staff-registration". Below the title bar, there are standard browser icons for back, forward, stop, and search. The main content area displays a "Scholar" logo and a "Staff Registration" heading. A large rectangular form is centered, containing fields for First Name, Last Name, Username, Email, Password, Password Confirmation, a dropdown menu for Student status, another dropdown menu for Position, and a prominent "Register" button at the bottom.

Figure 194 Staff Registration (Wireframe)

A Web Page

https://teacher-registration

Scholar

Teacher Registration

First Name <hr/>	Last Name <hr/>
Username <hr/>	Email <hr/>
Password <hr/>	Password Confirmation <hr/>
Student ▼	Subjects ▼
Class ▼	
Register	

//

Figure 195 Teacher Registration (Wireframe)

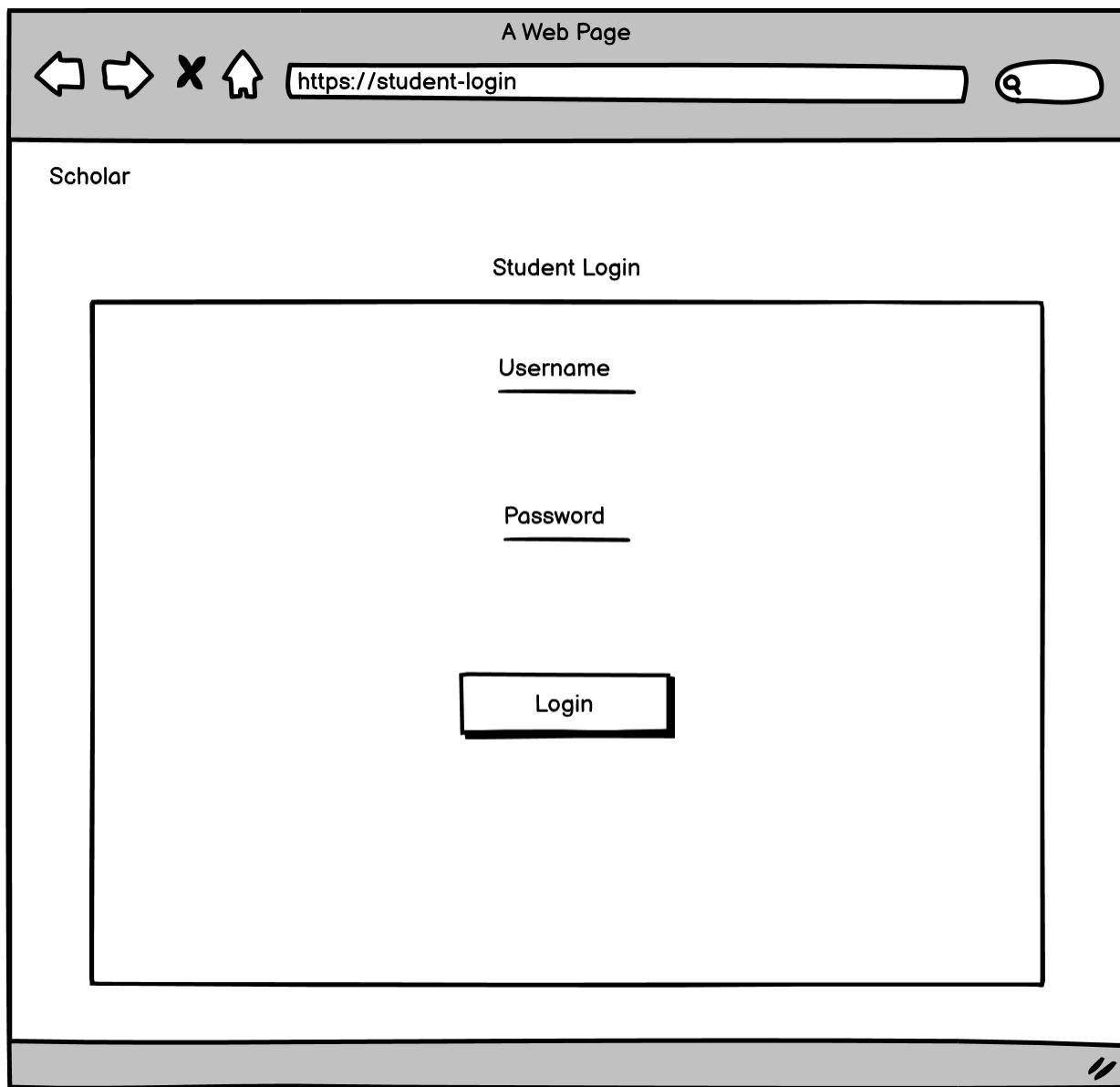


Figure 196 Student Login (Wireframe)

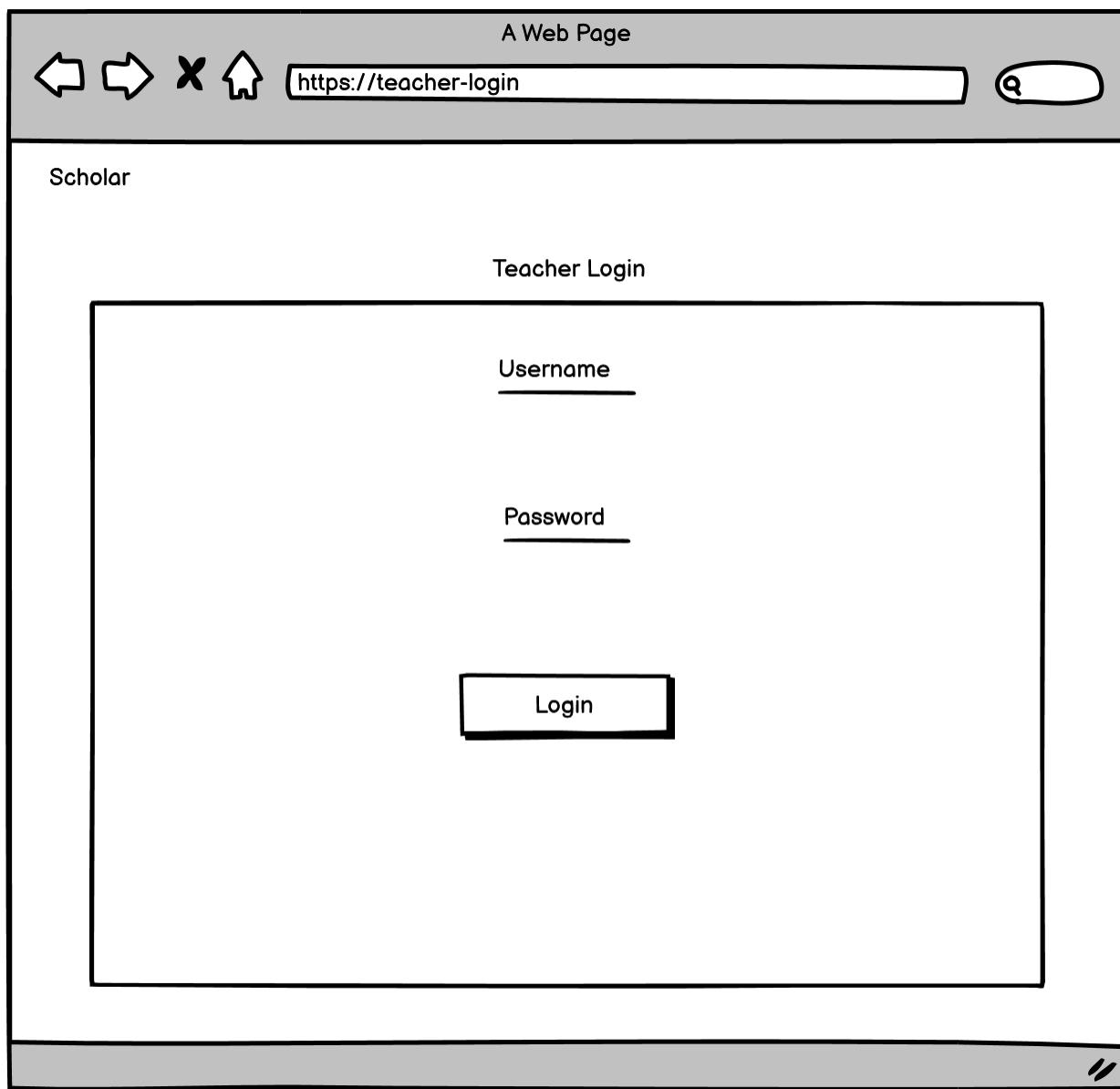


Figure 197 Teacher Login (Wireframe)

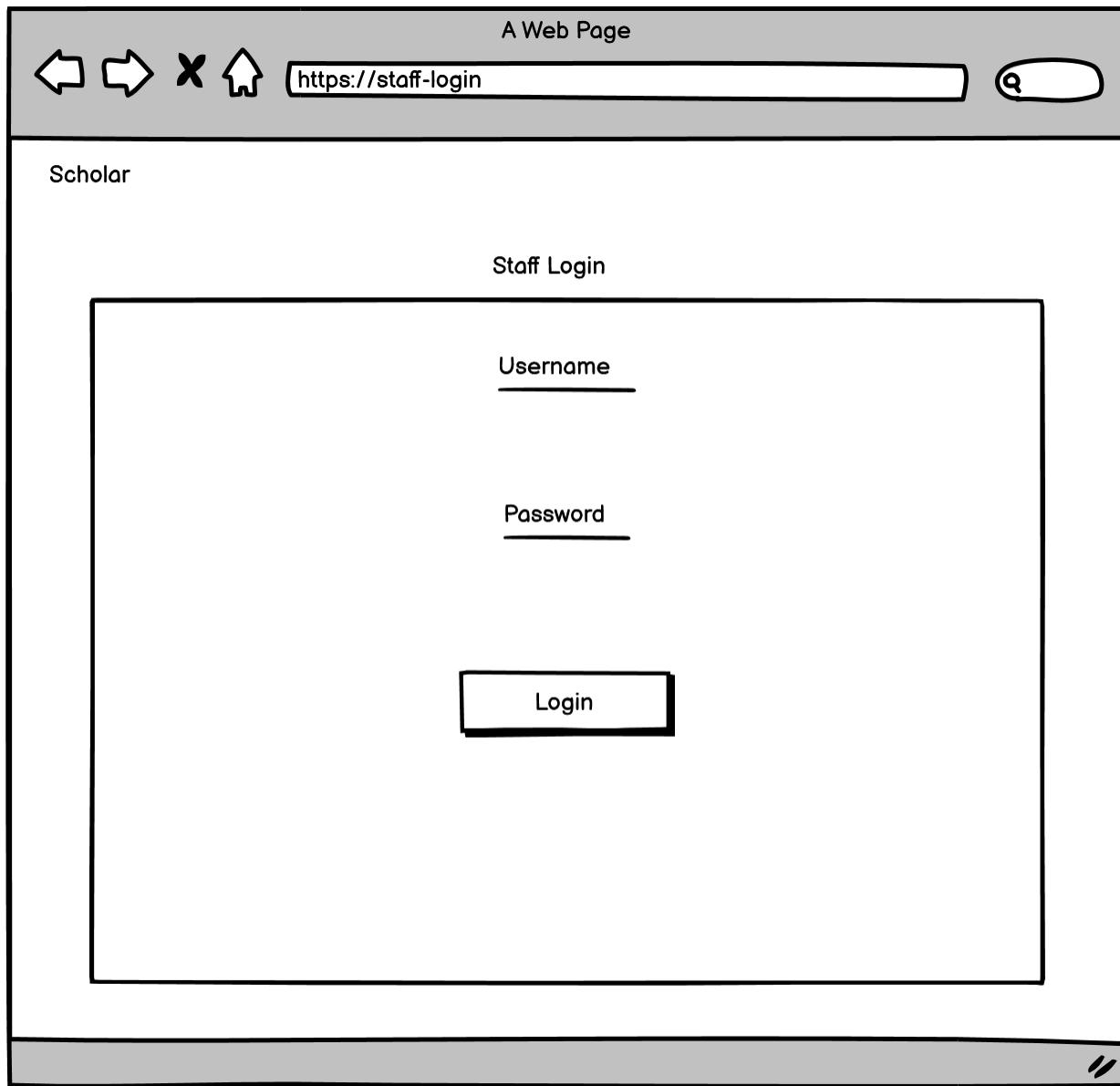


Figure 198 Staff Login (Wireframe)

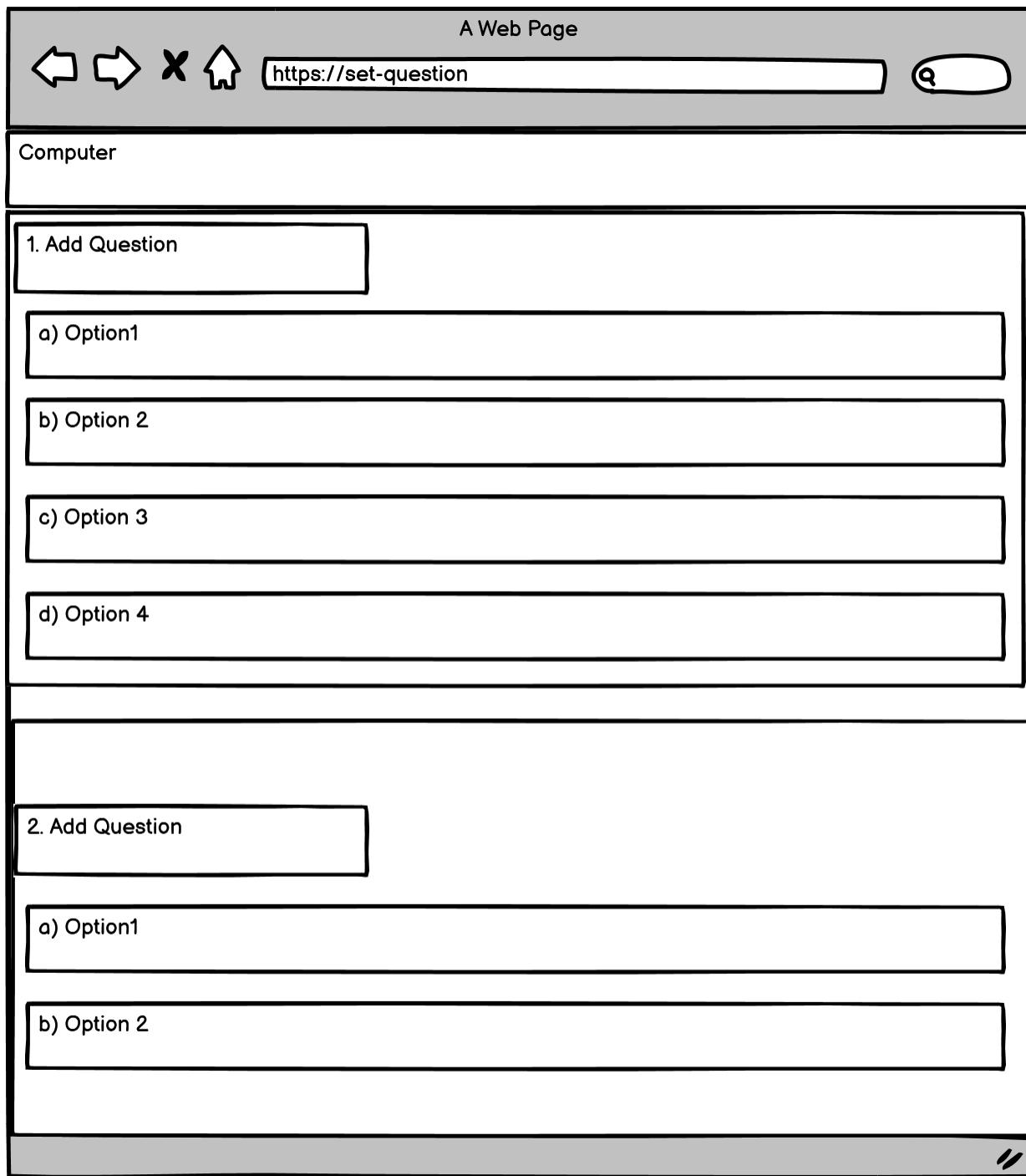


Figure 199 Set questions (Wireframe)

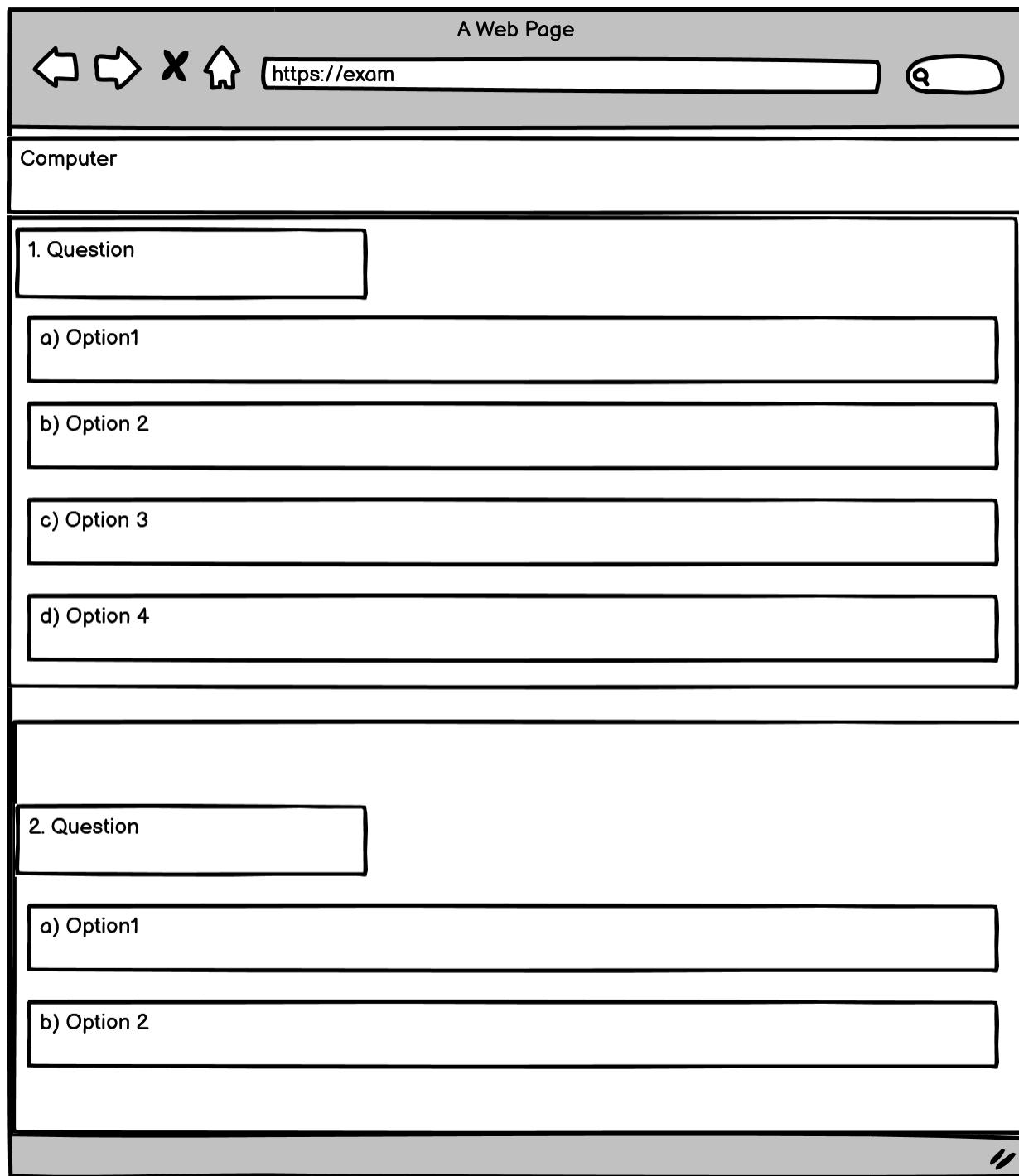


Figure 200 Take the exam (Wireframe)

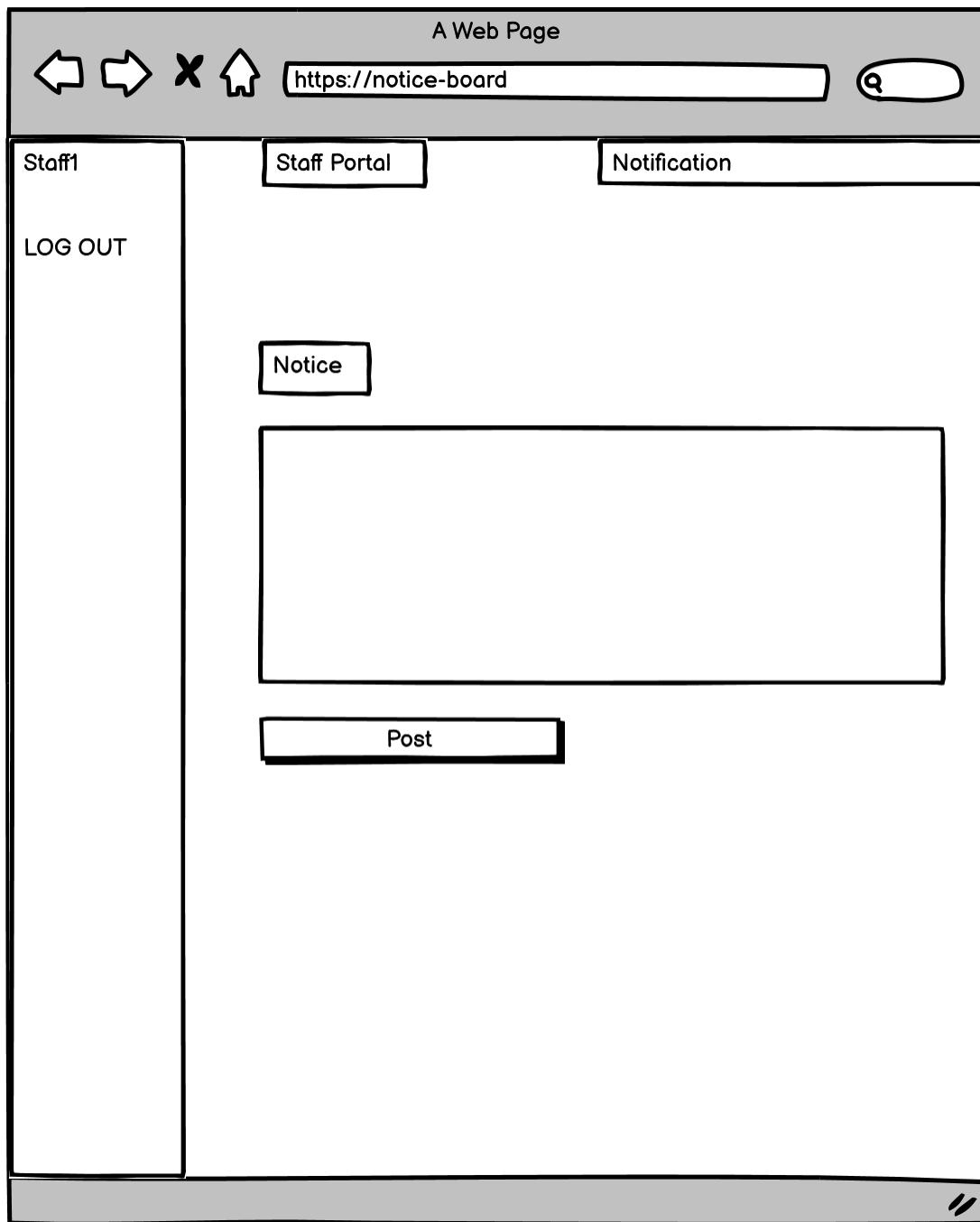


Figure 201 Notice Board (Wireframe)

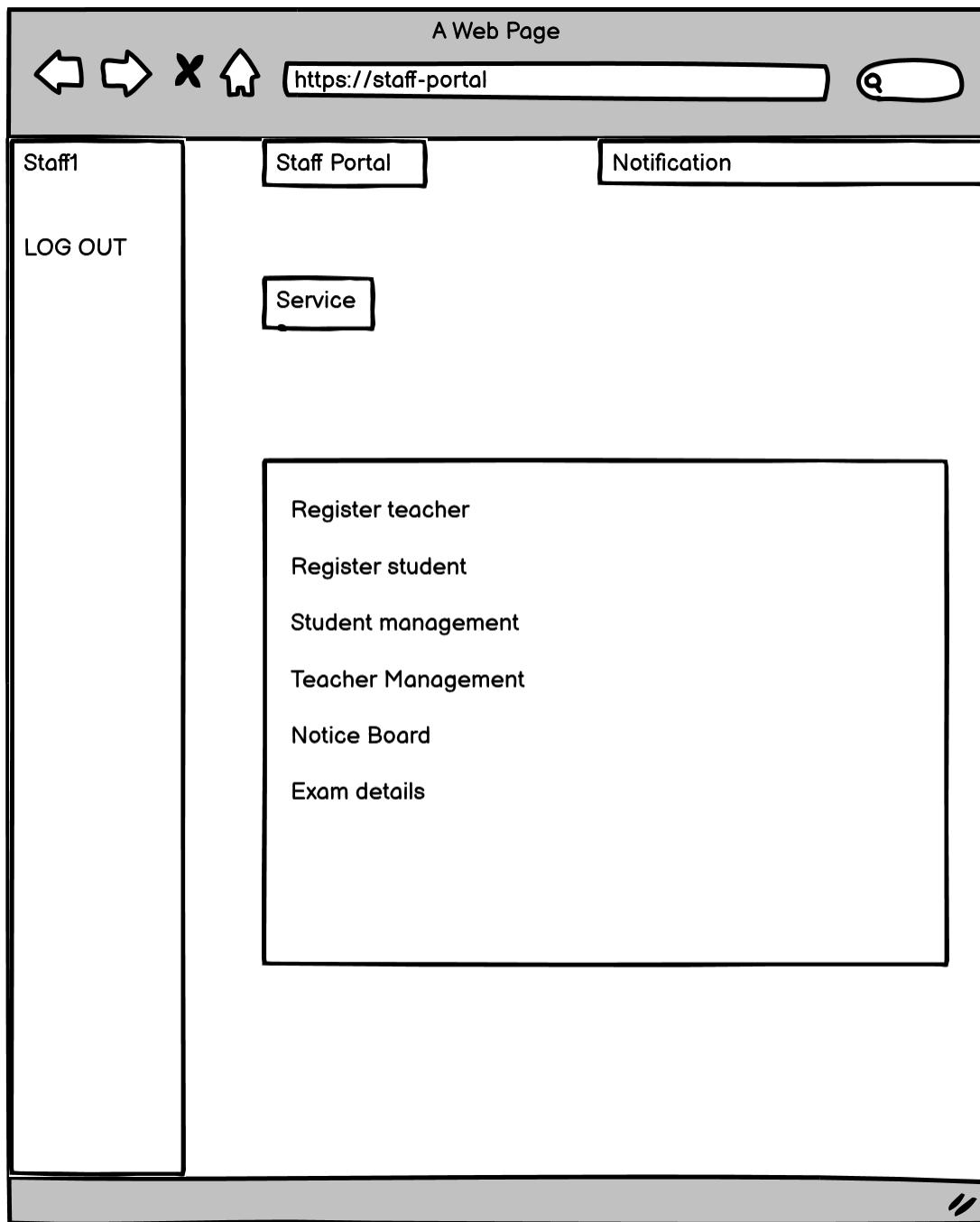


Figure 202 Staff portal (Wireframe)

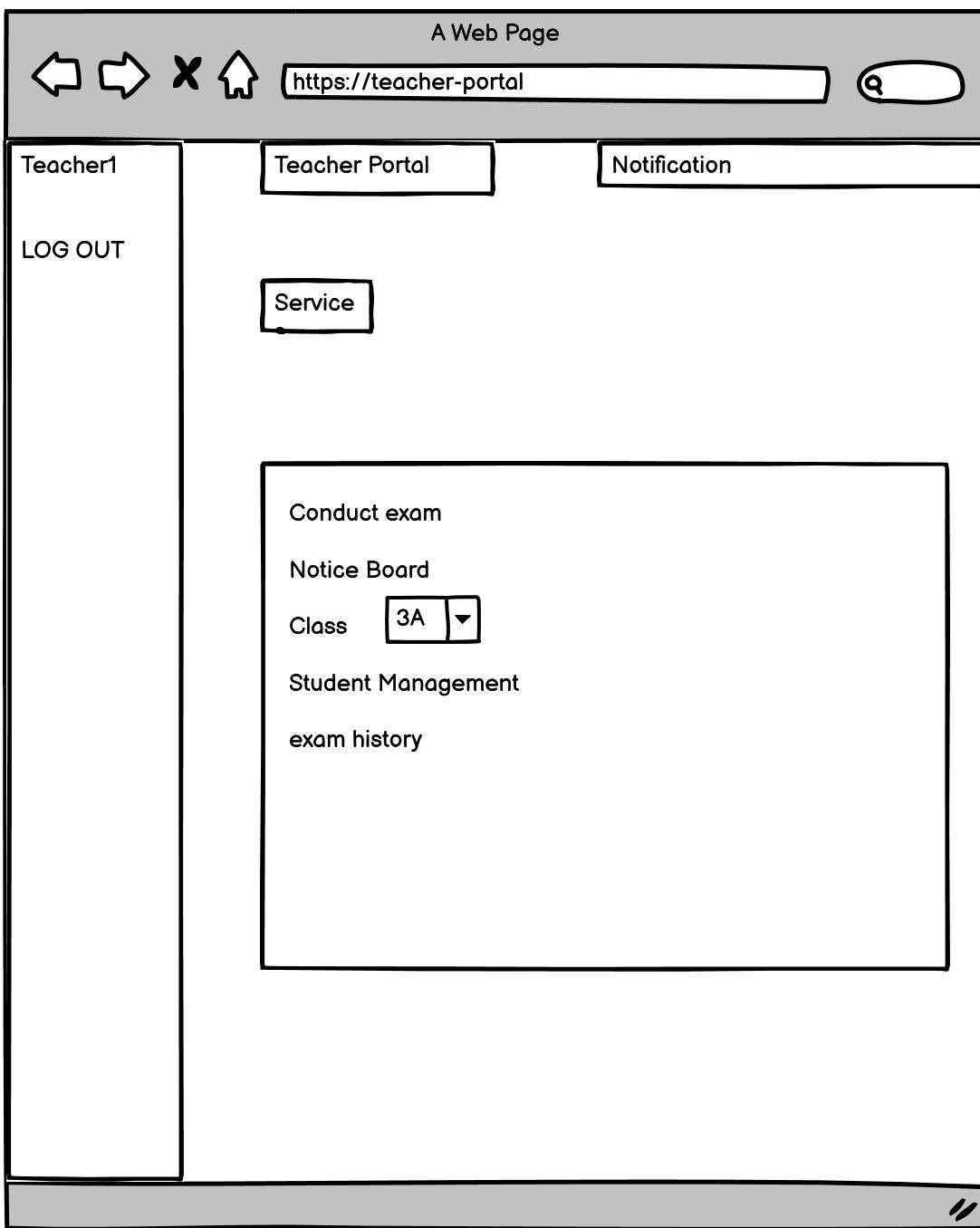


Figure 203 Teacher Portal (Wireframe)

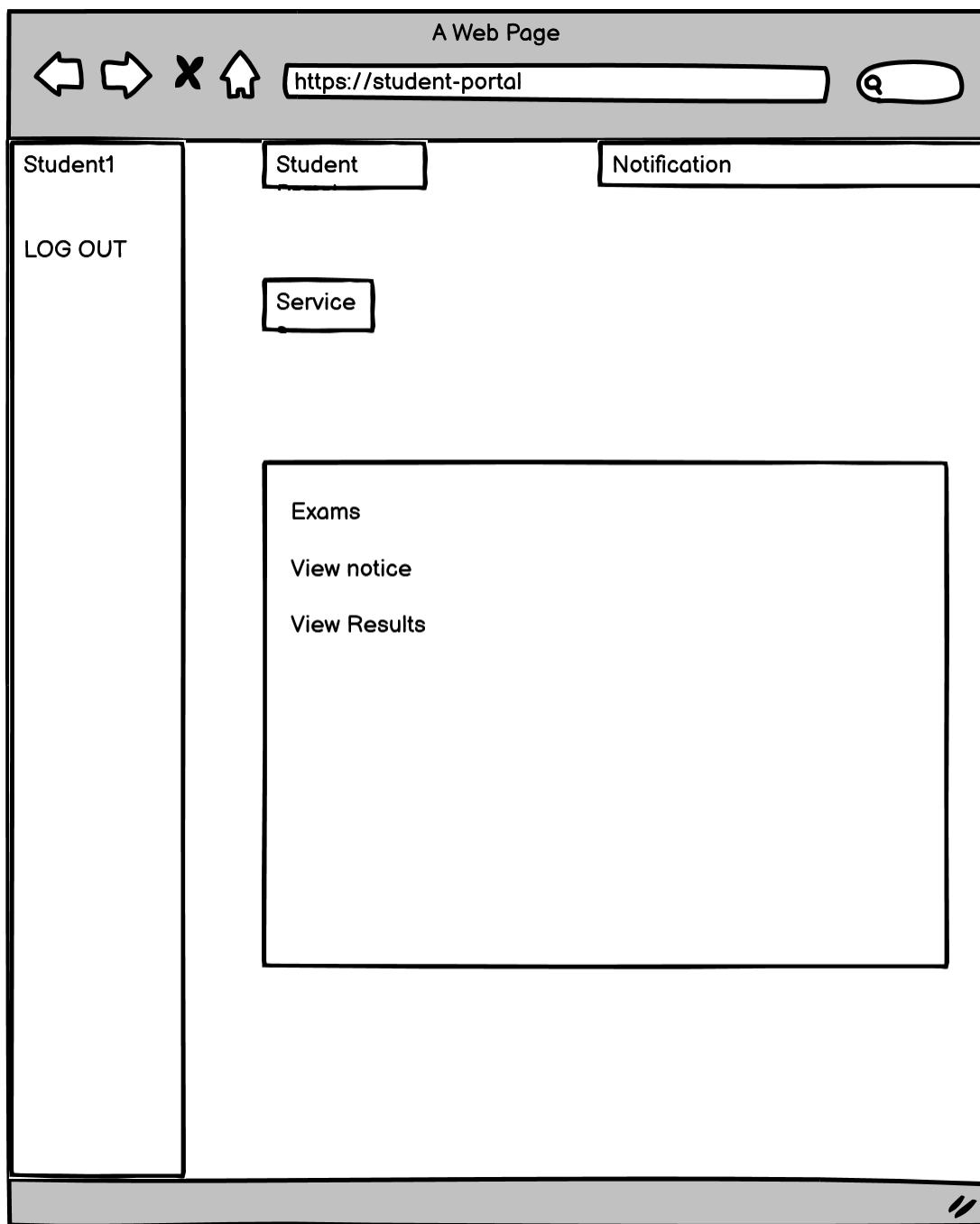


Figure 204 Student Portal (Wireframe)

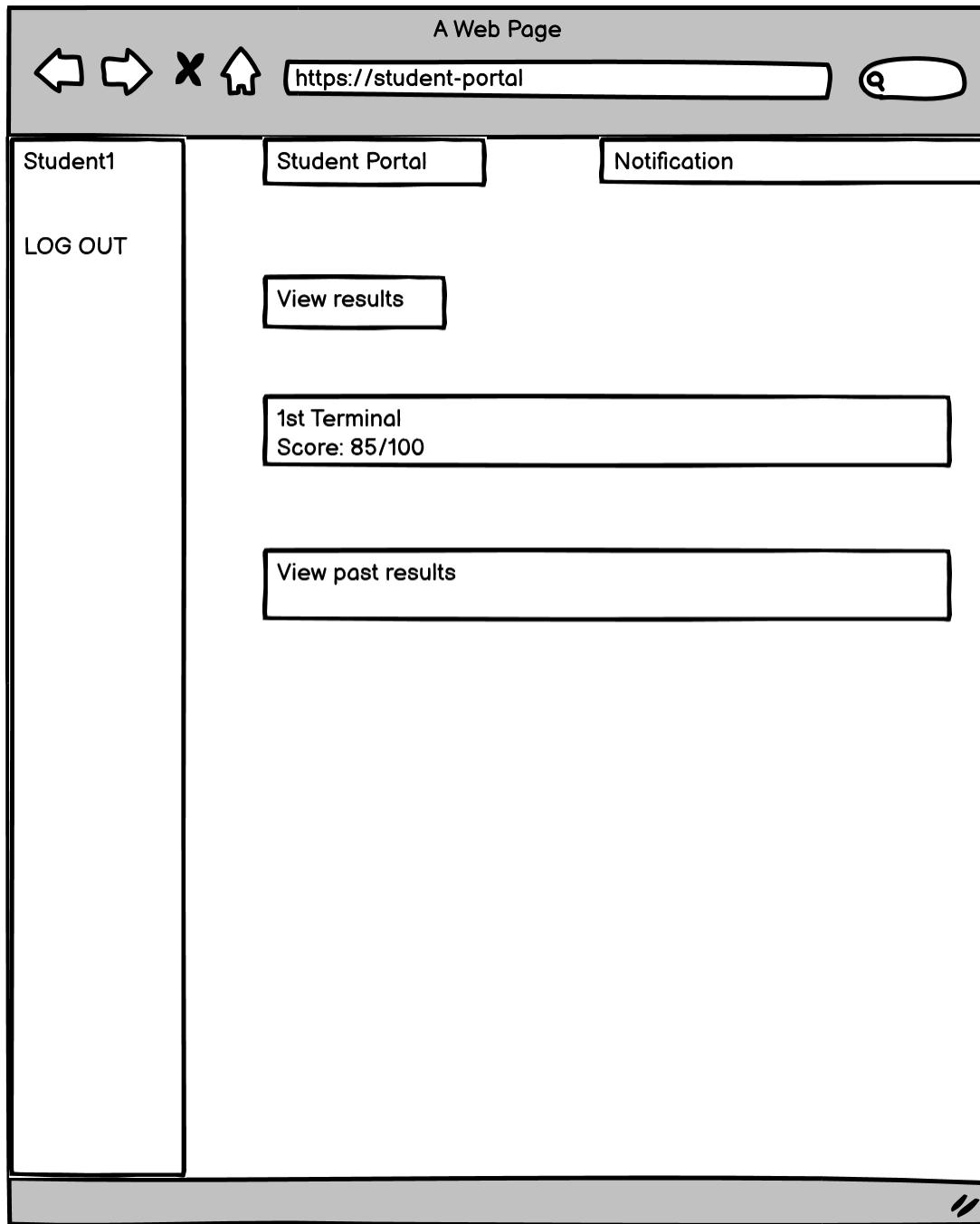
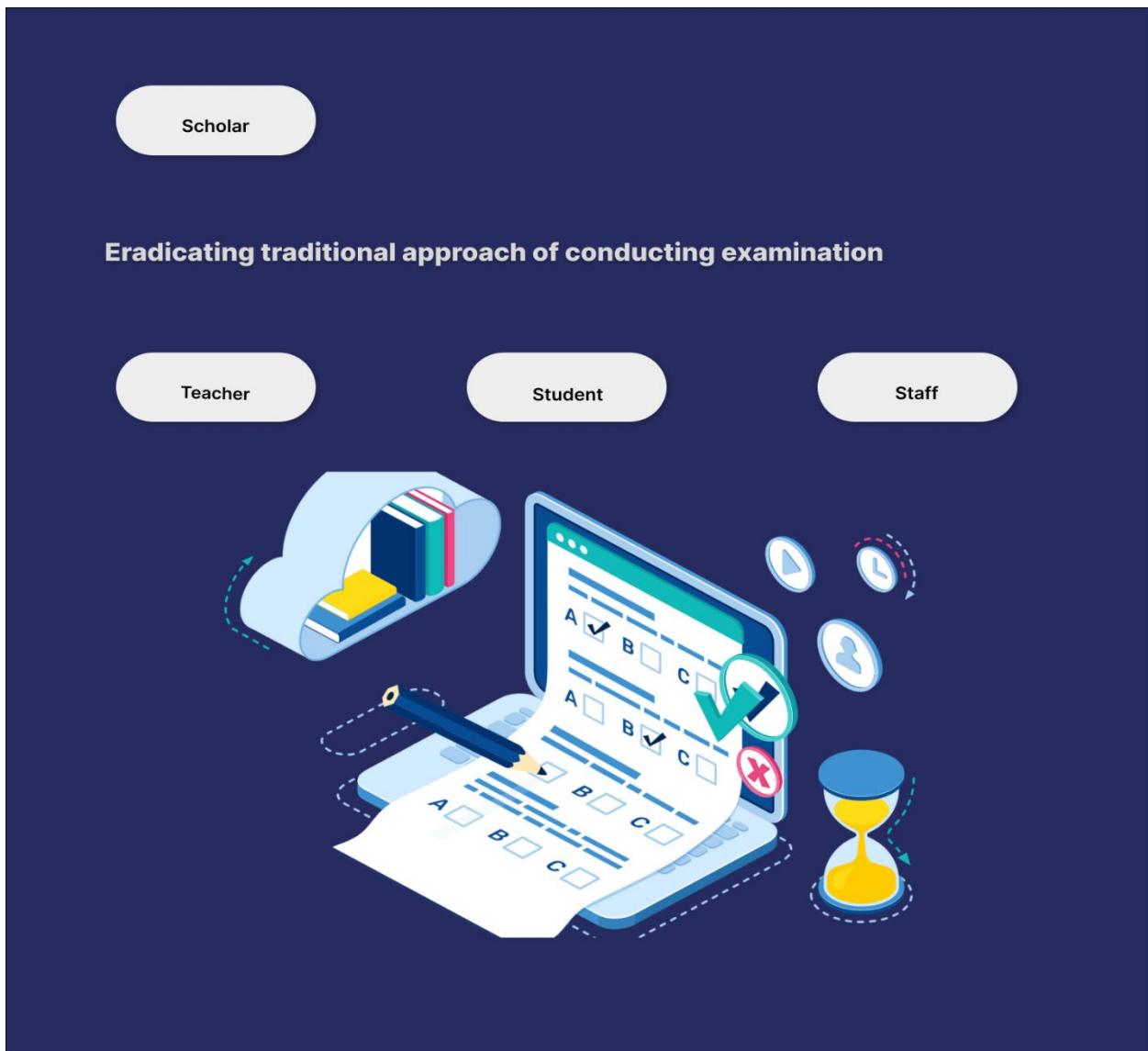


Figure 205 Student Portal view results (Wireframe)

7.4.5. Prototype (UI/UX (Graphic designing/Page layout)):



Scholar

INFORMATION

QUICK LINKS

CONTACT US

Eradicating traditional approach
of conducting examination

[ABOUT US](#)
[FAQ](#)

[SERVICES](#)
[ABOUT US](#)

+2727727272

COPYRIGHT 2022

ALL RIGHT RESERVED

Figure 206 Home page (Prototype design)

Scholar

Student Registration

The image shows a prototype design for a 'Student Registration' form. The form is contained within a white rectangular box with rounded corners, set against a dark blue background. The fields are arranged in two columns. The left column contains 'First Name' and 'Username' input fields, followed by 'Password' and 'Role' fields, each with a small downward arrow icon indicating a dropdown menu. The right column contains 'Last Name' and 'Email' input fields, followed by 'Passwrod confirmation' and 'Subjects' fields, also each with a small downward arrow icon. At the bottom left of the form is a dark blue button labeled 'Register'.

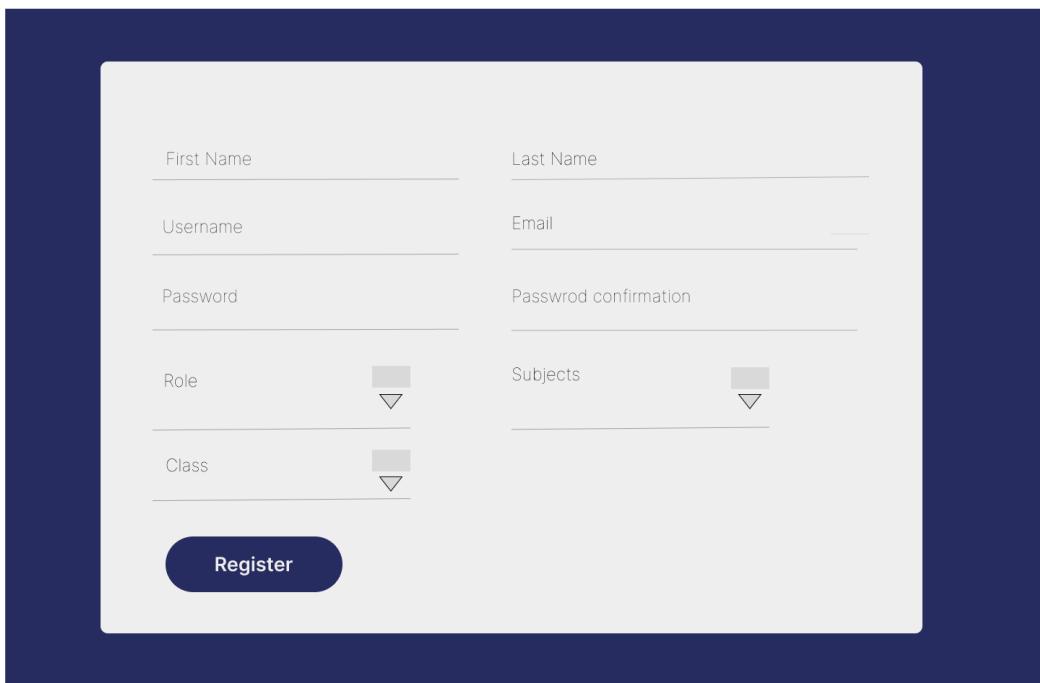
First Name	Last Name
Username	Email
Password	Passwrod confirmation
Role	Subjects
Class	

Register

Figure 207 Student Registration (Prototype design)

Scholar

Teacher Registration



The image shows a prototype design for a Teacher Registration form. The form is contained within a white rectangular box with rounded corners, set against a dark blue background. The fields are arranged in two columns. The left column contains 'First Name' and 'Username' input fields, followed by 'Password' and 'Role' dropdown menus. The right column contains 'Last Name' and 'Email' input fields, followed by 'Passwrod confirmation' and 'Subjects' dropdown menus. A large blue 'Register' button is positioned at the bottom center of the form.

First Name	Last Name
Username	Email
Password	Passwrod confirmation
Role	Subjects
Class	
Register	

Figure 208 Teacher Registration (Prototype design)

Scholar

Staff Registration

The image shows a prototype design for a 'Staff Registration' form. The form is contained within a white rectangular box with rounded corners, set against a dark blue background. The fields are arranged in two columns. The left column contains 'First Name' and 'Username' inputs, followed by 'Password' and 'Role' inputs. The right column contains 'Last Name' and 'Email' inputs, followed by 'Passwrod confirmation' and 'Position' inputs. Each input field has a small gray downward arrow icon to its right. A large blue 'Register' button is positioned at the bottom left of the form area.

First Name	Last Name
Username	Email
Password	Passwrod confirmation
Role	Position

Register

Figure 209 Staff Registration (Prototype design)

Scholar

Student Login

Username _____

Password _____
•••••••• 

LOGIN

Figure 210 Student Login (Prototype design)

Scholar

Teacher Login

Username _____

Password _____
•••••••• 

LOGIN

Figure 211 Teacher Login (Prototype design)

Scholar

Staff Login

Username —————

Password —————
•••••••• 

LOGIN

Figure 212 Staff Login (Prototype design)

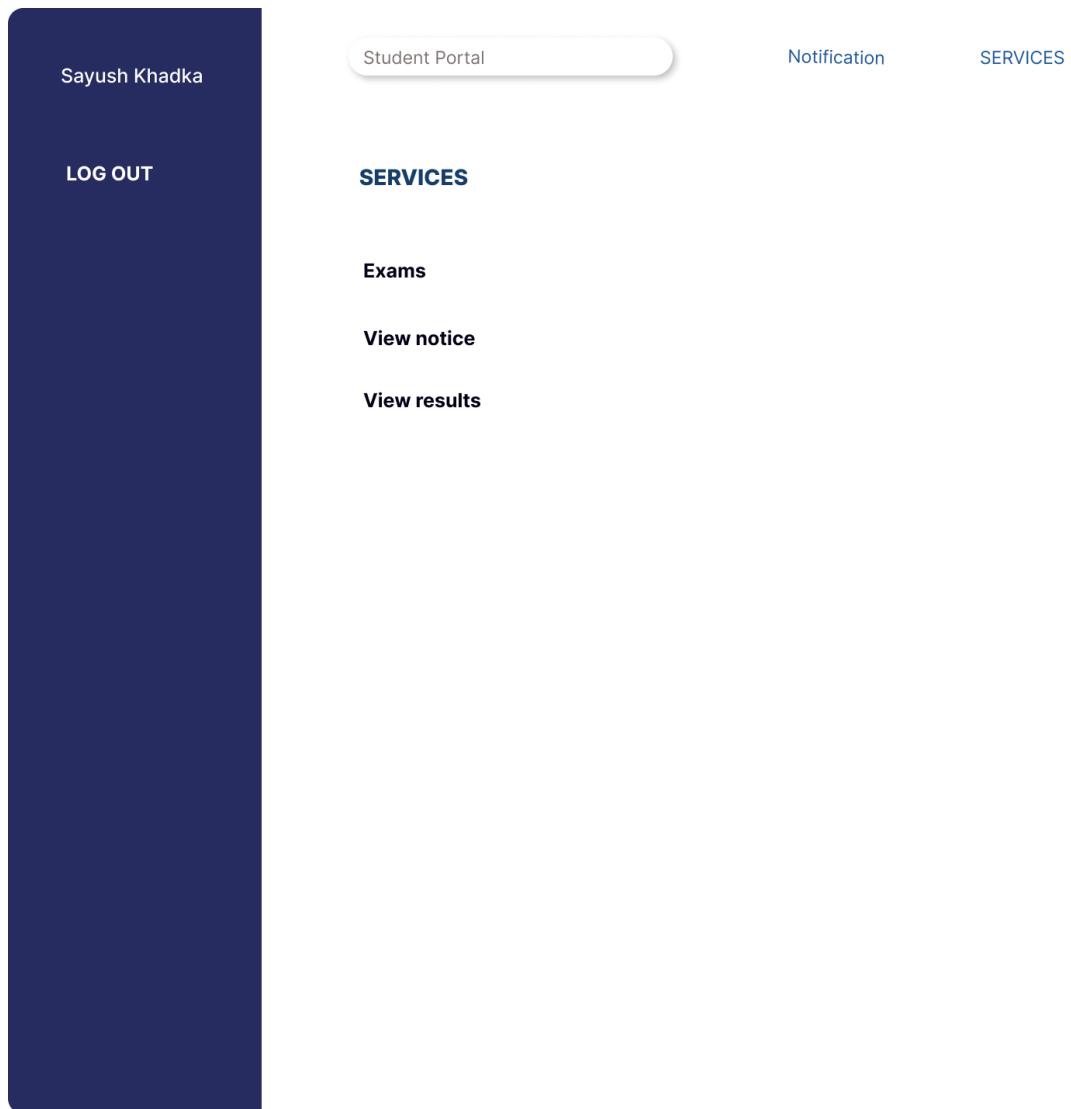


Figure 213 Student Portal (Prototype design)

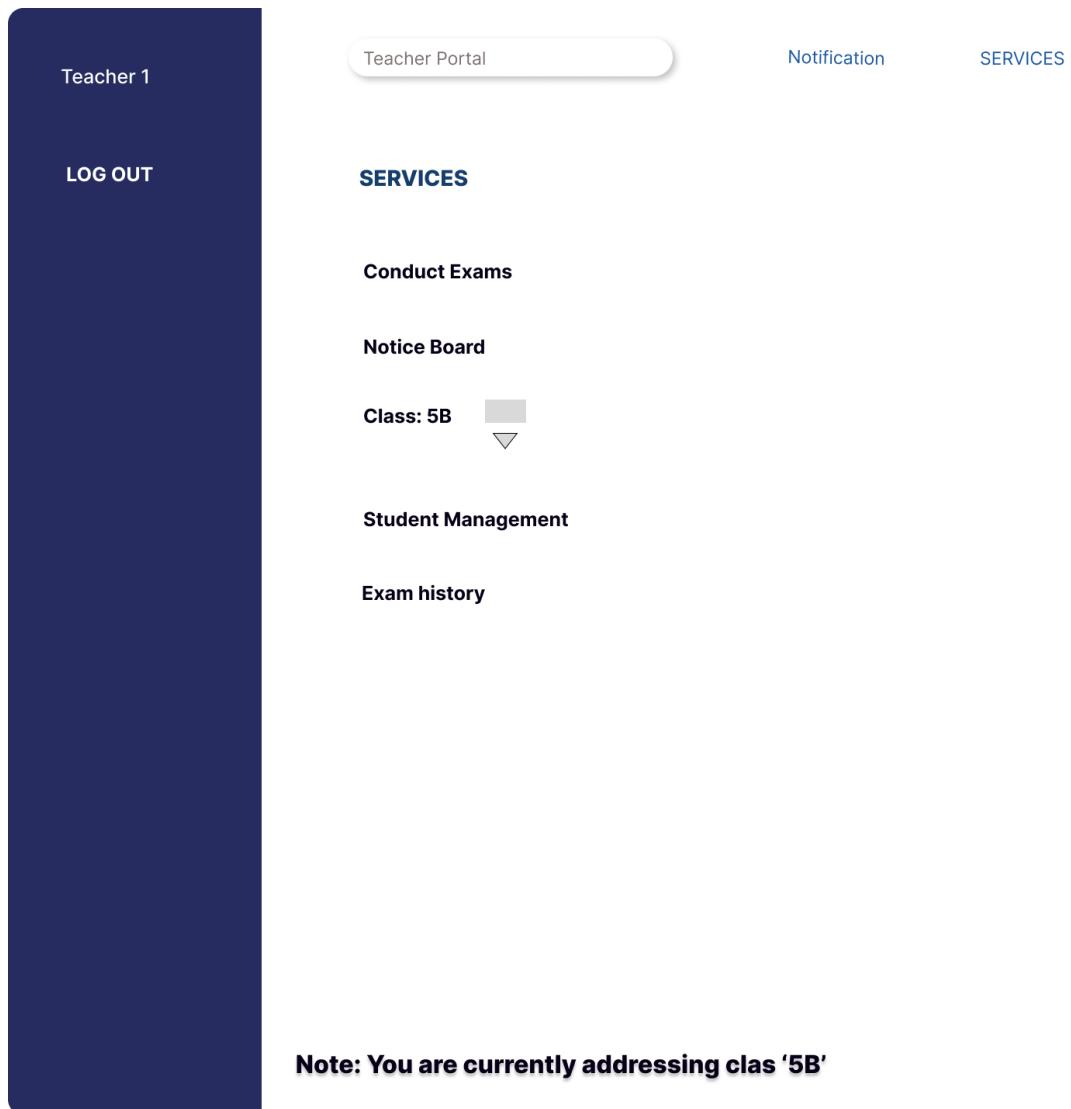


Figure 214 Teacher Portal (Prototype design)

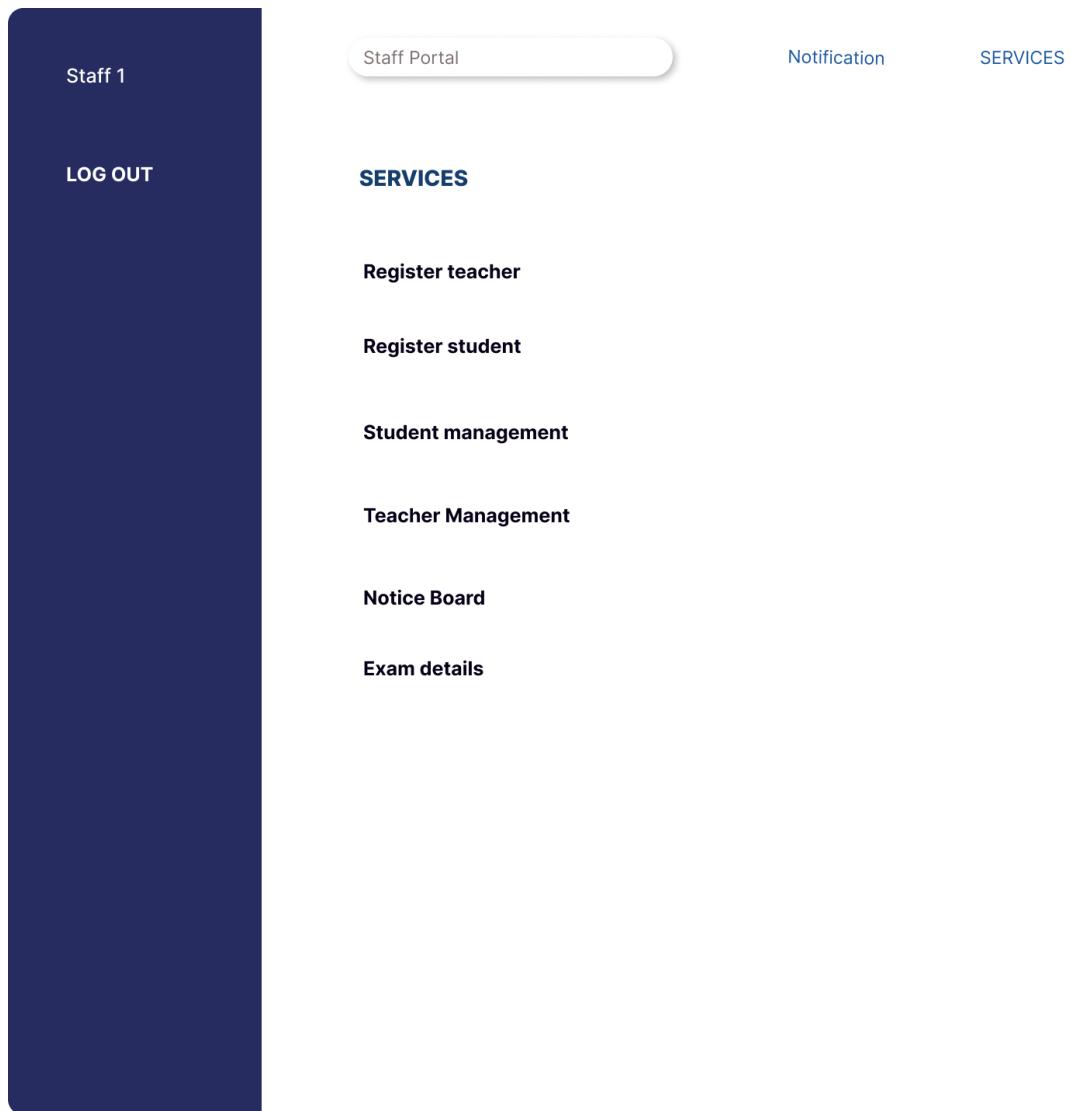


Figure 215 Staff Portal (Prototype design)

Computer

1st Terminal

1. Who was the father of computer

 Michael Faraday Charles Babbage Adam Smith Add answer

2. When was the first computer made?

 1945 AD 1500 AD 1290 AD None of the above

Figure 216 Conduct examination (Prototype design)

Computer

1. Who was the father of computer

Michael Faraday

Charles Babbage

Adam Smith

Philo Taylor Farnsworth II

2. When was the first computer made??

1945 AD

1500 AD

1290 AD

None of the above

Finish Test

Figure 217 Take examination (Prototype design)

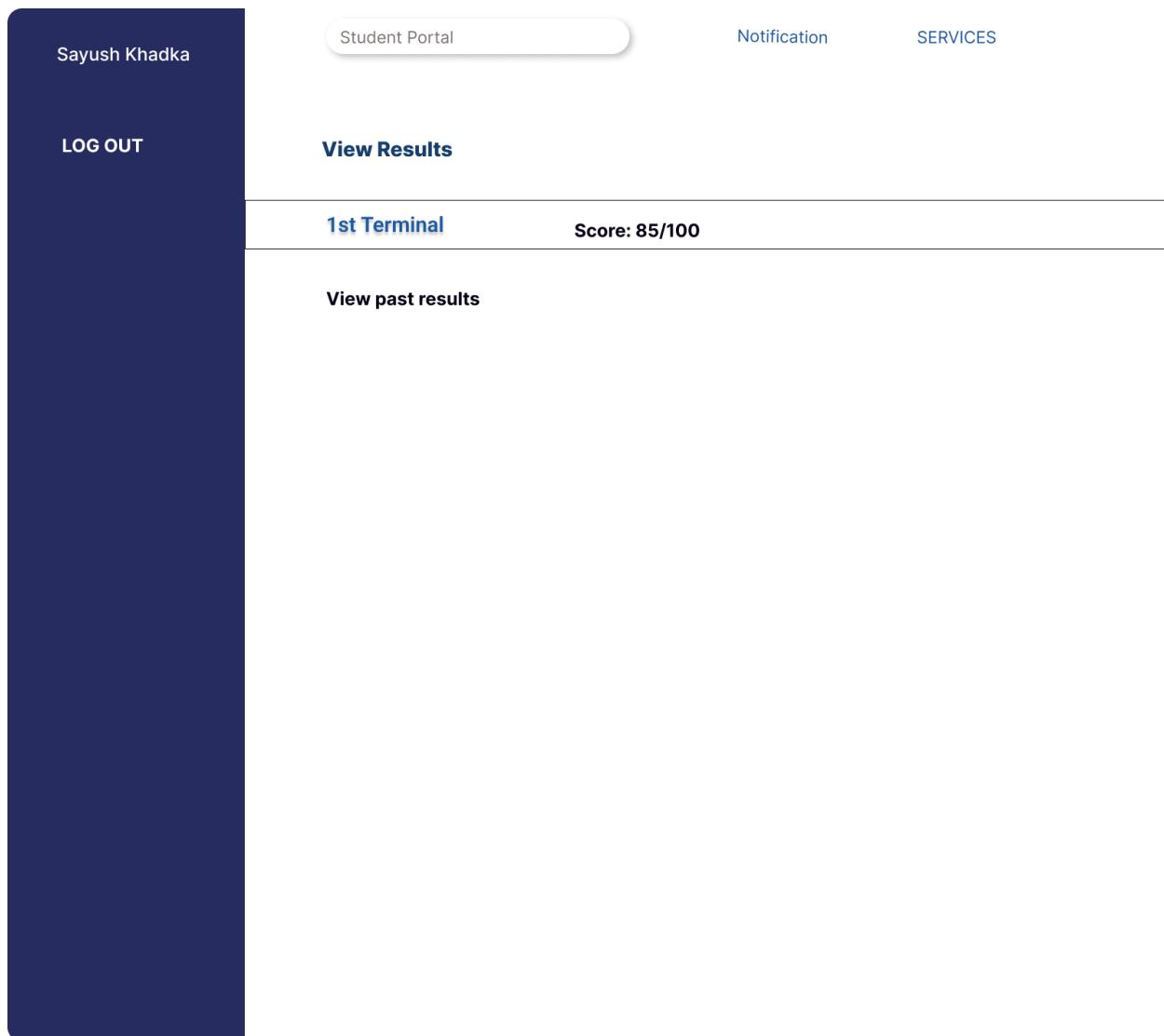


Figure 218 View result (Prototype design)

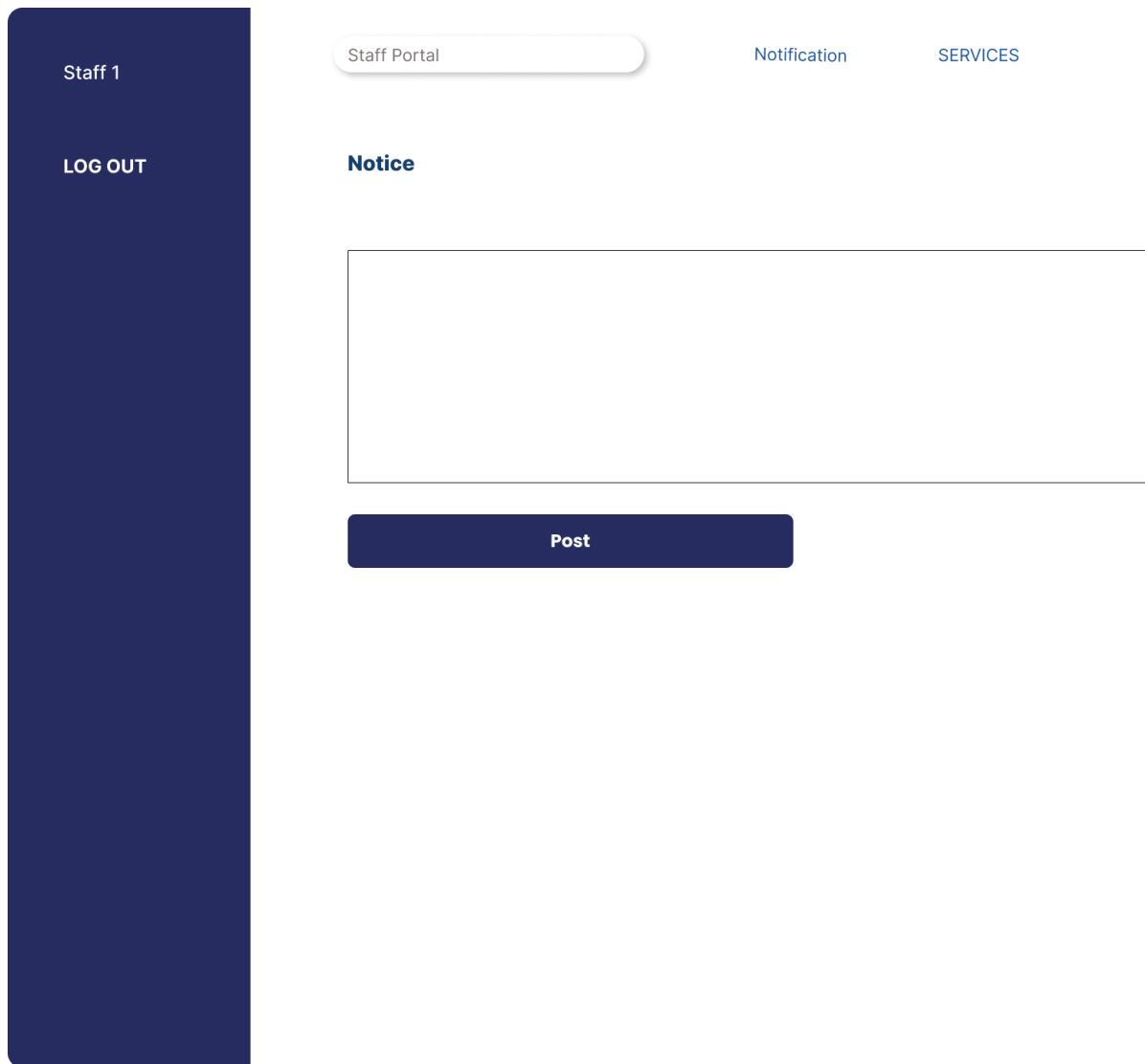


Figure 219 Post notice (Prototype design)

7.5. Screenshots of the system:

7.5.1. Home Page:



Figure 220 Scholar Home Page

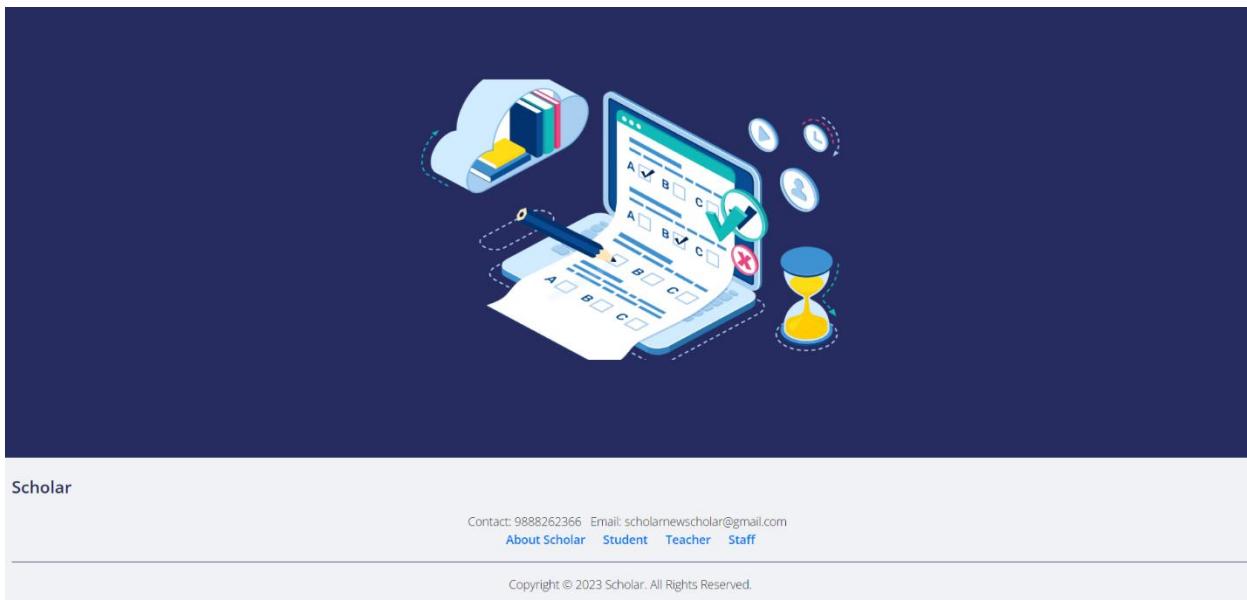


Figure 221 Scholar Home Page (continued)

7.5.2. Staff:

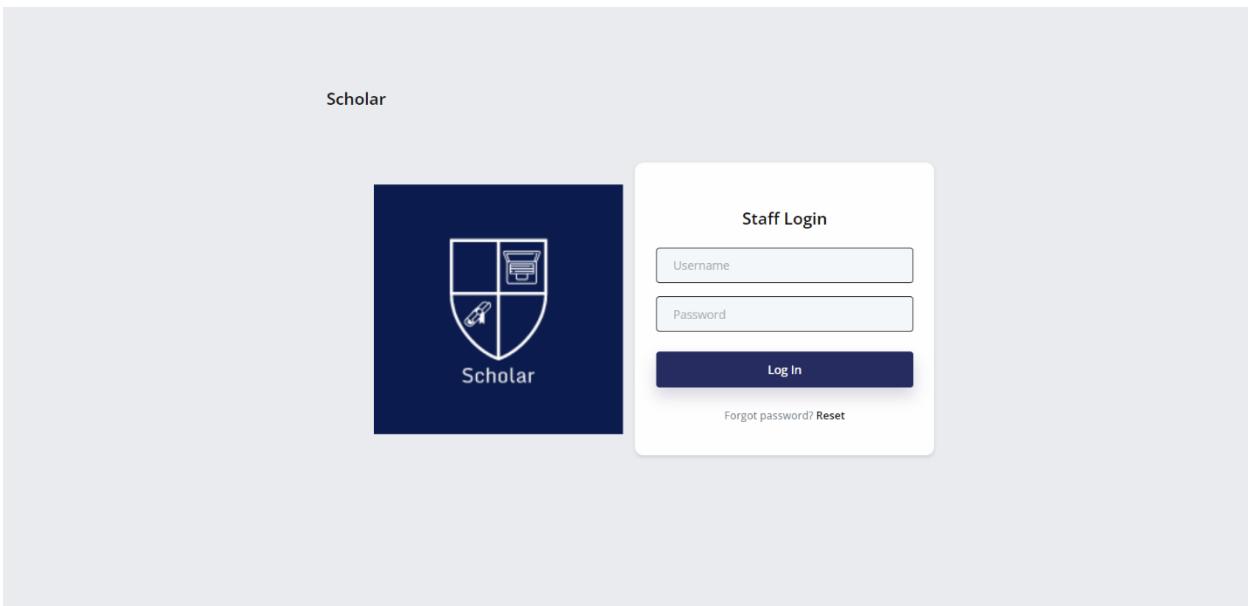


Figure 222 Staff Login

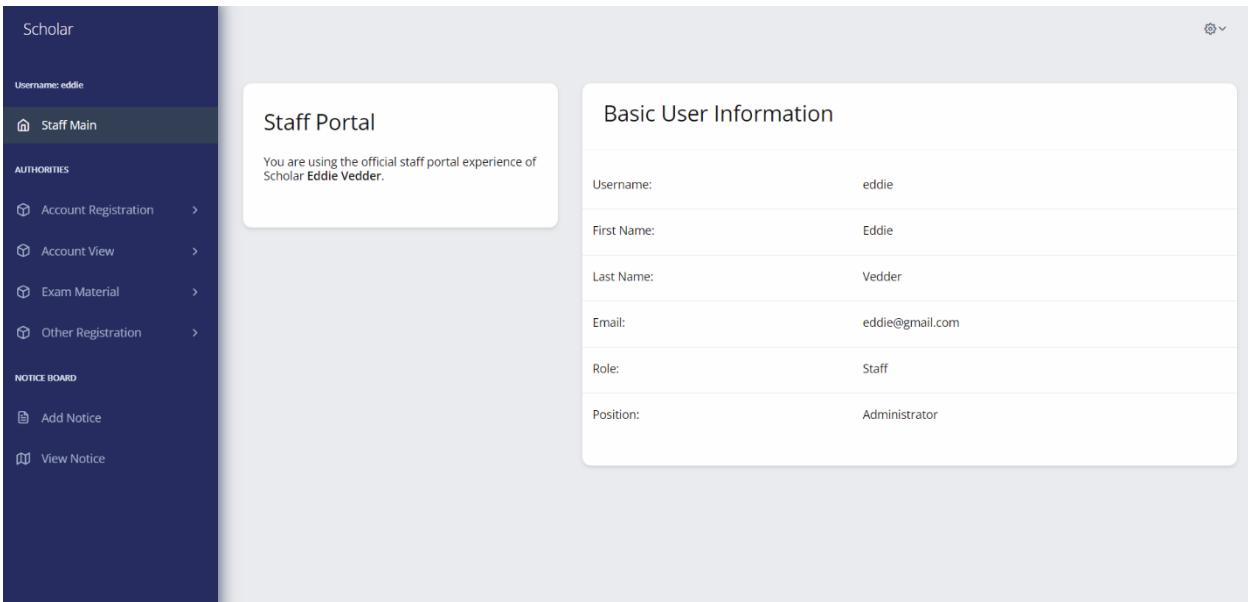


Figure 223 Staff Portal

Scholar

Student Registration

First Name: _____ Role: student

Last Name: _____ Class: _____

Username: _____ Section: _____

Email: _____

Password: _____ Subjects('ctrl' and 'click' to select multiple subjects)

Maths
English
Science

Register

Get back to staff page: [Staff Page](#)

This screenshot shows the 'Student Registration' form within the 'Scholar' application. The form consists of several input fields: 'First Name' and 'Last Name' (both required), 'Username', 'Email', 'Password', and 'Password Confirmation'. To the right of these are dropdown menus for 'Role' (set to 'student'), 'Class', and 'Section'. Below these is a subject selection area with a note: 'Subjects('ctrl' and 'click' to select multiple subjects)'. A scrollable list contains 'Maths', 'English', and 'Science'. At the bottom is a large blue 'Register' button.

Figure 224 Student Registration

Scholar

Teacher Registration

First Name: _____ Role: teacher

Last Name: _____ Class: _____

Email: _____

Username: _____ Section: _____

Password: _____ Subjects('ctrl' and 'click' to select multiple subjects)

Maths
English
Science

Register

Get back to staff page: [Staff Page](#)

This screenshot shows the 'Teacher Registration' form within the 'Scholar' application. The layout is identical to the 'Student Registration' form, with fields for First Name, Last Name, Username, Email, Password, Password Confirmation, Role (set to 'teacher'), Class, Section, and a subject selection dropdown. The subjects listed are Maths, English, and Science. A 'Register' button is at the bottom.

Figure 225 Teacher Registration

Scholar

Staff Registration

First Name	Password
Last Name	Password Confirmation
Username	Role
Email	staff
	Position

Register

Get back to staff page: [Staff Page](#)

This screenshot shows the 'Staff Registration' form within the 'Scholar' application. It includes fields for First Name, Last Name, Username, Email, Password, Password Confirmation, Role (set to 'staff'), and Position. A 'Register' button is at the bottom, and a link to the 'Staff Page' is at the bottom right.

Figure 226 Staff Registration

Student List
@ Staff Portal

Student List

S.no	Username	First Name	Last Name	Email	Delete
1	sayush	Sayush	Khadka	sayushkhadka777@gmail.com	Delete
2	messi	Lionel	Messi	messi@gmail.com	Delete
3	jimmy	Jimmy	Page	jimmy@gmail.com	Delete

This screenshot shows the 'Student List' table on the 'Staff Portal'. It displays three rows of student information: Sayush Khadka (Username: sayush), Lionel Messi (Username: messi), and Jimmy Page (Username: jimmy). Each row includes a 'Delete' button.

Figure 227 Student List

S.no	Username	First Name	Last Name	Email	Delete
1	john	John	Mayer	john@gmail.com	Delete

Figure 228 Teacher List

S.no	Username	First Name	Last Name	Email	Position	Delete
1	eddie	Eddie	Vedder	eddie@gmail.com	Administrator	Delete

Figure 229 Staff List

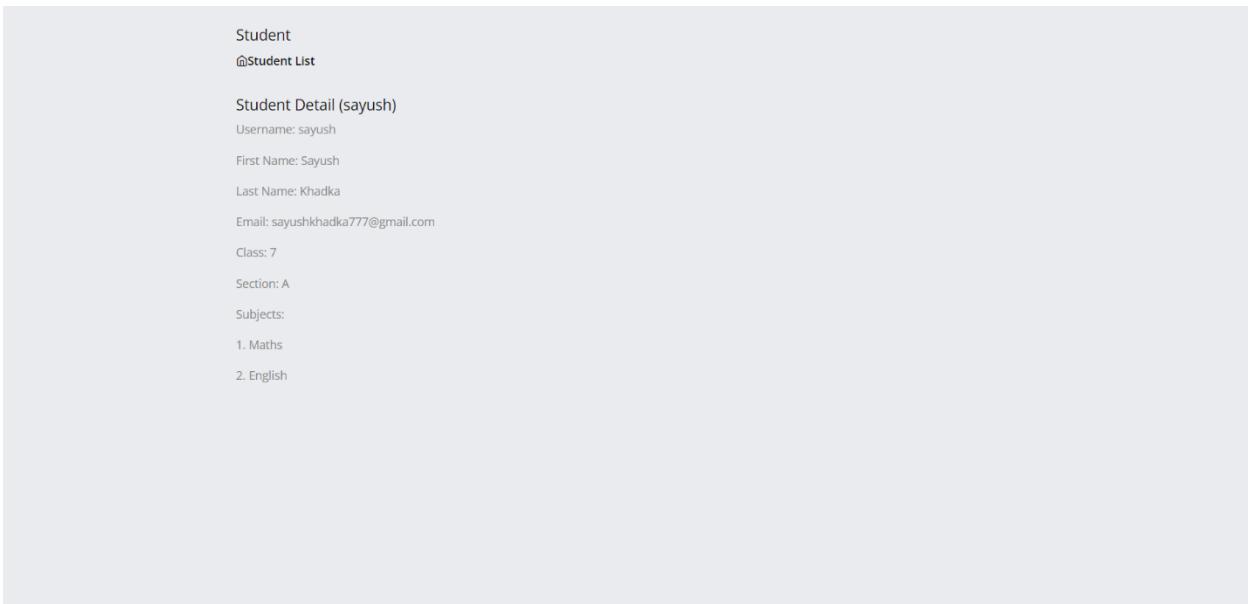


Figure 230 Student Detail

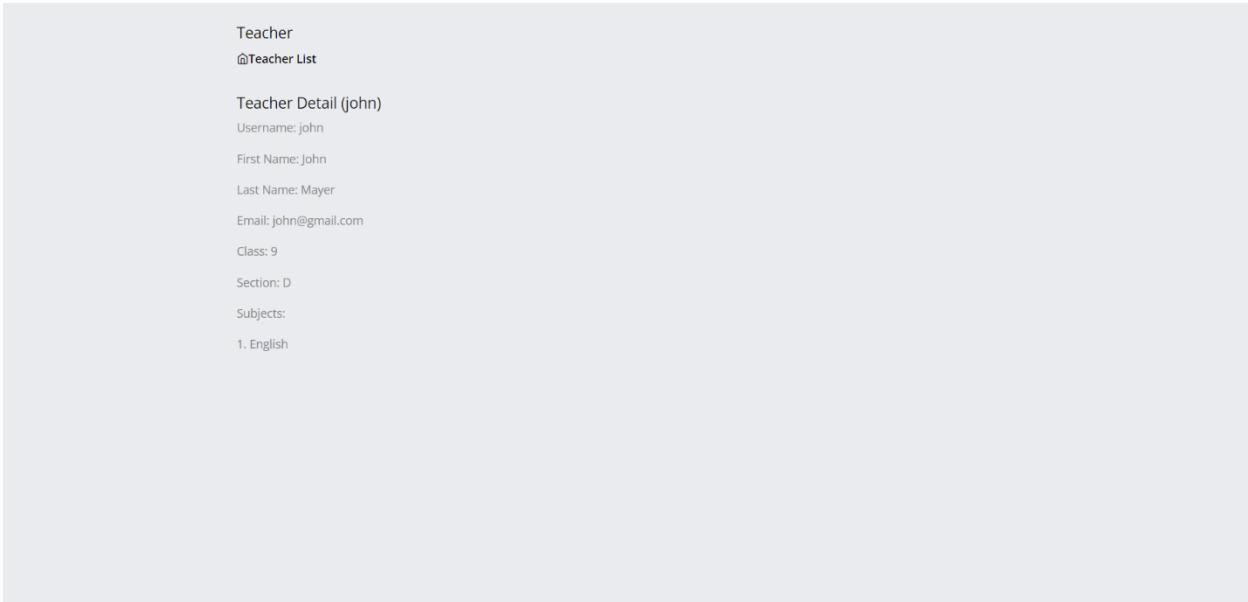


Figure 231 Teacher Detail

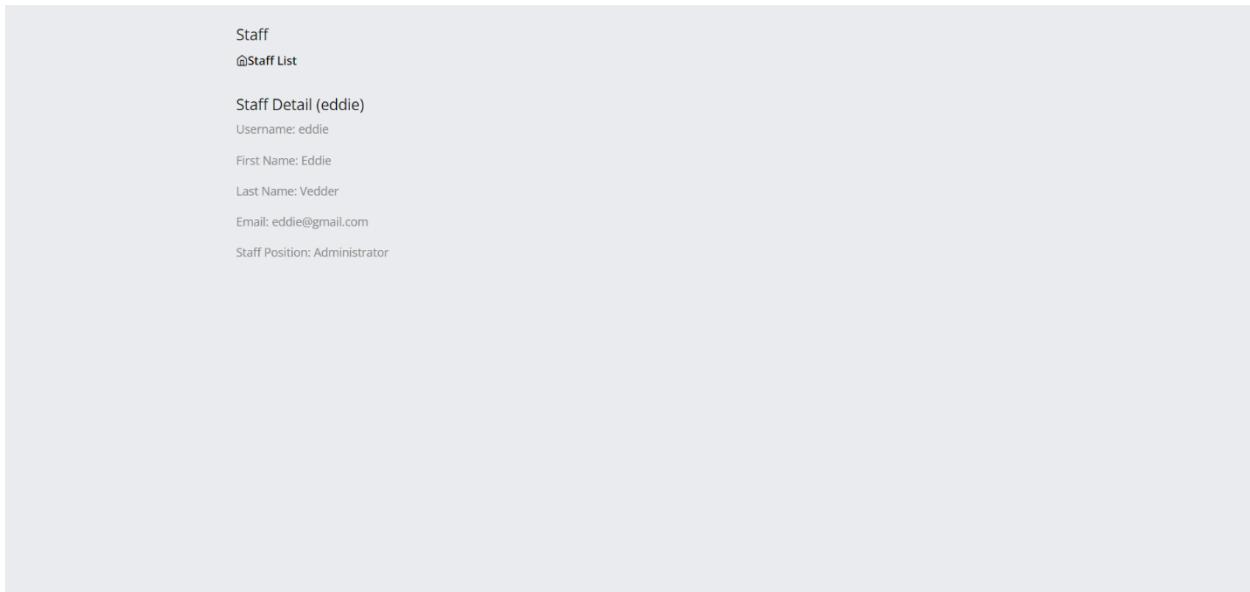


Figure 232 Staff Detail

The screenshot shows a web application interface for a student list. At the top left, there is a 'Student List' button and a 'Staff Portal' link. A modal dialog box is centered on the screen, displaying the message '127.0.0.1:8000 says' and 'Are you sure you want to delete sayush?'. It has two buttons: 'OK' and 'Cancel'. Below the dialog, the main content area is titled 'Student List' and contains a table with the following data:

S.no	Username	First Name	Last Name	Email	Delete
1	sayush	Sayush	Khadka	sayushkhadka777@gmail.com	Delete
2	messi	Lionel	Messi	messi@gmail.com	Delete
3	jimmy	Jimmy	Page	jimmy@gmail.com	Delete

Figure 233 Delete student

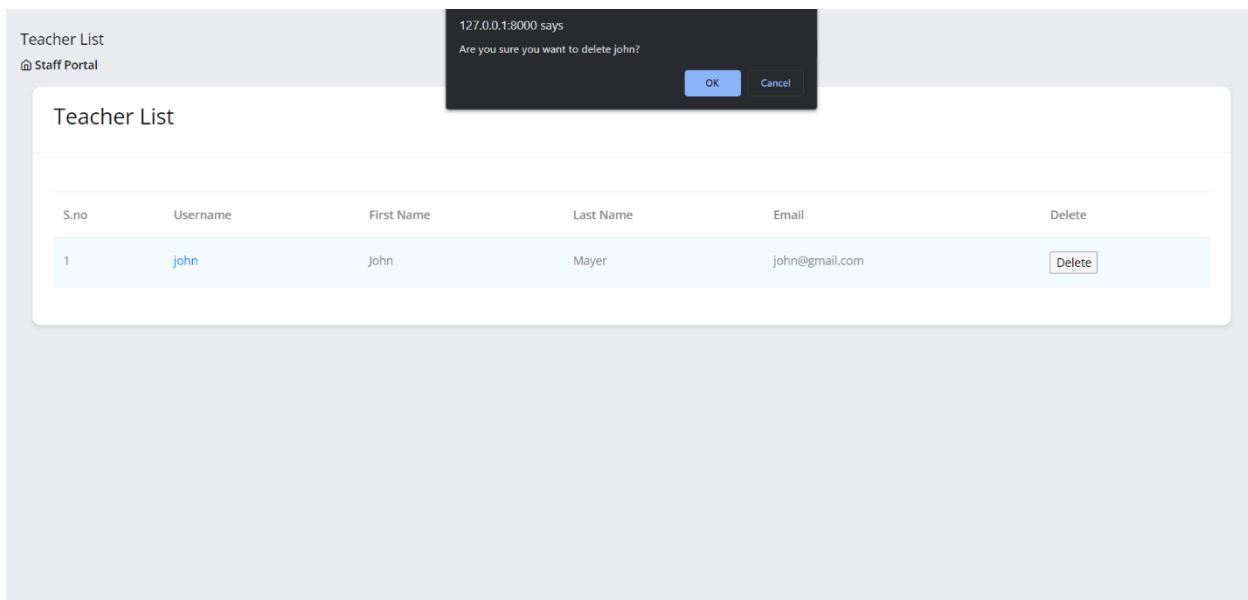


Figure 234 Delete Teacher

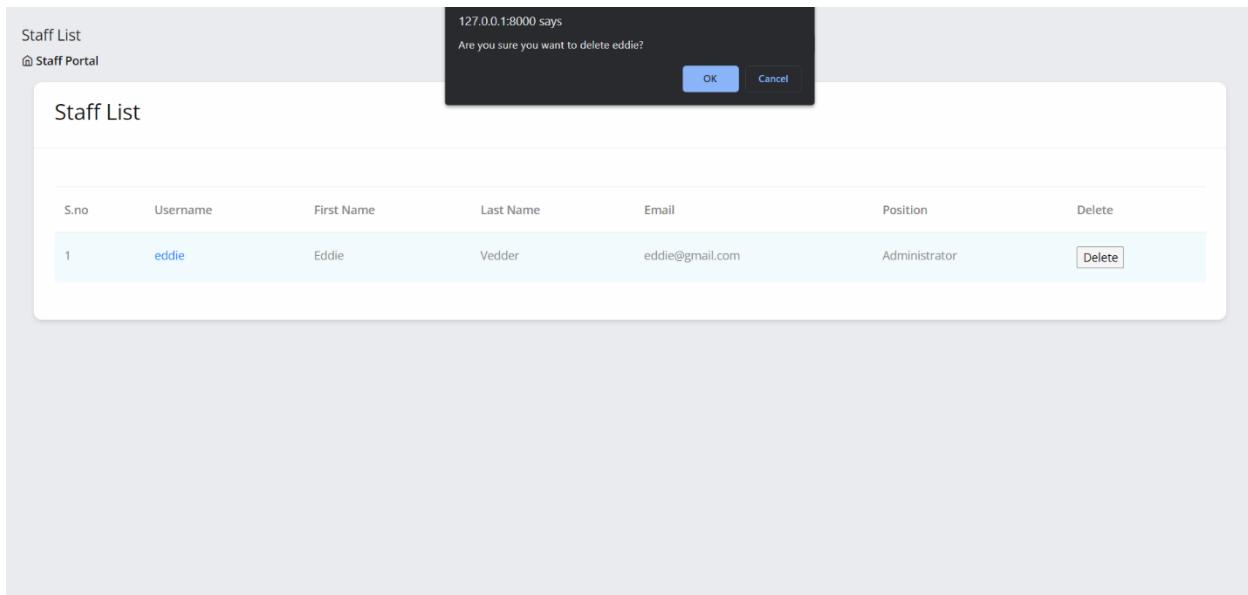


Figure 235 Delete Staff

Scholar

Exam Registration

Class

Section

Subject

Exam name

No of Questions(*Cannot be changed later*)

Time (In minutes)

Score To Pass

Register

Get back to staff page: [Staff Page](#)

Figure 236 Exam Registration

Exam List
[Staff Portal](#)

Exam List

S.no	Title	Subject	Level	Level Section	Update	Delete
1	Maths: First Term	Maths	9	D	Update Maths: First Term	Delete
2	English: Class Test 1	English	7	A	Update English: Class Test 1	Delete
3	Maths: Class Test 2	Maths	7	B	Update Maths: Class Test 2	Delete
4	Science: First Term (Science)	Science	5	C	Update Science: First Term (Science)	Delete
5	Science: Second Term (Science)	Science	9	D	Update Science: Second Term (Science)	Delete

Figure 237 Exam List

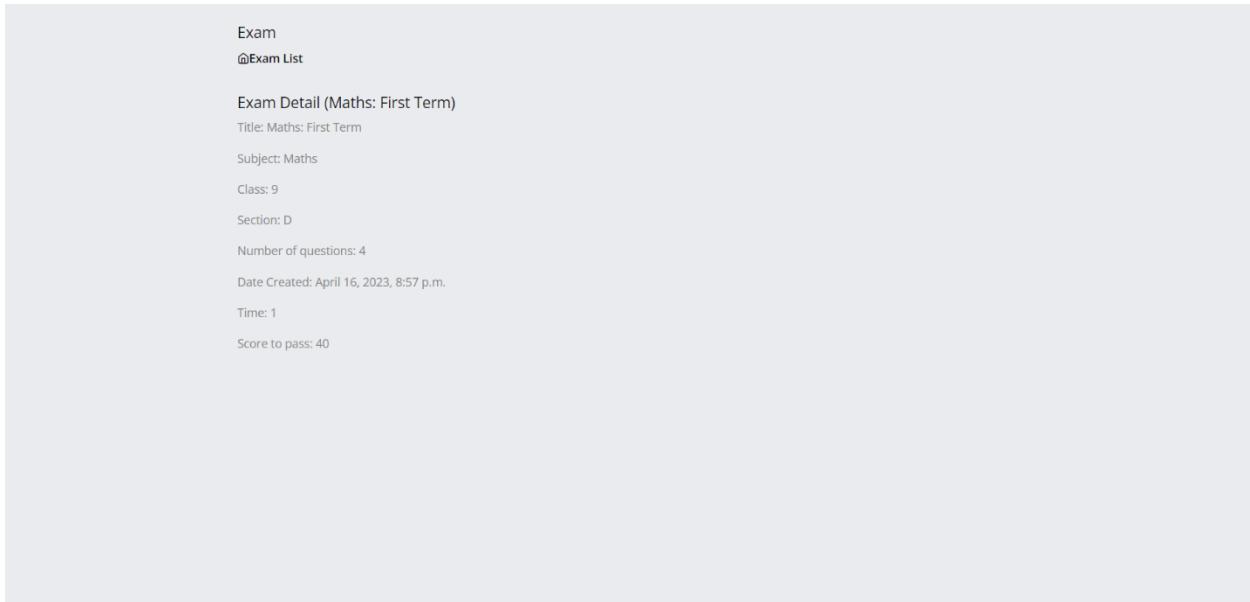


Figure 238 Exam Detail

The screenshot shows a web page titled "Scholar" with a sub-section titled "Exam Update". The form contains the following fields:

Class	9
Section	-----
Subject	Maths
Term	First Term
Questions	4
Time	1
Score	40

Update

Get back to exam list: [Exam List](#)
Get back to staff page: [Staff Page](#)

Figure 239 Exam Update

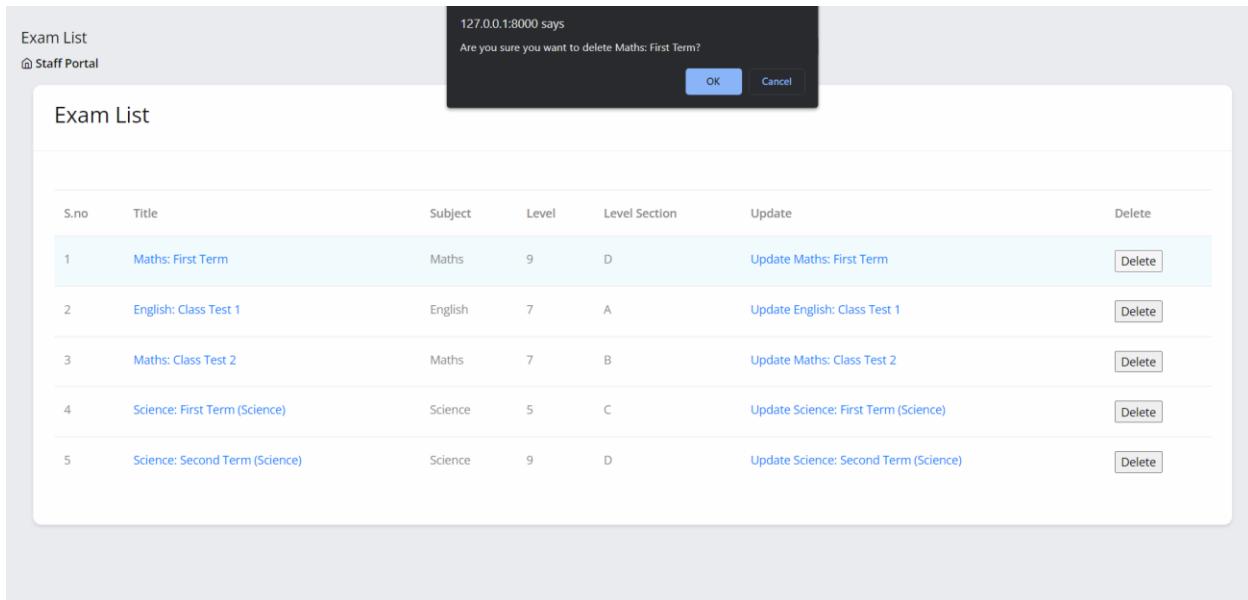


Figure 240 Delete Exam

The screenshot shows a web application interface for viewing exam results. At the top left, it says "Exam Result" and "Staff Portal". A table titled "Exam Result List" is displayed with the following columns: S.no, Username, First Name, Last Name, Email, Class, Section, Subject, Exam Title, Exam Name, Number of Questions, Date Created, and TIR. The table contains eight rows of exam result data:

S.no	Username	First Name	Last Name	Email	Class	Section	Subject	Exam Title	Exam Name	Number of Questions	Date Created	TIR
1	sayush	Sayush	Khadka	sayushkhadka777@gmail.com	9	D	Maths	First Term	Maths: First Term	4	April 16, 2023, 8:57 p.m.	1
2	sayush	Sayush	Khadka	sayushkhadka777@gmail.com	9	D	Maths	First Term	Maths: First Term	4	April 16, 2023, 8:57 p.m.	1
3	sayush	Sayush	Khadka	sayushkhadka777@gmail.com	9	D	Maths	First Term	Maths: First Term	4	April 16, 2023, 8:57 p.m.	1
4	sayush	Sayush	Khadka	sayushkhadka777@gmail.com	9	D	Maths	First Term	Maths: First Term	4	April 16, 2023, 8:57 p.m.	1
5	sayush	Sayush	Khadka	sayushkhadka777@gmail.com	9	D	Maths	First Term	Maths: First Term	4	April 16, 2023, 8:57 p.m.	1
6	sayush	Sayush	Khadka	sayushkhadka777@gmail.com	9	D	Maths	First Term	Maths: First Term	4	April 16, 2023, 8:57 p.m.	1
7	sayush	Sayush	Khadka	sayushkhadka777@gmail.com	9	D	Maths	First Term	Maths: First Term	4	April 16, 2023, 8:57 p.m.	1
8	jimmy	Jimmy	Page	jimmy@gmail.com	9	D	Maths	First Term	Maths: First Term	4	April 16, 2023, 8:57 p.m.	1

Figure 241 Exam result list

The screenshot shows a web-based class registration form. At the top left, the word "Scholar" is displayed in a dark font. Below it, the title "Class Registration" is centered. A rectangular input field contains the placeholder text "Class". Underneath the input field is a note: "Note: Enter positive integer number only.". To the right of the input field is a dark blue rectangular button with the white text "Register". At the bottom right of the form area, there is a small link: "Get back to staff page: [Staff Page](#)". The entire form is set against a light gray background.

Figure 242 Class registration

Scholar

Section Registration

Class

Section

Register

Get back to staff page: [Staff Page](#)

A screenshot of a web-based application interface titled "Scholar". The main section is titled "Section Registration". It contains two input fields: "Class" and "Section", both represented by dropdown menus. Below these is a dark blue button labeled "Register". At the bottom right of the form area, there is a link "Get back to staff page: Staff Page".

Figure 243 Section registration

Scholar

Subject Registration

Subject

Register

Get back to staff page: [Staff Page](#)

A screenshot of a web-based application interface titled "Scholar". The main section is titled "Subject Registration". It contains one input field labeled "Subject", which is a dropdown menu. Below it is a dark blue button labeled "Register". At the bottom right of the form area, there is a link "Get back to staff page: Staff Page".

Figure 244 Subject registration

The screenshot shows a web page titled "Scholar" with a sub-section titled "Staff Position Registration". It features a single input field labeled "Position" and a dark blue "Register" button. Below the form is a link to "Get back to staff page: Staff Page".

Figure 245 Staff Position registration

The screenshot shows a "Staff Notice Board" page. At the top, there's a link to "Staff Main Page" and a message stating "Total number of notice: 1". Below this is a button labeled "Add more". A single notice is displayed, titled "FYP". The notice content includes "FYP submission on Wednesday.", "By: Eddie", and "email- eddie@gmail.com". At the bottom of the notice is a link "Views from other users".

Figure 246 Staff Notice Board

The screenshot shows a web-based application for adding notices. At the top, a header reads "Notices". Below it, a sub-header says "Add Notice". There are four input fields: "Who is posting?", "Email", "Topic", and "Description", each with a red error message below it stating "• This field is required.". A large blue "Post" button is at the bottom. At the very bottom of the page, there is a link "Go back to staff notice board? Staff notice board".

Figure 247 Add Notice

The screenshot shows a user profile on the left with the title "Scholar" and "Username: eddie". Below it are sections for "AUTHORITIES" (Account Registration, Account View, Exam Material, Other Registration) and "NOTICE BOARD" (Add Notice, View Notice). The "View Notice" link is highlighted with a dark bar. On the right, a notice titled "FYP" is displayed. It contains the text "FYP submission on Wednesday.", the author "By: Eddie", the email "Email: eddie@gmail.com", and a section for "Comments from students".

Figure 248 Staff View Notice

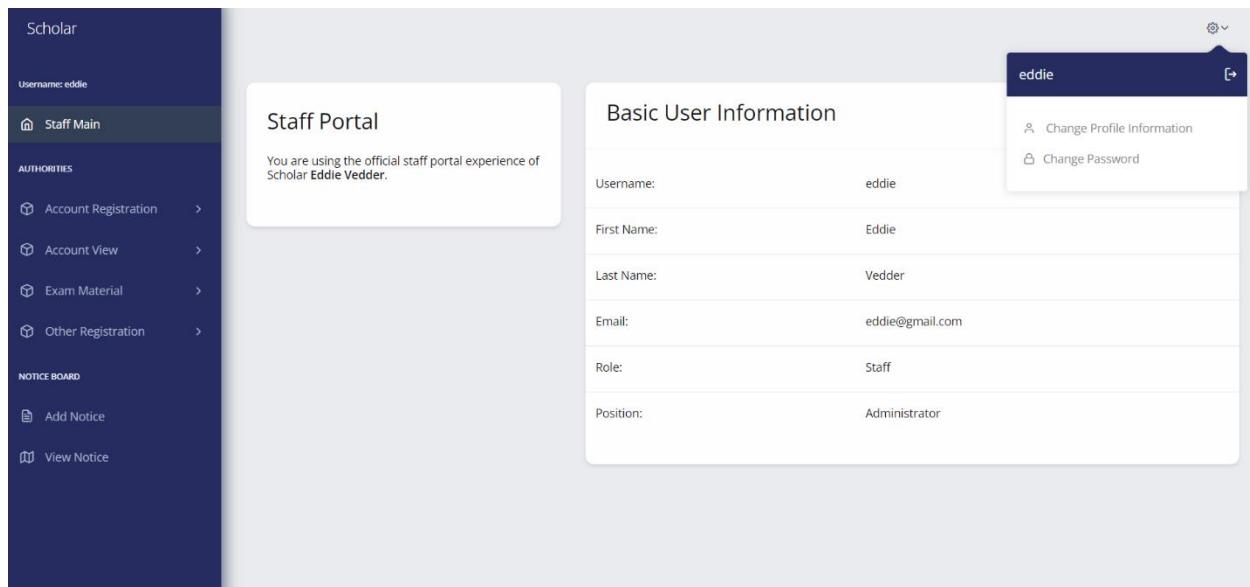


Figure 249 Staff setting

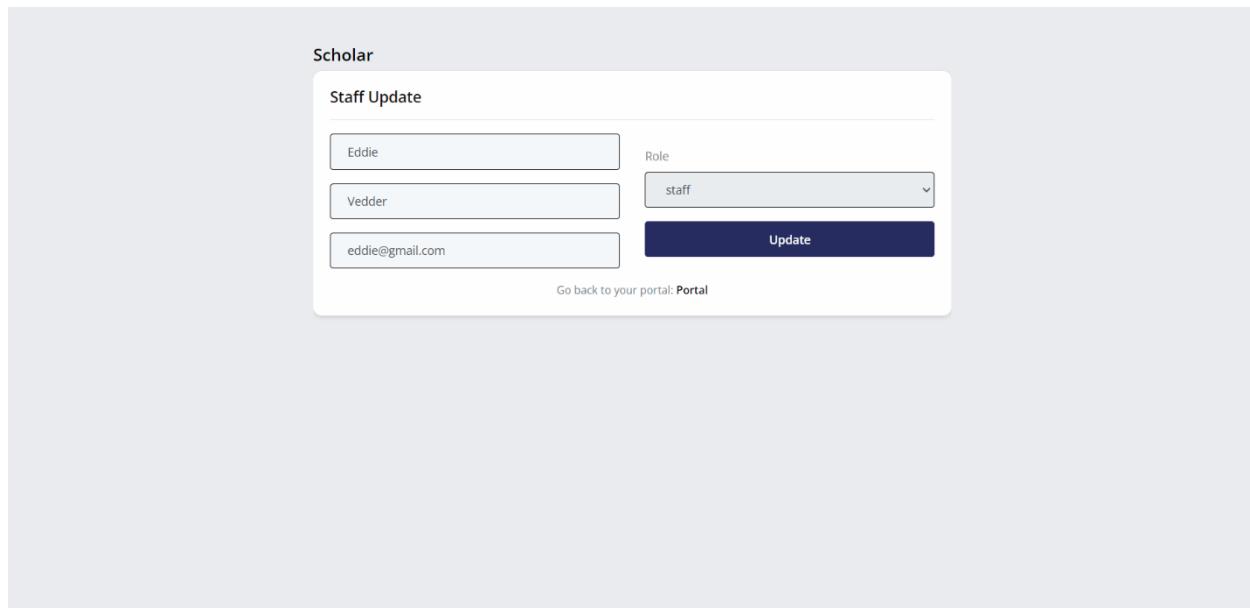


Figure 250 Staff Update

Password Change

Password Change Form

Old password:

New 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.

New password confirmation:

Change Password

The screenshot shows a 'Password Change' form. At the top, it says 'Password Change' and 'Password Change Form'. Below that is a section for 'Old password' with an input field. Next is 'New password' with an input field. Underneath 'New password' is a list of four bullet points: '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.', and 'Your password can't be entirely numeric.' Below 'New password' is another input field for 'New password confirmation'. At the bottom is a dark blue button labeled 'Change Password'.

Figure 251 Staff change password

7.5.3. Teacher:

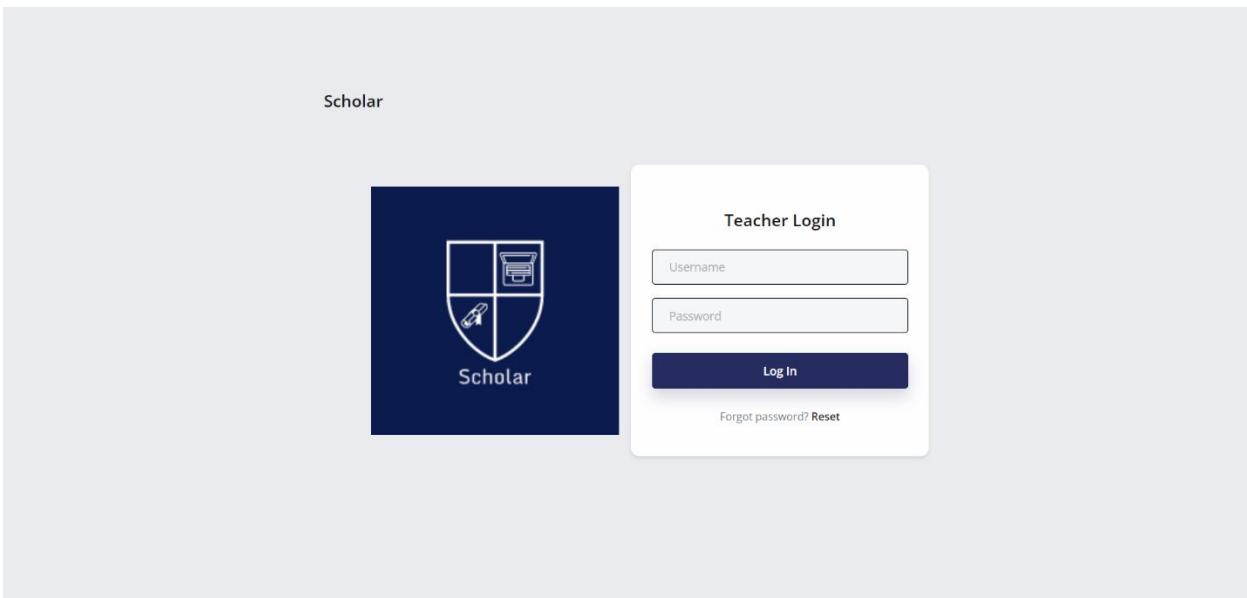


Figure 252 Teacher Login

A screenshot of the "Teacher Portal" dashboard. On the left is a dark sidebar with the "Scholar" logo at the top, followed by "Username: john", a "Teacher Main" button, "AUTHORITIES" section with "Exam Material" and a right-pointing arrow, and a "NOTICE BOARD" section with "Add Notice" and "View Notice". The main content area has a white header "Teacher Portal" and a sub-header "You are using the official teacher portal experience of Scholar John Mayer.". To the right is a large white box titled "Basic User Information" containing a table of user details:

Username:	john
First Name:	John
Last Name:	Mayer
Email:	john@gmail.com
Role:	Teacher
Class:	9
Section:	D
Subjects:	1. English

Figure 253 Teacher Portal

Scholar

Exam Registration

Class

Section

Subject

Exam name

No of Questions(*Cannot be changed later*)

Time (In minutes)

Score To Pass

Register

Now create questions? [Create Questions](#)
Get back to staff page: [Teacher Page](#)

This screenshot shows the 'Exam Registration' form on the Scholar platform. It includes fields for Class, Section, Subject, Exam name, No of Questions, Time (In minutes), and Score To Pass. A prominent 'Register' button is at the bottom. Below the form, there are links to 'Create Questions' and 'Teacher Page'.

Figure 254 Exam registration

Add Questions

[Return to your portal](#)

Exam name *

Type of Question *

Title *

Question

Add Answers

Answer *	Correct(Write true for the correct answer) *	Delete?
	False	<input type="checkbox"/>

This screenshot shows two forms: 'Add Questions' and 'Add Answers'. The 'Add Questions' form has fields for Exam name, Type of Question (set to 'Multiple Choice'), and Title. The 'Add Answers' form has fields for Answer, Correct (set to 'False'), and Delete? (with a checkbox).

Figure 255 Create question

Type of Question * **Multiple Choice**

Title *

Question

Add Answers

Answer *	Correct(Write true for the correct answer) *	Delete?
<input type="text"/>	False	<input type="checkbox"/>

Add answers

Submit

Figure 256 Create question (continued)

Question List

Question List

S.no	Question	Exam Name	Subject	Update	Delete
1	1+1?	Maths: First Term	Maths	Update 1+1?	Delete
2	2+2?	Maths: First Term	Maths	Update 2+2?	Delete
3	2*2?	Maths: First Term	Maths	Update 2*2?	Delete
4	5-4?	Maths: First Term	Maths	Update 5-4?	Delete

Figure 257 Question List

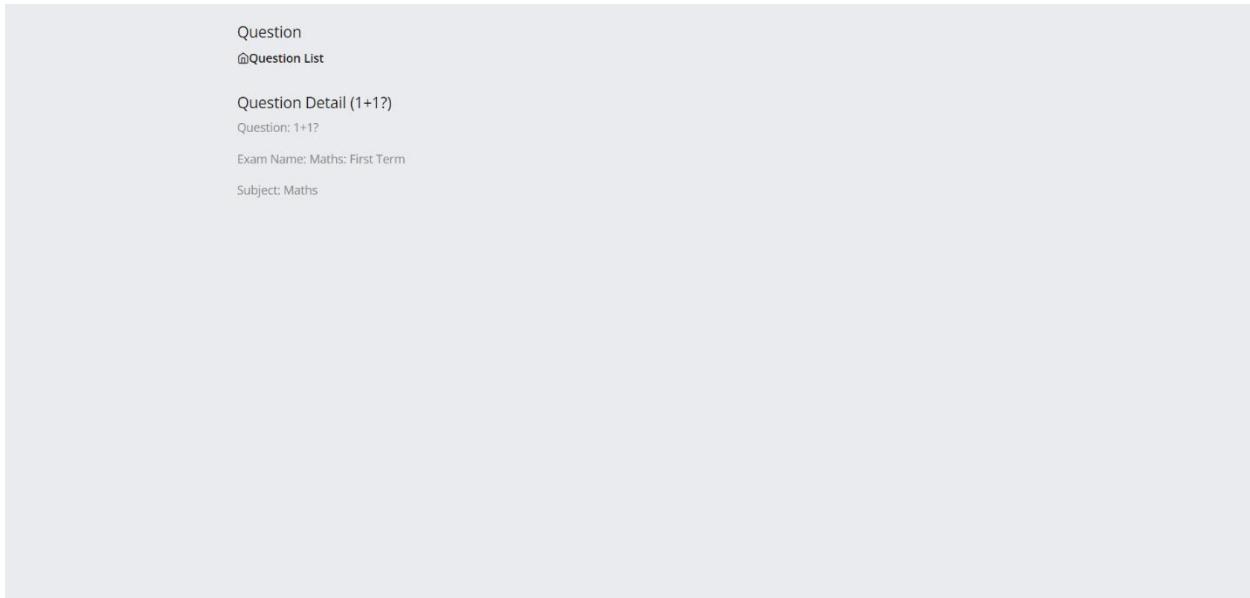


Figure 258 Question Detail

The screenshot shows a 'Question List' page. On the left, there's a 'Question List' button and a 'Teacher Portal' link. The main area displays a table with four rows of question data:

S.no	Question	Exam Name	Subject	Update	Delete
1	1+1?	Maths: First Term	Maths	Update 1+1?	Delete
2	2+2?	Maths: First Term	Maths	Update 2+2?	Delete
3	2*2?	Maths: First Term	Maths	Update 2*2?	Delete
4	5-4?	Maths: First Term	Maths	Update 5-4?	Delete

A modal dialog box is centered over the table, asking 'Are you sure you want to delete 1+1??' with 'OK' and 'Cancel' buttons.

Figure 259 Delete question

Add Questions

@Return to your portal

Exam name *

Maths: First Term

Type of Question * | Multiple Choice ▾

Title *

1+1?

Add Answers

Answer *	Correct(Write true for the correct answer) *	Delete?
2	True	<input type="checkbox"/>
4	False	<input type="checkbox"/>
5	False	<input type="checkbox"/>
3	False	<input type="checkbox"/>
	False	<input type="checkbox"/>

Figure 260 Update question

Add Answers

Answer *	Correct(Write true for the correct answer) *	Delete?
2	True	<input type="checkbox"/>
4	False	<input type="checkbox"/>
5	False	<input type="checkbox"/>
3	False	<input type="checkbox"/>
	False	<input type="checkbox"/>

Add answers

Submit

Figure 261 Update question (continued)

S.no	Username	First Name	Last Name	Email	Class	Section	Subject	Exam Title	Exam Name	Number of Questions	Date Created	Tin
1	sayush	Sayush	Khadka	sayushkhadka777@gmail.com	9	D	Maths	First Term	Maths: First Term	4	April 16, 2023, 8:57 p.m.	1
2	sayush	Sayush	Khadka	sayushkhadka777@gmail.com	9	D	Maths	First Term	Maths: First Term	4	April 16, 2023, 8:57 p.m.	1
3	sayush	Sayush	Khadka	sayushkhadka777@gmail.com	9	D	Maths	First Term	Maths: First Term	4	April 16, 2023, 8:57 p.m.	1
4	sayush	Sayush	Khadka	sayushkhadka777@gmail.com	9	D	Maths	First Term	Maths: First Term	4	April 16, 2023, 8:57 p.m.	1
5	sayush	Sayush	Khadka	sayushkhadka777@gmail.com	9	D	Maths	First Term	Maths: First Term	4	April 16, 2023, 8:57 p.m.	1
6	sayush	Sayush	Khadka	sayushkhadka777@gmail.com	9	D	Maths	First Term	Maths: First Term	4	April 16, 2023, 8:57 p.m.	1
7	sayush	Sayush	Khadka	sayushkhadka777@gmail.com	9	D	Maths	First Term	Maths: First Term	4	April 16, 2023, 8:57 p.m.	1
8	jimmy	Jimmy	Page	jimmy@gmail.com	9	D	Maths	First Term	Maths: First Term	4	April 16, 2023, 8:57 p.m.	1

Figure 262 Exam result list

Teacher Notice Board

Return back to: [Teacher Main Page](#)

Total number of notice: 1

[Add more](#)

FYP

FYP submission on Wednesday.

By: Eddie
email- eddie@gmail.com

Views from other users

Figure 263 Teacher Notice Board

Notices

Add Notice

Who is posting?
• This field is required.

Email
• This field is required.

Topic
• This field is required.

Description

Post

Go back to teacher notice board? [Teacher notice board](#)

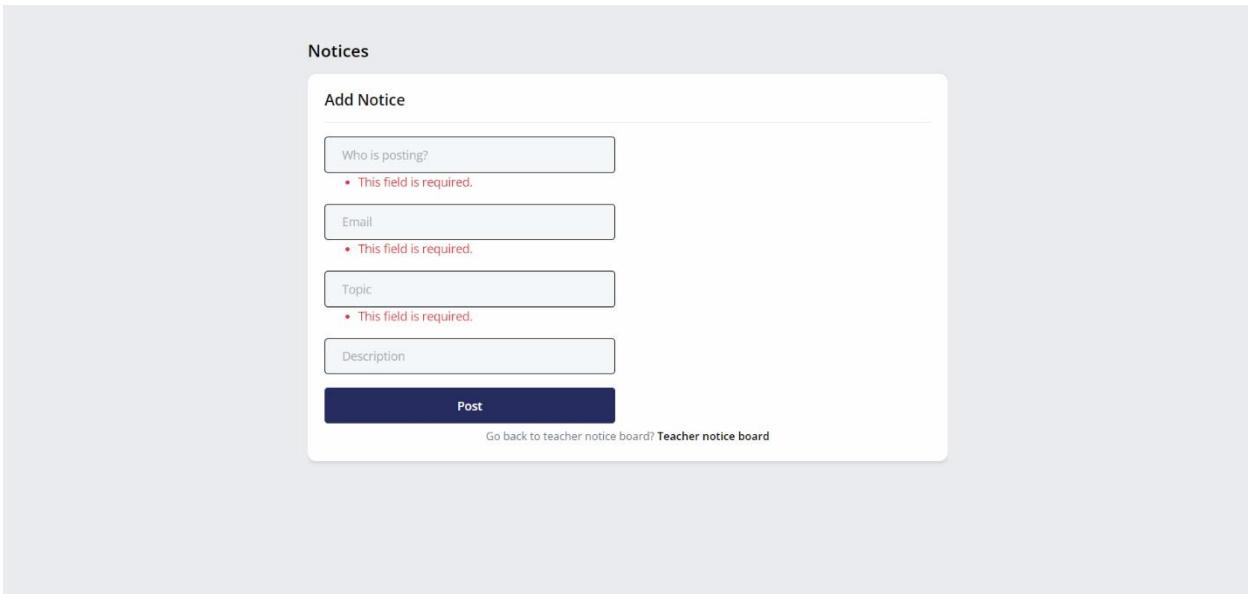


Figure 264 Add notice

Scholar

Username: john

Teacher Main

AUTHORITIES

Exam Material >

NOTICE BOARD

Add Notice

View Notice

FYP

FYP submission on Wednesday.

By: Eddie

Email: eddie@gmail.com

Comments from students

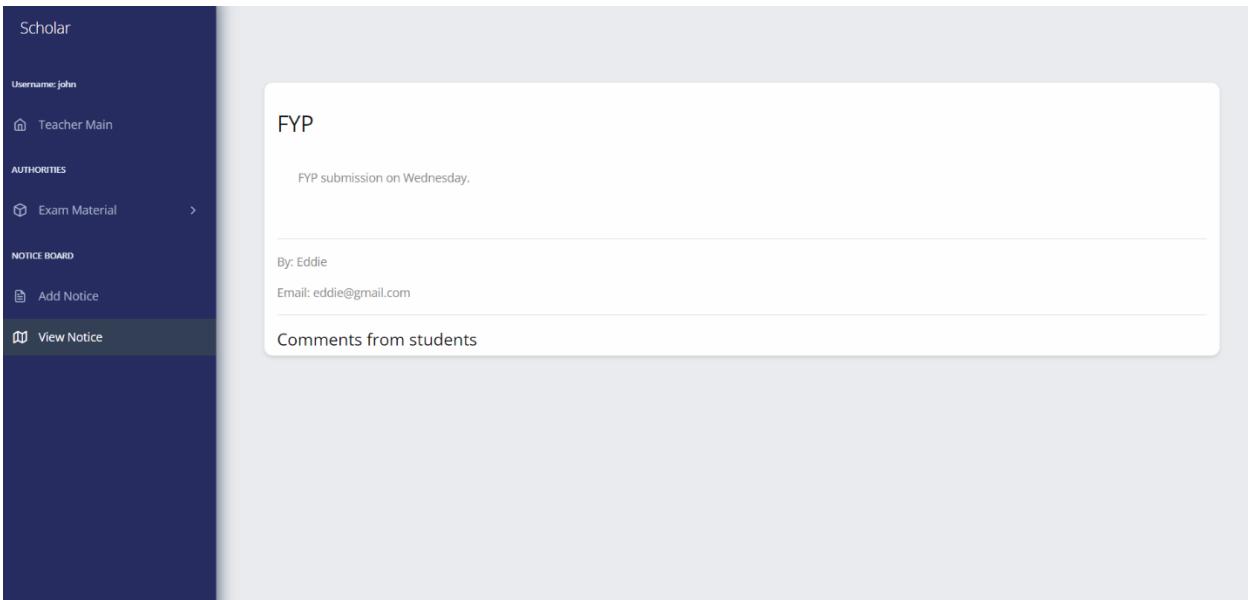


Figure 265 View Notice

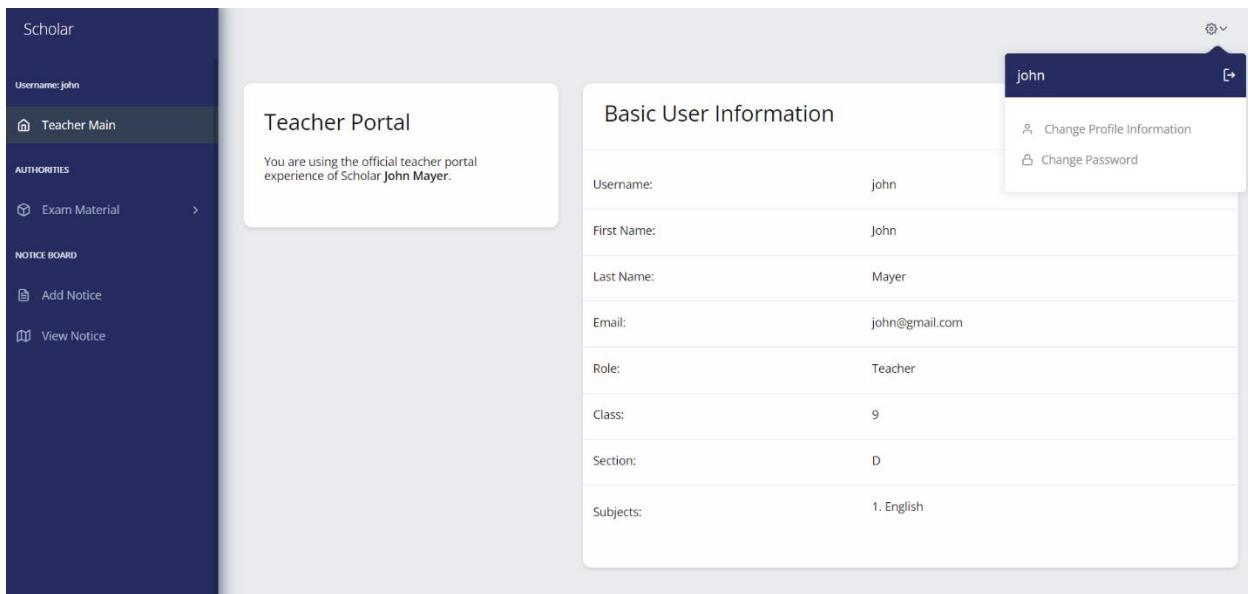


Figure 266 Teacher setting

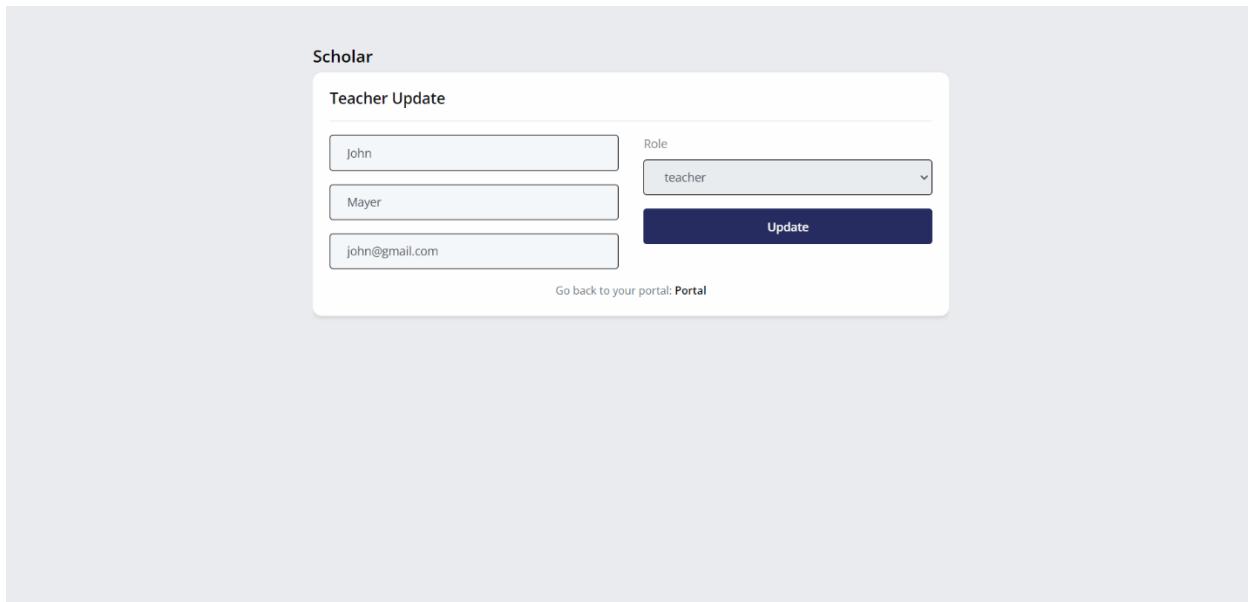


Figure 267 Teacher Update

Password Change

Password Change Form

Old password:

New 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.

New password confirmation:

Change Password

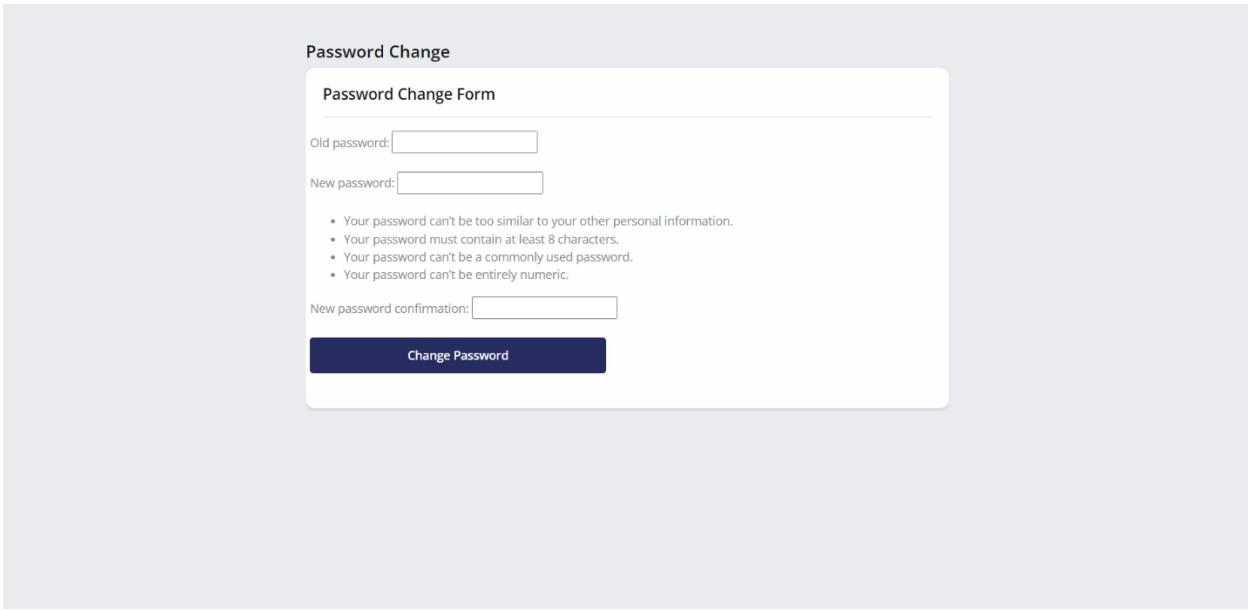


Figure 268 Teacher change password

7.5.4. Student:

The screenshot shows a digital mailbox interface. At the top, there is a search bar labeled "Search mail" and a filter icon. Below the search bar are standard email navigation icons: back, forward, reply, delete, etc. The main content area displays an incoming email. The recipient is listed as "scholarnewscholar@gmail.com" with a blue profile picture. The subject of the email is "Login credentials, student." A small "Inbox" button with an "x" is next to the subject. The body of the email contains a welcome message and login details:
Hello Sayush Khadka. Welcome to Scholar.
Your login credentials are as follows:
Username: sayush
Password: test123456789
Subjects: ['Maths', 'English']

Figure 269 Login credentials provided to student

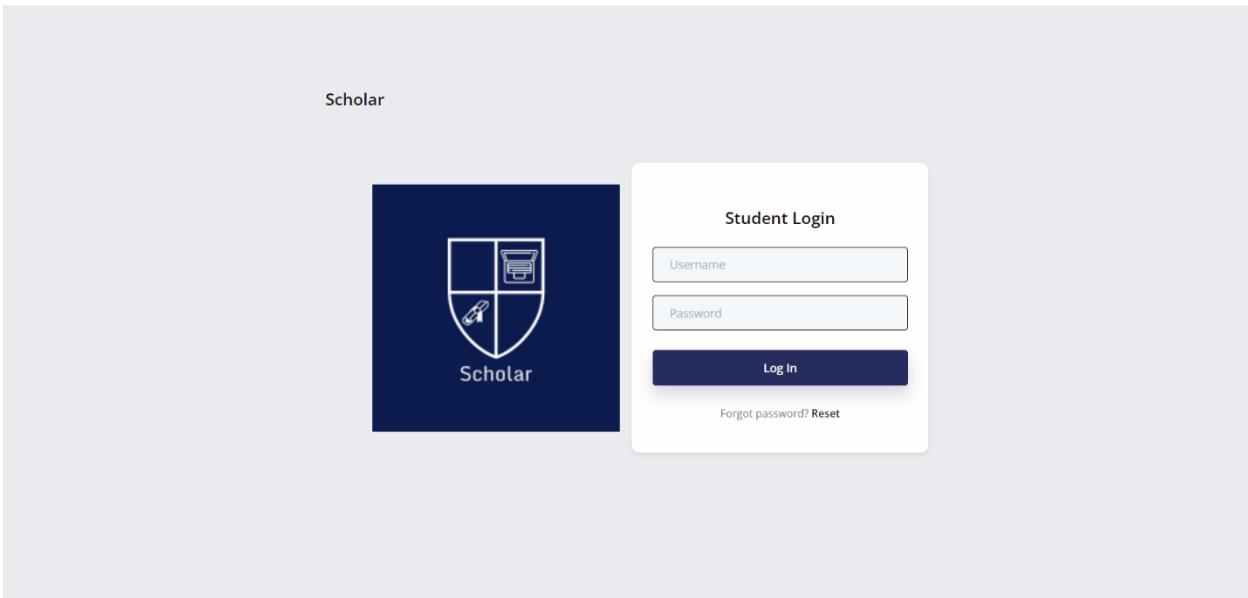


Figure 270 Student Login

A screenshot of the Scholar Student Portal. On the left is a dark blue sidebar with the word "Scholar" at the top. Below it, the text "Username: sayush" is shown, followed by a green bar containing the text "Student Main". Underneath this are sections for "AUTHORITIES" (with a link to "Available exams") and "NOTICE BOARD" (with a link to "View Notice"). The main content area has a light gray background. At the top left of this area is a white box labeled "Student Portal" which contains the text "You are using the official student portal experience of Scholar Sayush Khadka.". To the right of this is a large white box titled "Basic User Information". Inside this box are several data entries separated by horizontal lines:

Username:	sayush
First Name:	Sayush
Last Name:	Khadka
Email:	sayushkhadka777@gmail.com
Role:	Student
Class:	7
Section:	A
Subjects:	1. Maths 2. English

Figure 271 Student Portal

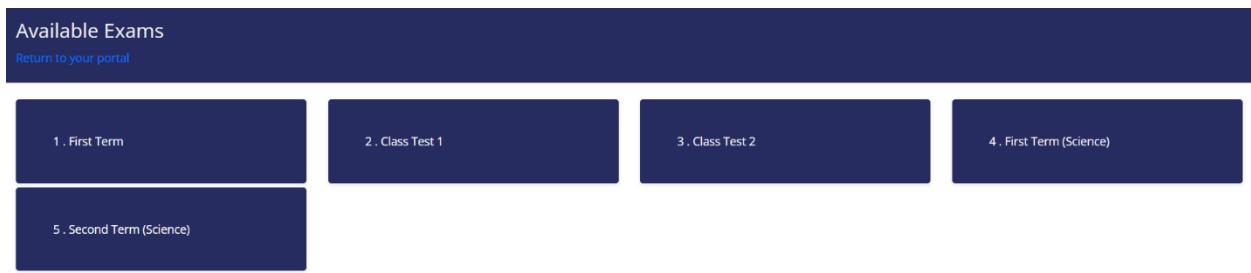


Figure 272 Available Exams

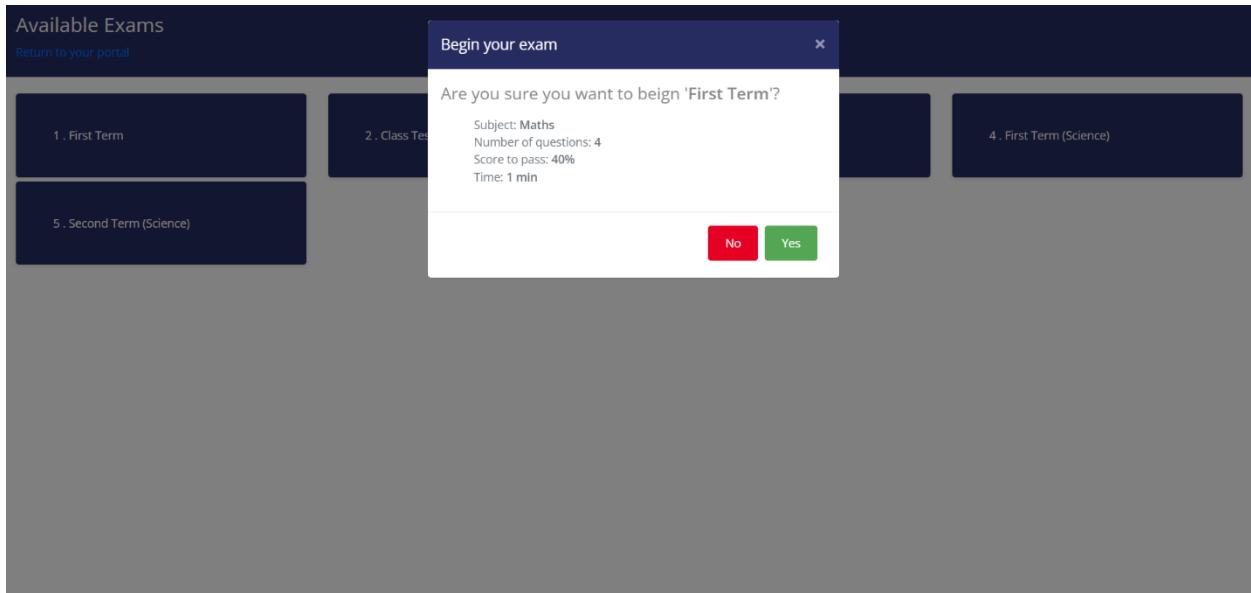


Figure 273 Begin Exam

Subject: Maths
Exam title: First Term
Number of Questions: 4
Score to pass: 40%
[Return to available exams](#)

Timer: 00:54

1+1?

- 2
- 4
- 5
- 3

2+2?

- 5
- 7
- 4
- 3

2*2?

- 5
- 4
- 3
- 1

Figure 274 Attend Exam

2+2?

- 5
- 7
- 4
- 3

2*2?

- 5
- 4
- 3
- 1

5-4?

- 5
- 1
- 2
- 4

[Submit your paper](#)

Figure 275 Attend Exam (continued)



Figure 276 Student View Result

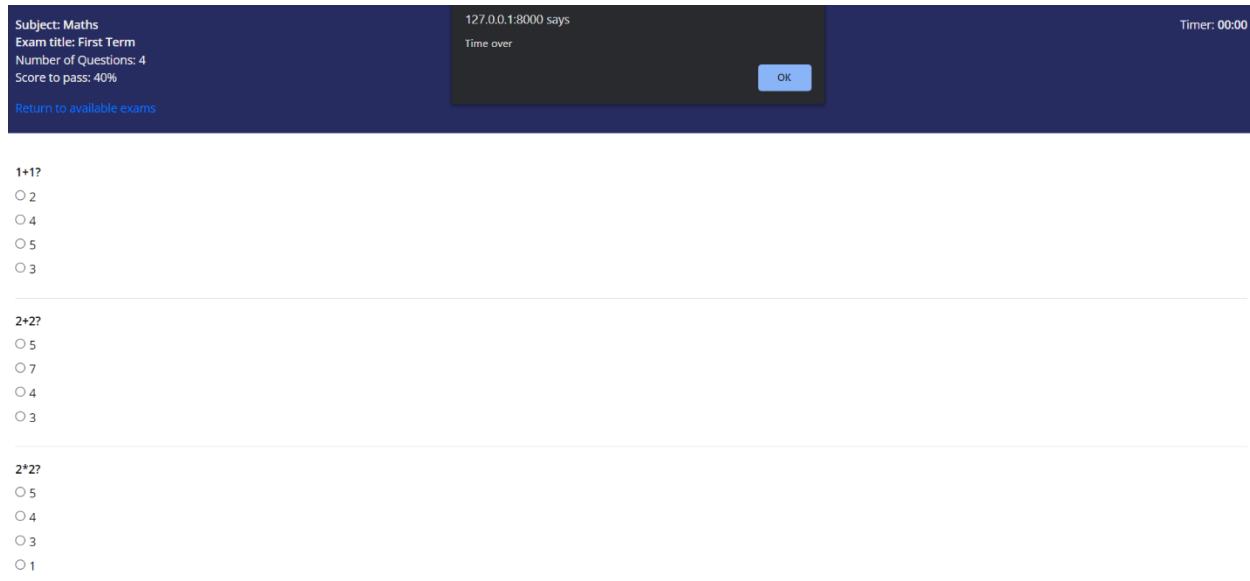


Figure 277 Time over alert message

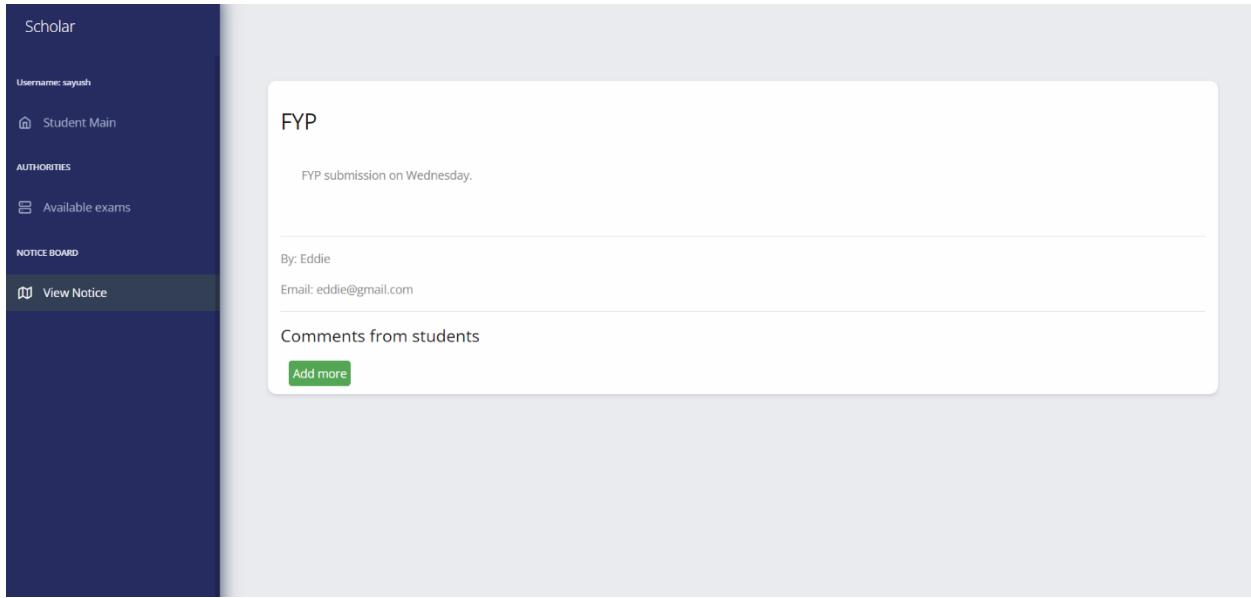


Figure 278 Student Notice View

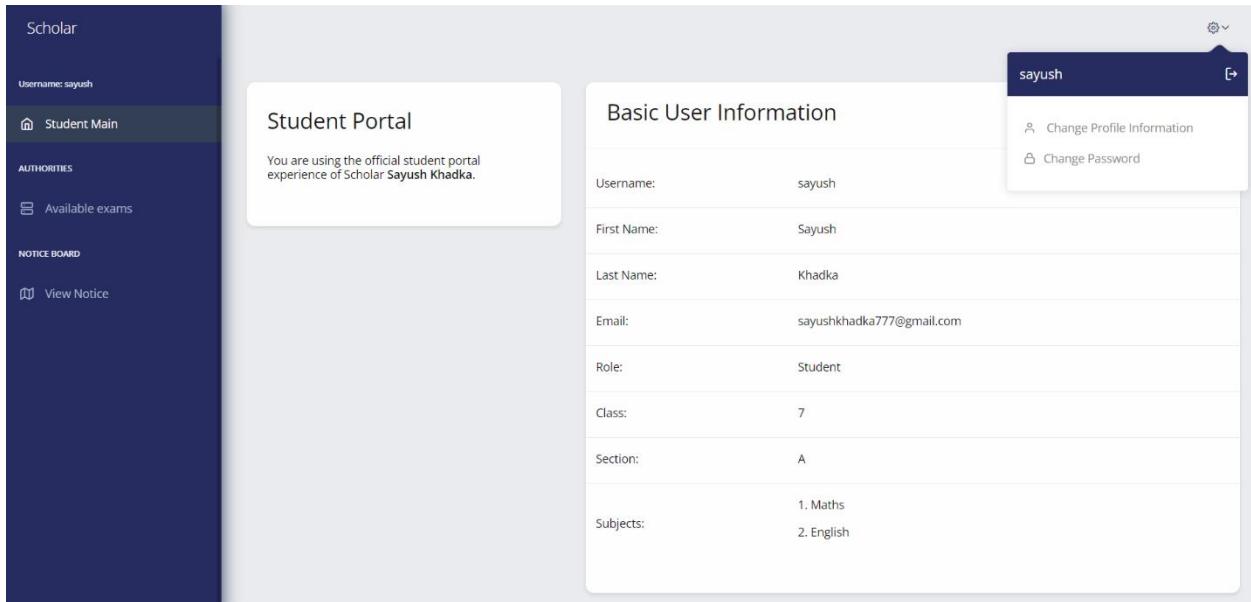


Figure 279 Student settings

The screenshot shows a web application interface titled "Scholar". Below it, a section titled "Student Update" contains three input fields: "Sayush" (Name), "Khadka" (Last Name), and "sayushkhadka777@gmail.com" (Email). To the right of these fields is a dropdown menu labeled "Role" with the option "student" selected. A large blue "Update" button is positioned below the role selection. At the bottom of the update form, there is a link "Go back to your portal: [Portal](#)".

Figure 280 Student Update Information

The screenshot shows a web application interface titled "Password Change". Below it, a section titled "Password Change Form" contains four input fields: "Old password", "New password", "New password confirmation", and a list of password requirements. The password requirements are:

- 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.

A blue "Change Password" button is located at the bottom of the form.

Figure 281 Student Change Password

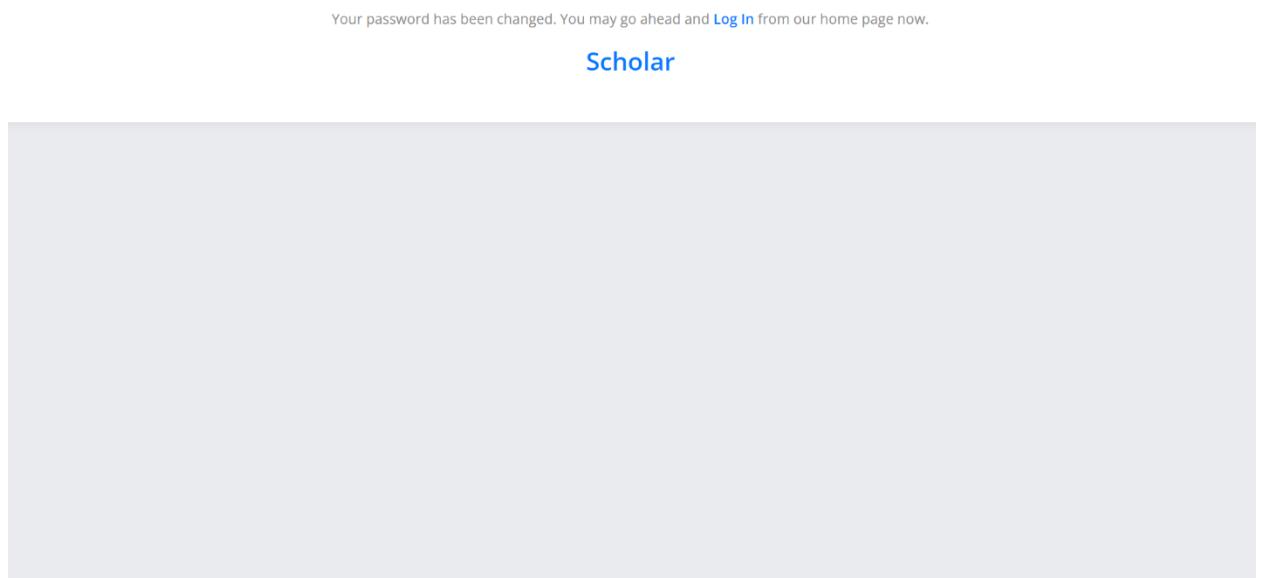


Figure 282 Student Change Password successful confirmation

**Forgot password feature works same for the other actors (staff and teacher) as well*

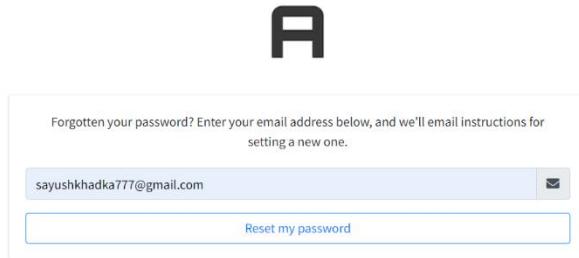


Figure 283 Student forgot password

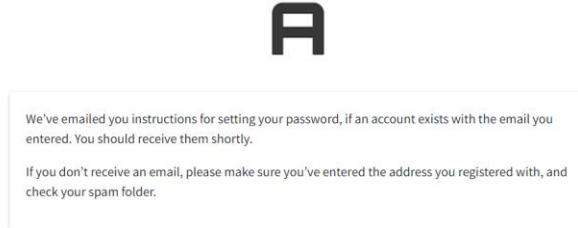


Figure 284 Student forgot password email sent confirmation

A screenshot of an email inbox. The search bar at the top contains the text "Search mail". Below the search bar are various icons for managing emails. The main content area shows an incoming email from "scholarnewscholar@gmail.com" with the subject "Password reset on 127.0.0.1:8000". The email body contains the following text:

You're receiving this email because you requested a password reset for your user account at 127.0.0.1:8000.
Please go to the following page and choose a new password:
<http://127.0.0.1:8000/reset/NA/bmvnwh-0606172a17e5bf854a84890ef4ad6423/>
Your username, in case you've forgotten: sayush
Thanks for using our site!
The 127.0.0.1:8000 team

At the bottom of the email are two buttons: "Reply" and "Forward".

Figure 285 Link sent to student email to reset new password (in case of forgot password)

A

Please enter your new password twice so we can verify you typed it in correctly.

New password

New password confirmation

[Change my password](#)

Figure 286 Student forgot password set new password

Your password has been reset. You may go ahead and [Log In](#) from our home page now.

Scholar

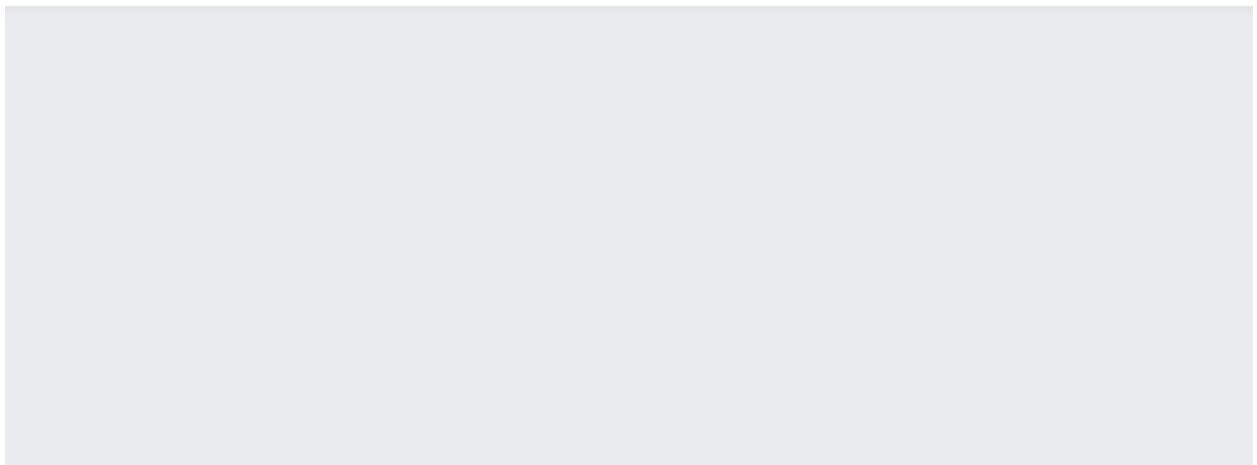


Figure 287 Student forgot password (new password reset successful confirmation)

7.5.5. Database:

The screenshot shows the pgAdmin 4 interface. On the left, the sidebar displays the database structure under 'PostgreSQL 15' and 'exam'. The main area shows a query editor with the following SQL code:

```
1 SELECT * FROM public.account_user
2 ORDER BY id ASC
```

Below the query editor is a table showing the results of the query. The table has columns: id, password, last_login, is_superuser, and username. The data is as follows:

	id	password	last_login	is_superuser	username
1	pbkdf2_sha256\$260000\$S6beTxpstIM2a9tBsbHr\$mymhHZeSSKL1kMz+LpnNEYssXJWtYY247y2t9...		2023-04-18 21:57:17.144476+05:45	true	admin
2	pbkdf2_sha256\$260000\$ZabATTpzTgQlqlqRS01VwU7t7UDlh2406/07mDEVOoyfjb1v+Ljhy2hl...		2023-04-19 01:01:44.842501+05:45	false	eddie
3	pbkdf2_sha256\$260000\$dB7sVhMMacBLaMDyMvYdSScNN2hVzb6EUxNGYy2n1XqNGZ/eSOv1dxXq...		2023-04-19 00:29:25.679463+05:45	false	sayush
4	pbkdf2_sha256\$260000\$kyY201IPGyx6t1T9bTh5sVwwhIn9vMPYJL1OLV4KMVmtpxXLiz27P2p...		2023-04-19 01:19:04.803455+05:45	false	john
5	pbkdf2_sha256\$260000\$4StvXJLJogemflfRUdmF6F05;brUV8Y+0M2wZXRoQgROVgyqYwVe2HL325r...		2023-04-17 05:54:46.556307+05:45	false	messi
6	pbkdf2_sha256\$260000\$ShkC8T2YeEAOTwai07HQiSPtLdx+x/efFATPCXjstu/p3tUM060Tcf0L6tvcZL...		2023-04-17 05:54:24.269349+05:45	false	jimmy

Total rows: 6 of 6 Query complete 00:00:00.237 Ln 1, Col 1

Figure 288 Database

7.5.6. Admin:

The screenshot shows the Django admin dashboard. The left sidebar contains navigation links for 'Dashboard', 'Account', 'Authentication and Authorization', 'Groups', and 'Exam_Material'. The main area is divided into several sections:

- Account** section: Level sections, Levels, Staff positions, Staffs, Students, Subjects, Teachers, Users.
- Exam_Material** section: Answers, Questions, Quizzes, Results.
- Main** section: Discussions, Forums.
- Recent actions** sidebar: FYP (6 hours, 10 minutes ago), Coordinator (2 days ago), C (2 days ago), ! (2 days ago), Administrator (2 days, 3 hours ago), Added "Administrator"., Manager (2 days, 3 hours ago).

Figure 289 Admin dashboard

7.6. Appendix F: User Feedback:

7.6.1. User Feedback Form:

The screenshot shows a user feedback form titled "Scholar". The form consists of three main sections: 1) A text input field labeled "Enter your name" with a placeholder "Short-answer text". 2) A text input field labeled "Please share in your own words what was your experience like on this application?" with a placeholder "Short-answer text". 3) A question "Did you encounter any technical difficulties while using the application? *". Below the question are two radio button options: "Yes" and "No". To the right of the form is a vertical toolbar with icons for adding (+), saving (disk), previewing (Tt), viewing (image), and other document-related functions.

Scholar

User feedback form

Enter your name

Short-answer text

Please share in your own words what was your experience like on this application?

Short-answer text

Did you encounter any technical difficulties while using the application? *

Yes

No

+
Disk
Tt
Image
Video
List

Figure 290 User Feedback Form (fig 1)

Did you feel confident using the application during the examination process? *

1 2 3 4 5

Low High

Would you recommend our online examination system application to others? *

Yes
 No

What feature of the online examination system application did you find most useful or innovative? *

Short-answer text

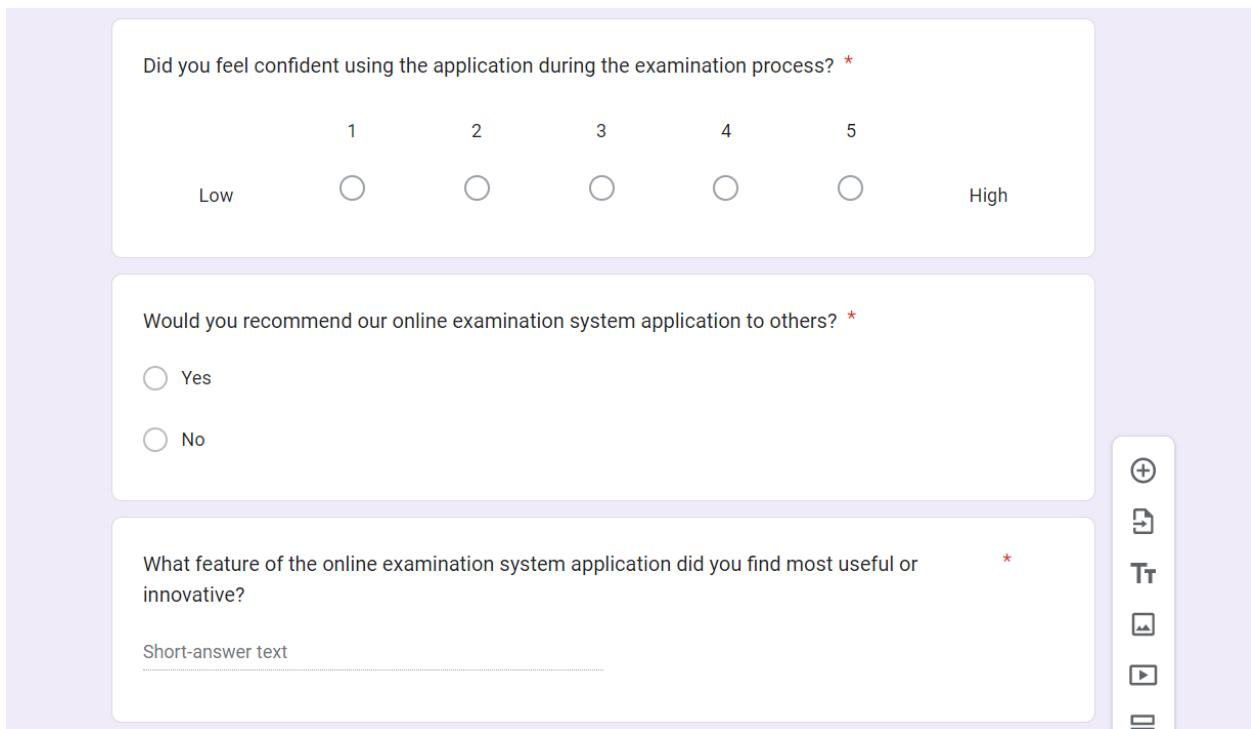


Figure 291 User Feedback Form (fig 2)

What feature of the online examination system application did you find most useful or innovative?

Short-answer text

Please provide your feedback.

Short-answer text

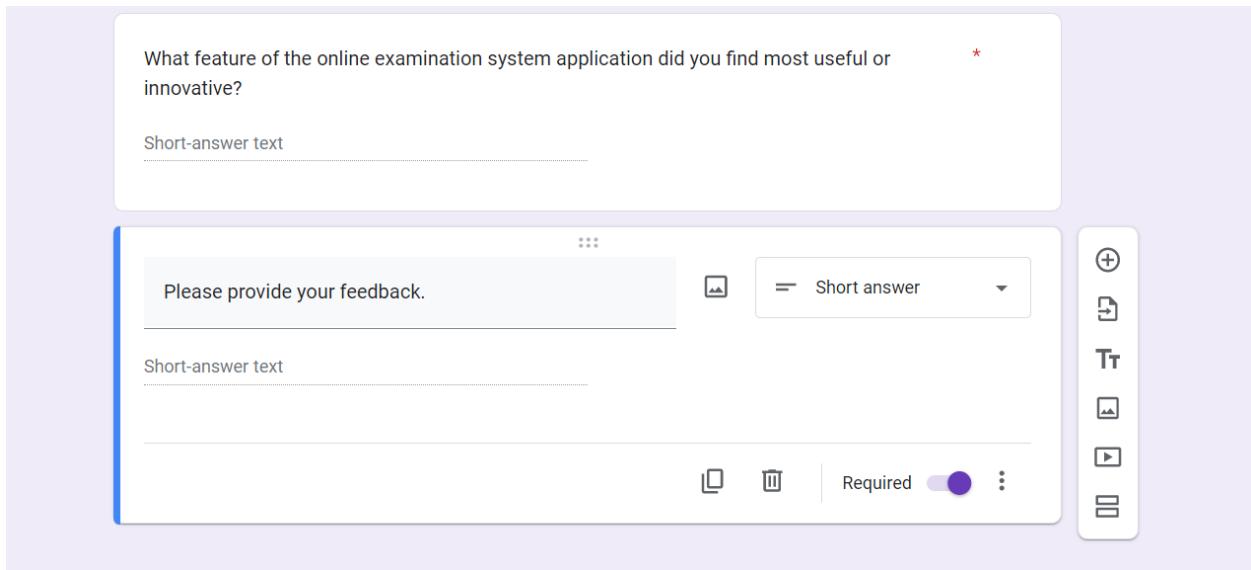


Figure 292 User Feedback Form (fig 3)

7.6.2. Sample of the Filled User Feedback Form:

The screenshot shows a user feedback form titled "Scholar". At the top, it says "User feedback form". Below that, the email address "aasrashakya@gmail.com" is listed with a "Switch account" link and a cloud icon. A note indicates that the form is "Not shared". A red asterisk at the bottom left of the header area indicates that the question below is required.

Enter your name

Aasra Shakya

Please share in your own words what was your experience like on this application?

The application was up to the mark in context to the moto set by it.

Did you encounter any technical difficulties while using the application? *

Yes

No

Figure 293 Sample Filled User Feedback Form (fig 1)

Did you feel confident using the application during the examination process? *

1 2 3 4 5

Low High

Would you recommend our online examination system application to others? *

Yes
 No

What feature of the online examination system application did you find most useful or innovative? *

Instant answer was the most useful and innovative feature of this application with the touch of great interface

Please provide your feedback. *

The overall application was great to use with easy navigation feature. The application looked

Figure 294 Sample Filled User Feedback Form (fig 2)

What feature of the online examination system application did you find most useful or innovative? *

Instant answer was the most useful and innovative feature of this application with the touch of great interface

Please provide your feedback. *

The overall application was great to use with easy navigation feature. The application looked tidy, easy and convenient to use. The application would be even better if it had a feature to share learning materials.

Submit **Clear form**

Never submit passwords through Google Forms.

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

Google Forms

Figure 295 Sample Filled User Feedback Form (fig 3)

7.7. Appendix G: Future Work:

7.7.1. Future Work (Elaborated):

7.7.1.1. Unification with Learning Management System:

It can be easier for instructors to administer exams when the online test system is integrated with a learning management system (LMS), and it can also improve the overall exam-taking experience for students.

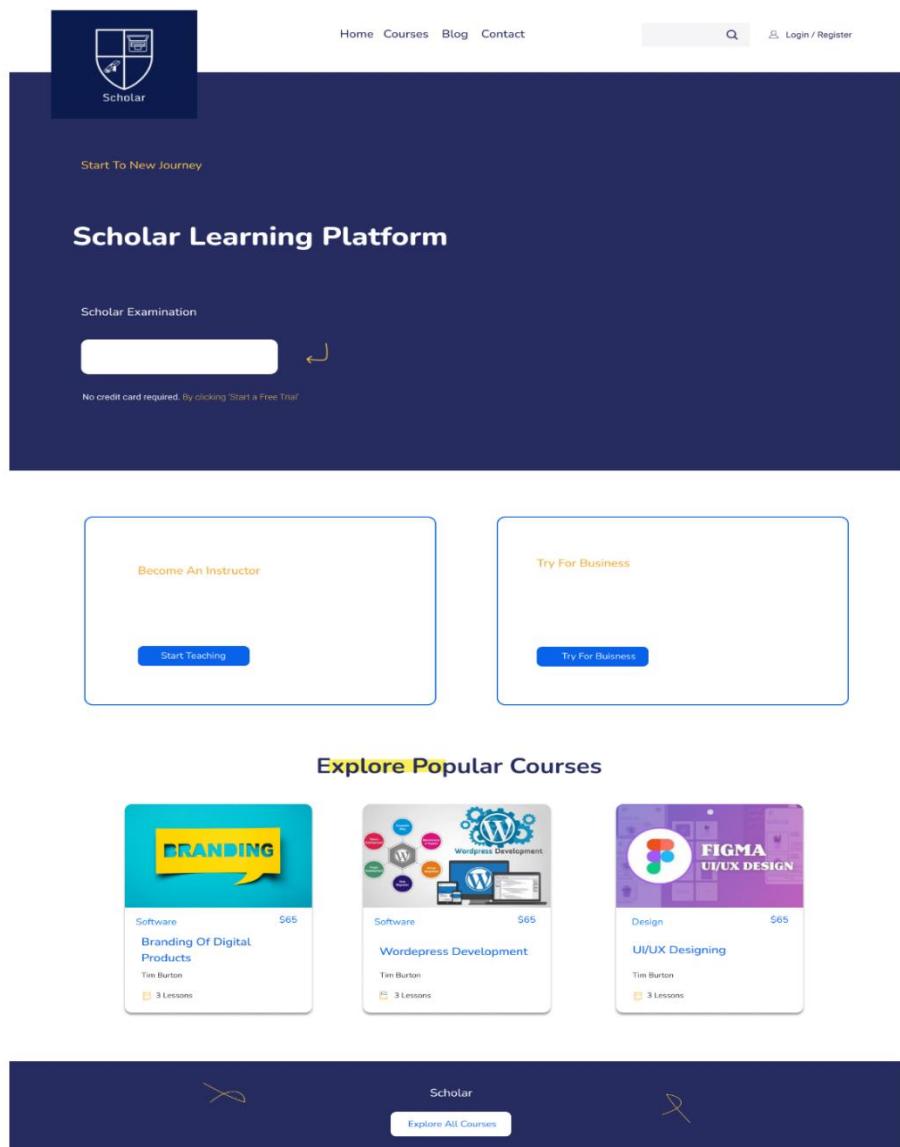


Figure 296 Unification with LMS (Prototype Design)

7.7.1.2. Mobile compatibility:

Making sure the online exam system is mobile-friendly is essential as more students access online resources through their mobile devices. By developing a mobile app that allows students to take examinations on their phones or tablets, accessibility and convenience could be increased.

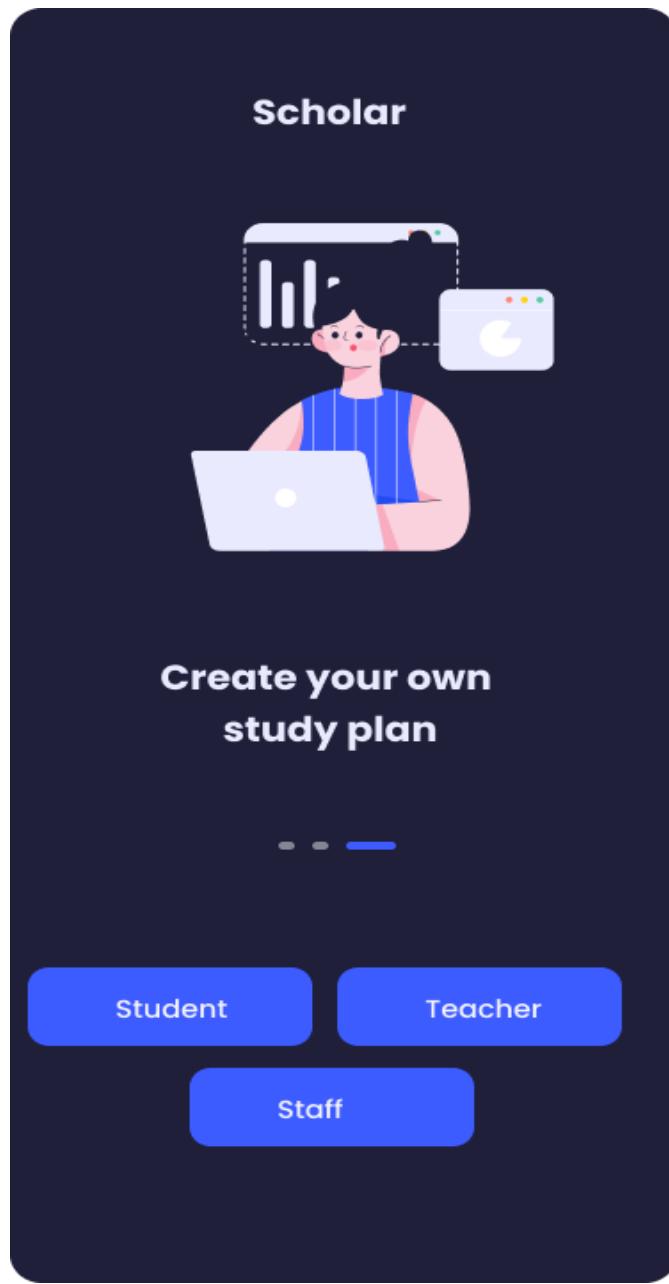


Figure 297 Mobility compatibility (Prototype Design)

7.7.1.3. Integrating multiple languages:

Making the system that supports multiple languages make it easier for students who don't speak and write English.

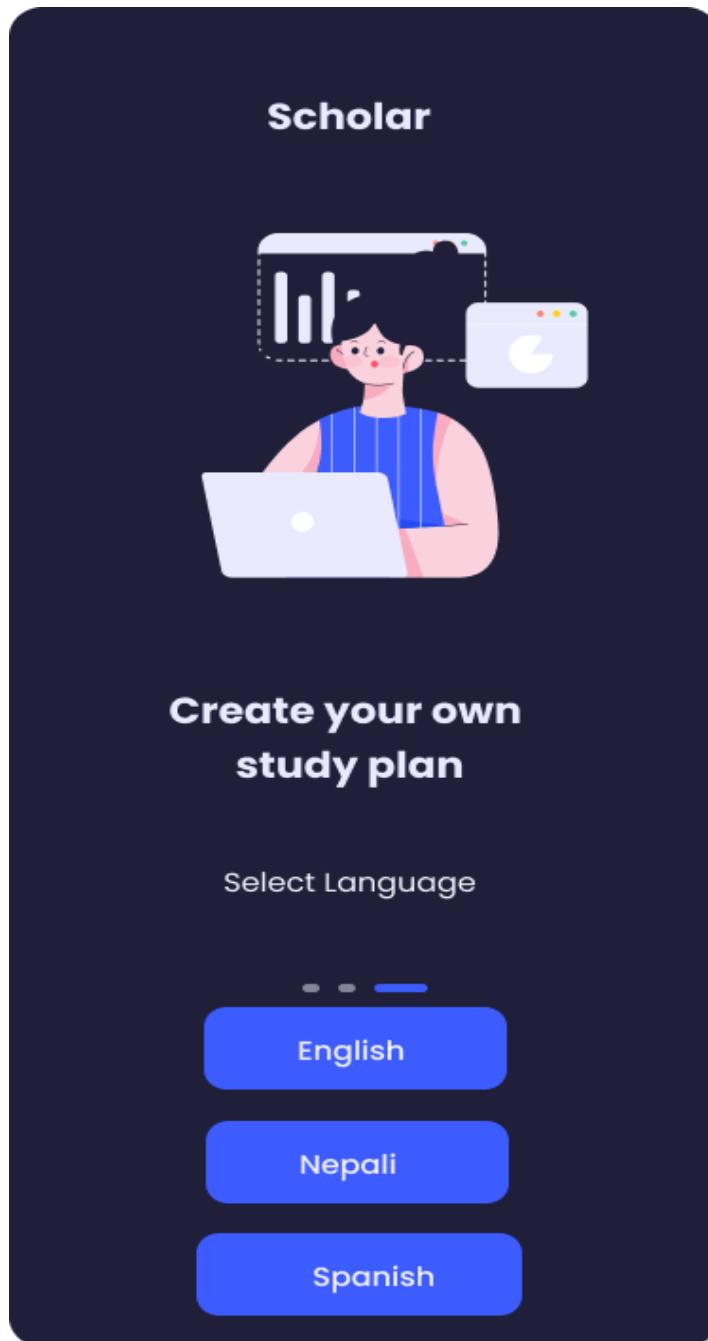


Figure 298 Integrating multiple languages (Prototype Design)

7.7.1.4. Adaptive testing:

Testing that is adaptive increases the level of difficulty of the questions based on the student's prior knowledge. The development of an adaptive testing framework may improve the accuracy and fairness of exam results.

7.7.1.5. Accessibility:

The accessibility of the online exam platform can be increased for students with impairments by including features like keyboard navigation, text-to-speech, and screen readers. This is one of the major future plans for Scholar.

7.7.1.6. Integration of AI:

By integrating AI into online exam systems, instructors may provide students with rapid feedback, analyse their performance, and generate insights that will improve the evaluation process.

7.7.1.7. Integration of data science:

Implementing different Python tools and applying key visualization ideas from data science. Future research on this subject will involve students visualizing and analysing up their results. The results of the students can also be seen and examined by the teacher and staff. This makes learning fun by incorporating competition and visualization.

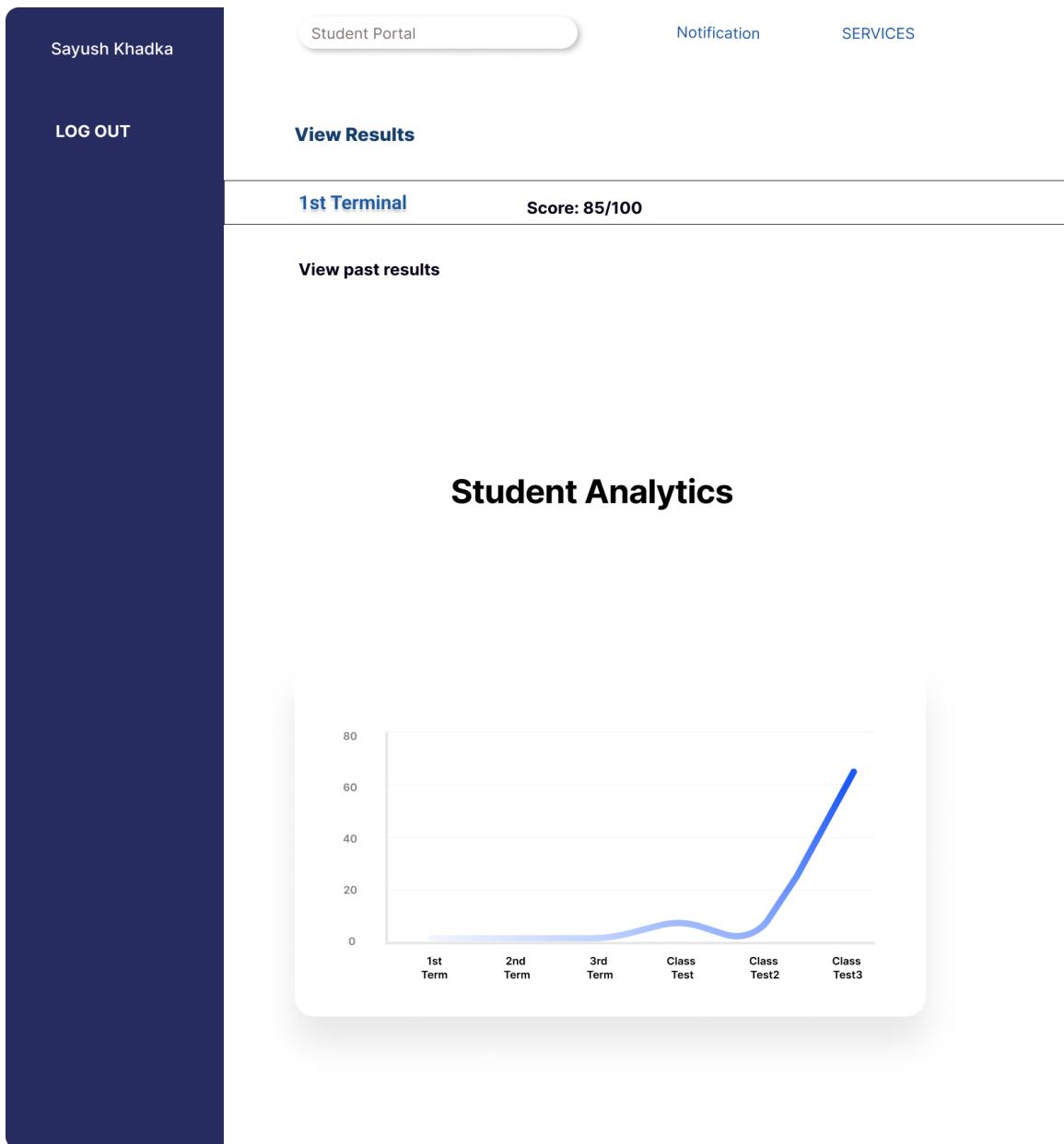


Figure 299 Integration of Data Science (Prototype Design)

7.7.1.8. Allowing the test-takers to conduct examination with different answer types:

For the time being, test-takers can only conduct MCQs in Scholar. Future plans include for letting test-takers administer exams in which participants can provide brief or long answers or write essays in accordance with the requirements of the exam and the test-taker.

The screenshot shows a user interface for an examination system. At the top, there are two buttons: "Teacher Portal" on the left and "Conduct Exam" on the right. Below these, a section labeled "Computer" is visible. The main area is titled "1st Terminal". It contains a question box with the text "1. Who was the father of computer?" followed by a large empty text input field. Below this, another question box contains the text "2. When was the first computer made?". Underneath are four answer options, each in its own box: "1945 AD" (unchecked), "1500 AD" (unchecked), "1290 AD" (unchecked), and "None of the above" (checked). The "None of the above" option is highlighted with a green background.

Figure 300 Allowing the test-takers to conduct examination with different answer types (Prototype Design)

7.12.9. Blockchain Integration:

By supplying a tamper-proof record of student grades, integrating blockchain technology into the online exam system can help to boost security and prevent cheating. This also falls as a major plan under future work.

7.12.10. Interaction Tools:

Including interaction tools can encourage more cooperative learning among students and allow them to work together on group tasks. Real-time chat and collaborative whiteboards are some of the interaction tools what Scholar plan to implement in the project.

7.12.11. User personalisation features:

Personalization can increase motivation and engagement by allowing students to tailor their exam experience, for as by selecting their preferred interface or the types of questions they want to answer.

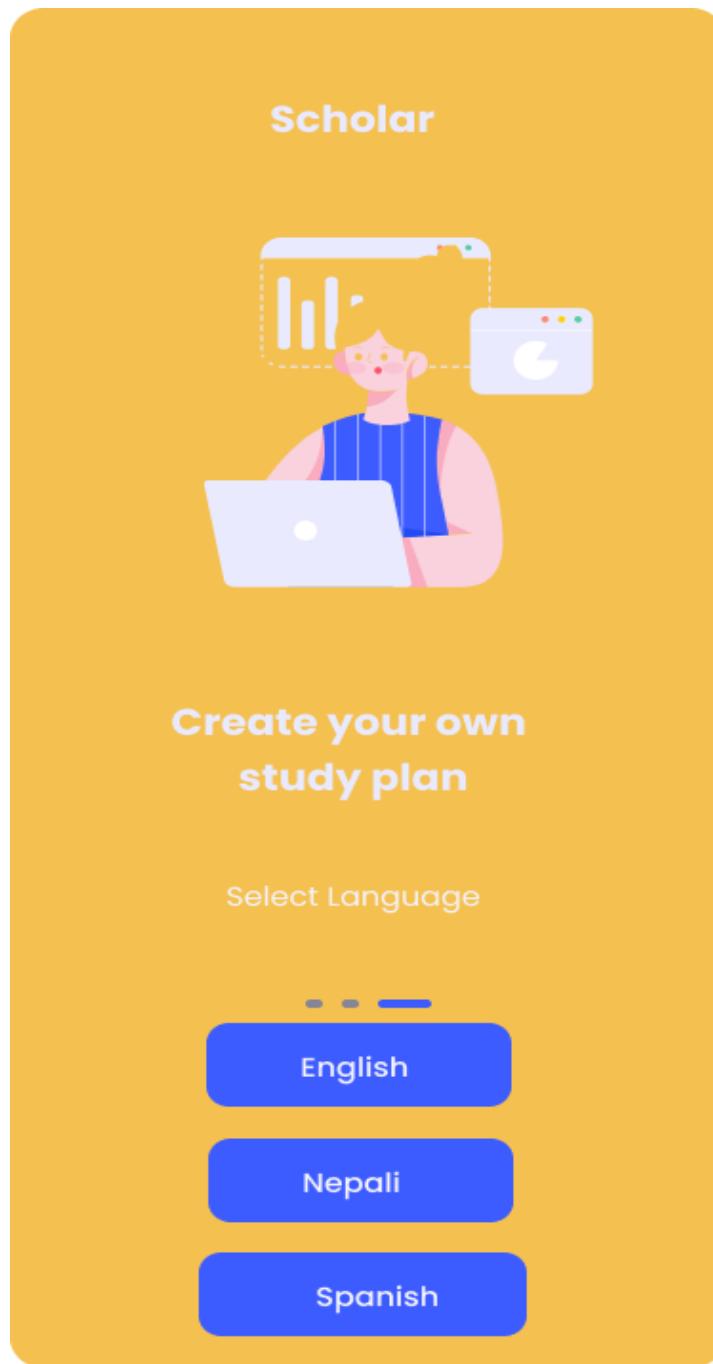


Figure 301 User personalisation features (Preferred Interface) (Prototype Design)

7.7.2. Readings for the Future Work:

Creating View in (views.py) :

```
from django.shortcuts import render
from .models import*
import pandas as pd
# Create your views here.

def home(request):
    item = Student.objects.all().values()
    df = pd.DataFrame(item)
    mydict = {
        "df": df.to_html()
    }
    return render(request, 'index.html', context=mydict)
```

Html Template Code for Data frame display in table format :

```
<html>
<body>
{{df|safe}}
</body>
</html>
```

Output

Figure 302 Working with pandas on Django



Manas Kinkar

Mar 6, 2019 · 1 min read · Listen



Integrating Speech to Text feature in a Django app.

Django is a very promising framework to design a Web Application. For this article it is assumed that you are familiar with django.

Let's get started

We will be using Web kit for speech recognition that the client side, the resultant text is send to the server using a simple AJAX post request, the request us handled at the client side and the text can be used for further processing.

1. Configuring a Web Kit at the browser side:

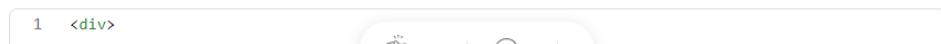


Figure 303 Using speech recognition feature

The screenshot shows the django CMS 3.11.1 documentation page for "How to serve multiple languages". The left sidebar contains navigation links for "Getting Started", "How-to guides", "Install django CMS by hand", "Use placeholders outside the CMS", "Serve multiple languages", "Create a multi-site set-up", "Work with templates", "Manage caching", "Enable frontend editing for Page and Django models", "Create sitemaps", "Manage Page Types", "Create plugins", "Create apphooks", "Manage complex apphook configuration", "Extend the Toolbar", and "Customise navigation menus". The main content area features a large heading "How to serve multiple languages". Below it, there is a paragraph about multilingual support and a code snippet for "urls.py". A sidebar on the right is titled "CONTENTS" and lists "Multilingual URLs", "Monolingual URLs", "Store the user's language preference", "Working in templates", "Display a language chooser in the page", "Get the URL of the current page for a different language", and "Configuring language-handling behaviour".

```
from django.conf.urls.i18n import i18n_patterns
from django.contrib import admin
from django.contrib.staticfiles.urls import staticfiles_urlpatterns
from django.urls import include, re_path
from django.views.i18n import JavaScriptCatalog

admin.autodiscover()

urlpatterns = i18n_patterns(
    re_path(r'^jsi18n/$', JavaScriptCatalog.as_view(), name='javascript-catalog'),
)
urlpatterns += staticfiles_urlpatterns()

# note the django CMS URLs included via i18n_patterns
```

Figure 304 Implementation of multiple languages



Abhinav Sharma

Oct 12, 2020 · 7 min read

• Listen



Live Whiteboard with Django, React, and WebSockets

Recently, I've been learning web development using Python, and wanted to implement real-time client-to-client interactions on an app. I thought a basic live whiteboard would be a good example of this. To do this, either long-polling or WebSockets would have to be implemented. For the purposes of this example, I went with WebSockets.

To start, we start by making sure Python is installed on our system, and you can do so using Homebrew. We also will need to use Redis to cache WebSocket data being sent to the server. This can also be installed with Homebrew.

```
brew install redis
brew services start redis
```

🕒 71 | ⚙ 2 |

Figure 305 Creative way of learning with the implementation of white board

Developing A Course Recommender System using Python



user.

Types of Recommender Systems

There are three broad categories of recommender systems:

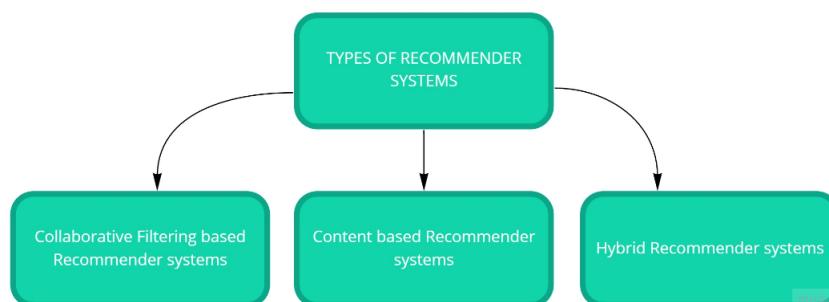


Image Source: [Google images](#)

Machine Learning

Become a full stack data scientist

Basics of Machine Learning	▼
Machine Learning Lifecycle	▼
Importance of Stats and EDA	▼
Understanding Data	▼
Probability	▼
Exploring Continuous Variable	▼
Exploring Categorical Variables	▼
Missing Values and Outliers	▼
Central Limit theorem	▼

Figure 306 After unification with Learning Management System; implementation of course recommendation

Developing A Course Recommender System using Python



Issues in Recommender Systems:

1. Cold Start Problem:

Whenever a new user enters a recommender system, the question arises of what to recommend him/her and on what basis as previous data is not available and similarity calculation could not be performed. One solution to this problem is to make the new users enter a small introduction form containing basic information about a person's interests, hobbies, occupation, and creating a basic user profile and then recommending items to the new user. This would solve the cold start problem to a great extent.

2. Data Sparsity Problem:

The major issue in a recommender system is the unavailability of appropriate data which is the main requirement for the recommendation process. Many users don't bother to review items they bought. As a result, the user-item rating matrix has many sparse entries which degrade the performance of the similarity calculation algorithm. So one solution is to predict sparse entries and many researchers have given algorithms to predict these ratings such as a negative weighted one slope algorithm.

Machine Learning

Become a full stack data scientist

Basics of Machine Learning	▼
Machine Learning Lifecycle	▼
Importance of Stats and EDA	▼
Understanding Data	▼
Probability	▼
Exploring Continuous Variable	▼
Exploring Categorical Variables	▼
Missing Values and Outliers	▼
Central Limit theorem	▼
Bivariate Analysis Introduction	▼

Figure 307 After unification with Learning Management System; implementation of course recommendation (continued)

7.8. Similar Projects (Elaborated):

7.8.1. Speed Exam:

In contrast to other platforms, SpeedExam provides a straightforward and logical process for building tests; users are not forced to build tests and add questions concurrently. The user can upload questions to create their question bank from which they can create tests as needed by uploading them here. The user can reuse their queries and might be able to provide their material more effectively (Exam, n.d.).

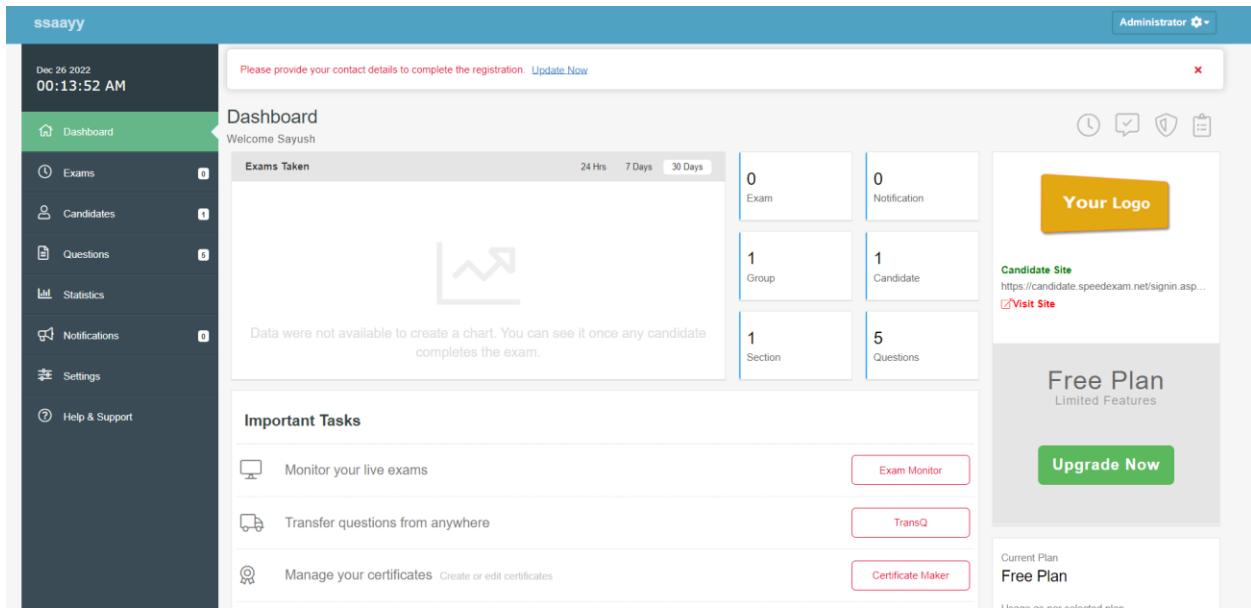


Figure 308 Speed Exam dashboard

Figure 309 Speed Exam create a new exam

Figure 310 Speed Exam candidates list

The screenshot shows the 'Add New Question' page within the Speed Exam application. The left sidebar displays the user's name 'ssaayy' and the date/time 'Dec 26 2022 00:16:16 AM'. The main content area is titled 'Add New Question' and includes fields for 'Question Type' (set to 'Multiple Choice (Radiobutton)'), 'Section' (set to 'Select Section'), and a rich text editor for the question text. Below this are four answer boxes labeled A, B, C, and D, each with a radio button and a set of editing icons. At the bottom of the form are 'Remove' buttons for each answer box. To the right of the main form is a 'Related Tasks' sidebar with links for 'Copy & Paste Questions', 'Import Questions', 'Export Data', and 'Question Feedback'. A large callout box titled 'TransQ' provides information about transferring questions between file formats.

Figure 311 Speed Exam add new question

7.8.2. Eklavvy:

Academics, professors, instructors, and even commercial organizations were consulted during the development of the Eklavvy platform. To increase the validity, precision, and scalability of academic tests and hiring evaluation procedures, Eklavvy is a simple Plug-and-Play SaaS platform. It might become tedious and exhausting to conduct each interview and examination by hand for institutions or larger organizations (Eklavvy, n.d.).

The system is accessible to everybody with a basic digital device because it has been designed to be both simple to use and incredibly effective at handling enormous workloads. A number of assessments have been taken using the Eklavvy platform all around the world. They may confidently assert that our features and services are unmatched in the domains of proctoring and digital exams (Eklavvy, n.d.).

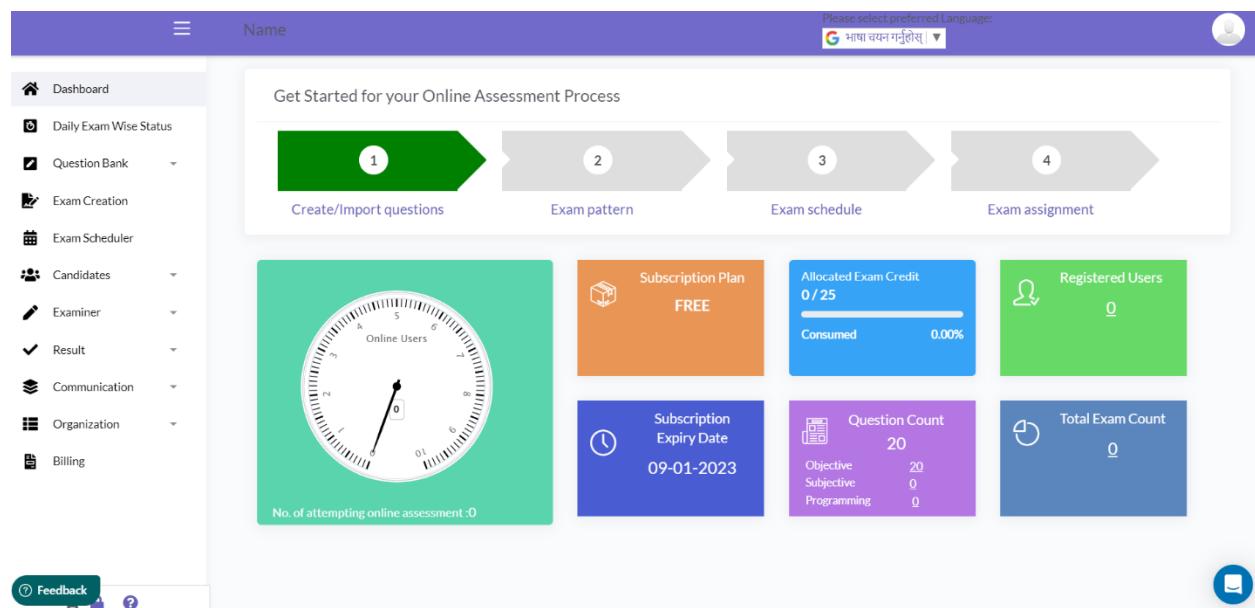


Figure 312 Eklavvy dashboard

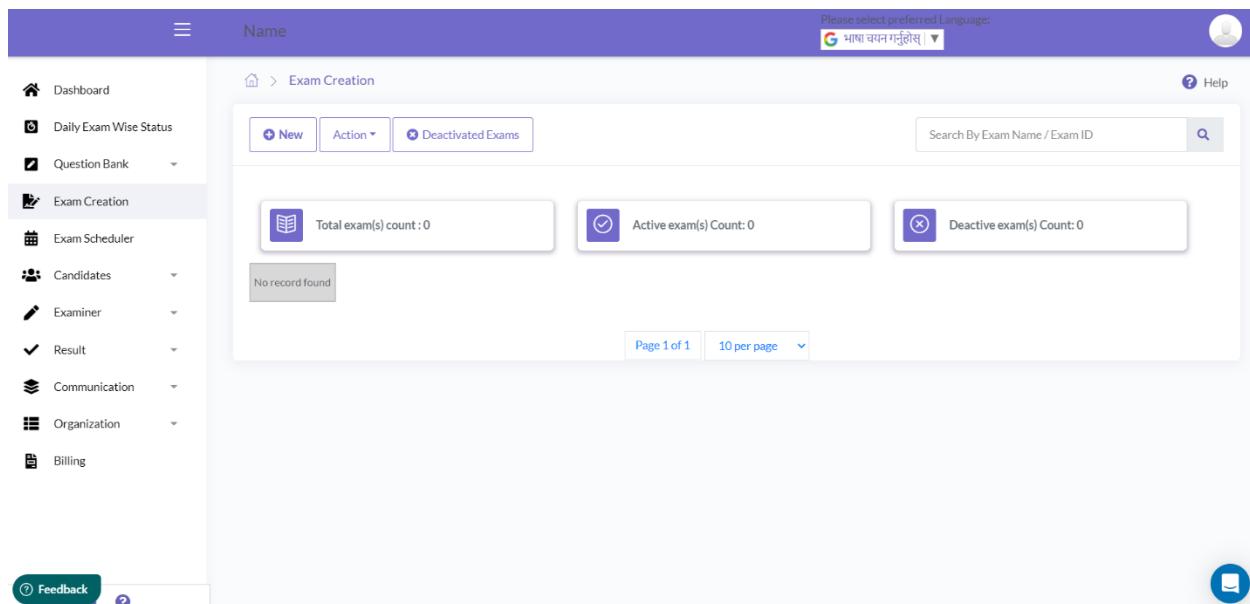


Figure 313 Eklavvy exam creation

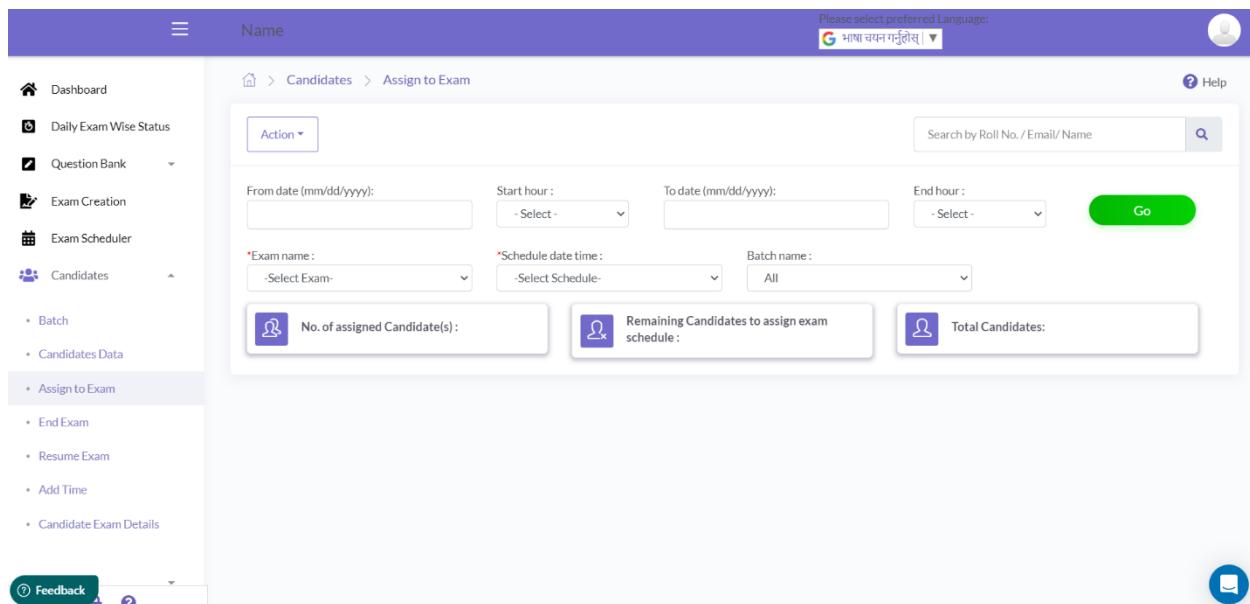


Figure 314 Eklavvy assign to exam

The screenshot shows the 'Candidate Result' section of the Eklavya application. The left sidebar includes links for Dashboard, Daily Exam Wise Status, Question Bank, Exam Creation, Exam Scheduler, Candidates, Examiner, and Result (with Candidate Result selected). The main area has a breadcrumb path: Home > Result > Candidate Result. It features a search bar ('Search by roll number or name') and filter options for From date, Start hour, To date, End hour, Exam, Schedule date time, Batch, and Fraud suspected. A button labeled 'Go' is present. A callout box displays 'All assigned candidate count:0'. The top right corner shows language selection ('Please select preferred Language: English | Marathi').

Figure 315 Eklavya candidate results

The screenshot shows the 'Assign Examiner' section of the Eklavya application. The left sidebar includes links for Dashboard, Daily Exam Wise Status, Question Bank, Exam Creation, Exam Scheduler, Candidates, Examiner (with Examiner Dashboard selected), and Result. The main area has a breadcrumb path: Home > Examiner > Assign Examiner. It features a dropdown menu for 'Action'. Filter options include From date, Start hour, To date, End hour, Exam name, Schedule date time, Batch name, Select Subject, and Select Topic. A callout box displays 'Total No of Candidate(s):'. The top right corner shows language selection ('Please select preferred Language: English | Marathi').

Figure 316 Eklavya assign examiner

7.8.3. Class Marker:

Since 2006, companies and academic institutions have utilized ClassMarker to create, administer, and grade their online quizzes and tests. This online testing method offers immediate test results for both the exam administrator and the test takers. Big companies, academic institutions, and individual instructors can benefit from ClassMarker, an online testing service provider with an easy-to-use quiz interface (ClassMarker, n.d.).

The premium-hosted online testing solution, which is easy to use across all widely used web browsers and satisfies all assessment requirements, is the greatest alternative for online testing (ClassMarker, n.d.).

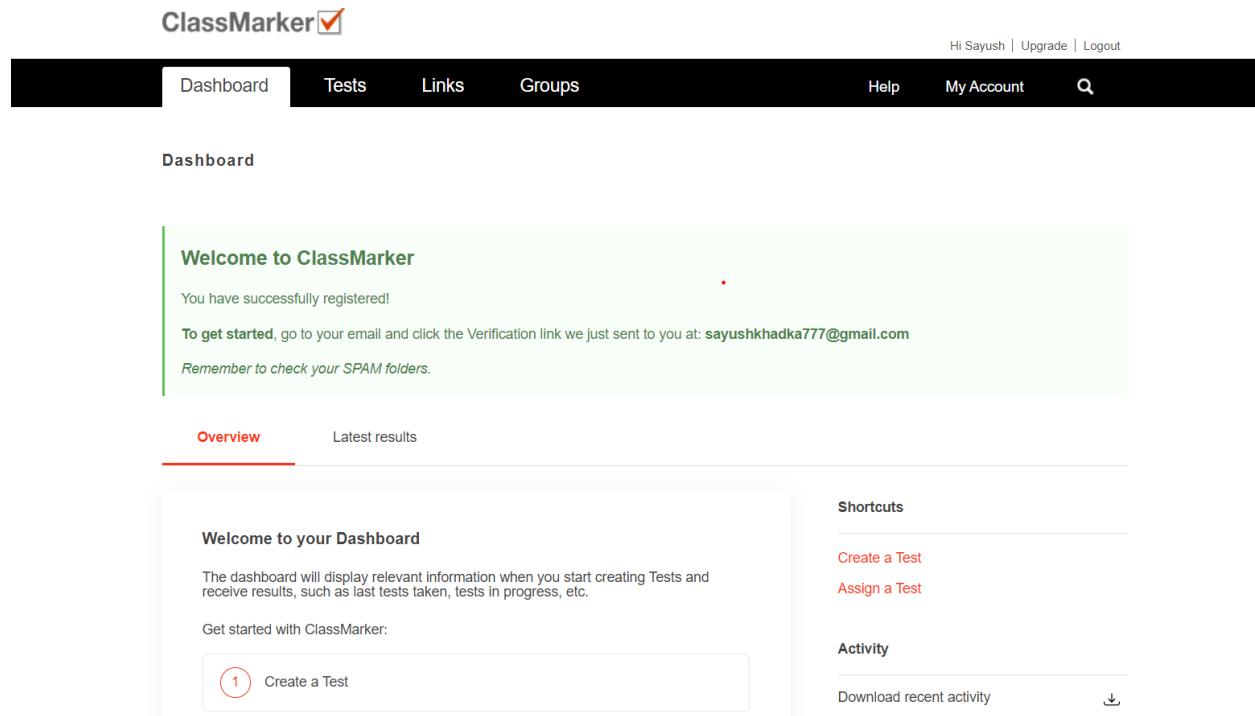


Figure 317 Class Marker dashboard

Figure 318 Class Marker create test

Question

(A) This answer option is correct

Answers ?

(A) This answer option is correct

(B) This answer option is correct

(C) This answer option is correct

Instructions

- Question Examples and Guides
- Copy & Paste Symbols
- Learn how to write great Tests

Figure 319 Class Marker set questions

7.9. Considered Methodology (Elaborated):

7.9.1. Waterfall methodology:

According to the waterfall technique, software development is carried out sequentially or linearly. Several jobs make up the project's division, with phases denoting the highest-level grouping. For a phase to be declared complete and include clear departure criteria, which is frequently a sign-off from the project stakeholders, the phases must be completed in the correct order (Sherman, 2015).

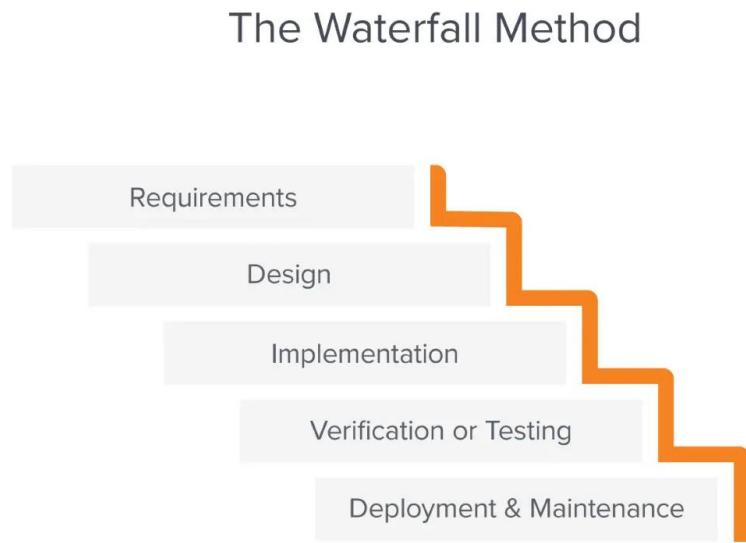


Figure 320 Waterfall Methodology (Team, 2022)

Reasons for not choosing this methodology:

- i) Waterfall methodology heavily relies on teams following a set of steps that keep them always moving forward. The system can rarely be changed in traditional versions. As my product requires frequent changes in the design and implementation as required by the client.

- ii) My product requires the involvement of the client in processes like requirement gathering and initial user evaluation. As an internal process, the Waterfall methodology lays little attention on the end user or client involved in a project.

These are the reasons why the waterfall methodology has not been Implemented.

7.9.2. Agile methodology:

Through the course of the project's software development lifecycle, the agile approach promotes constant testing and development. The Agile methodology for software testing comprises concurrent development and testing as compared to the Waterfall model (Hamilton, 2022).

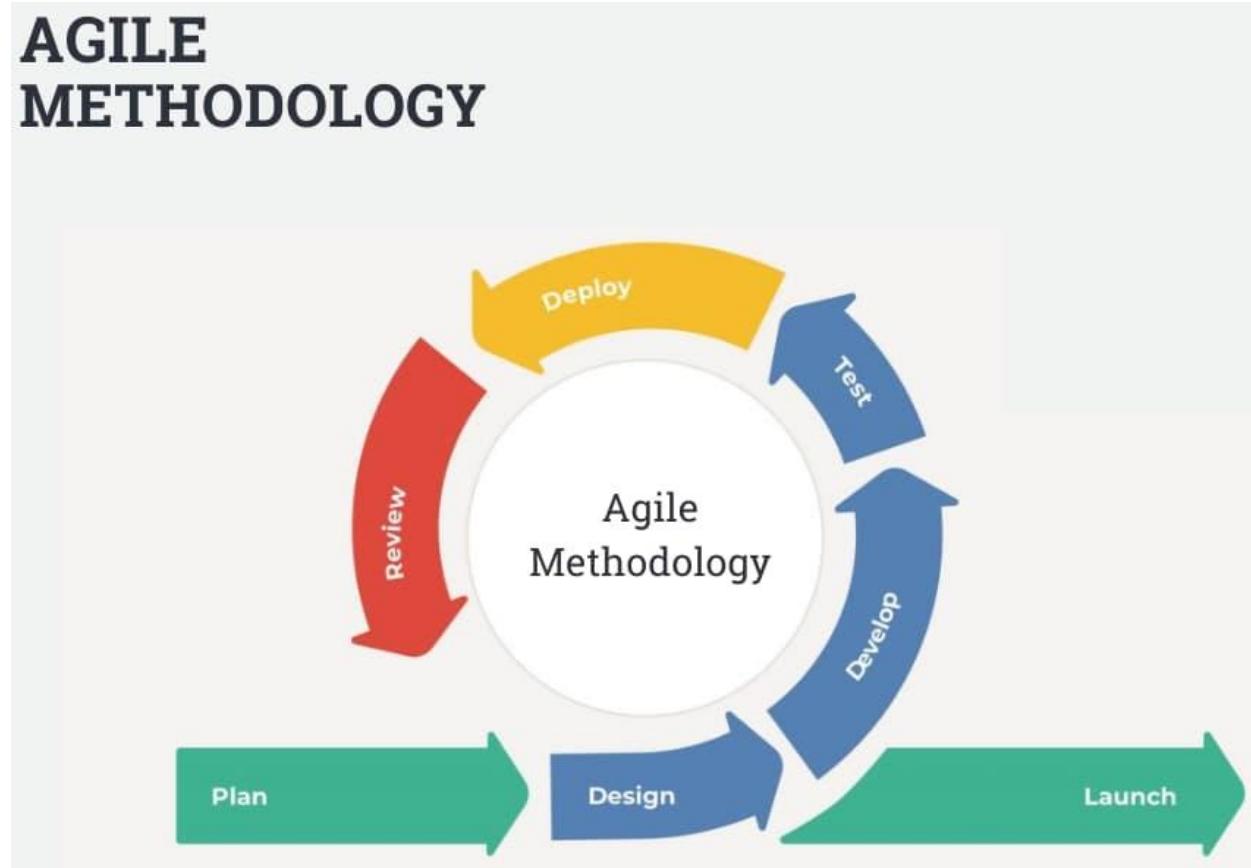


Figure 321 Agile methodology (InterQuality, n.d.)

Reasons for not choosing this methodology:

- i) Agile iterations are ideal for software development because they allow for the steady creation of smaller deliverables. However, this level of fragmentation would not work with a long-term project. My product is a long-term project and therefore this methodology is not suitable.
- ii) Agile methodology condenses massive volumes of data into shorter user stories with scant details. It will be difficult for me as a result to comprehend the client's requirements.

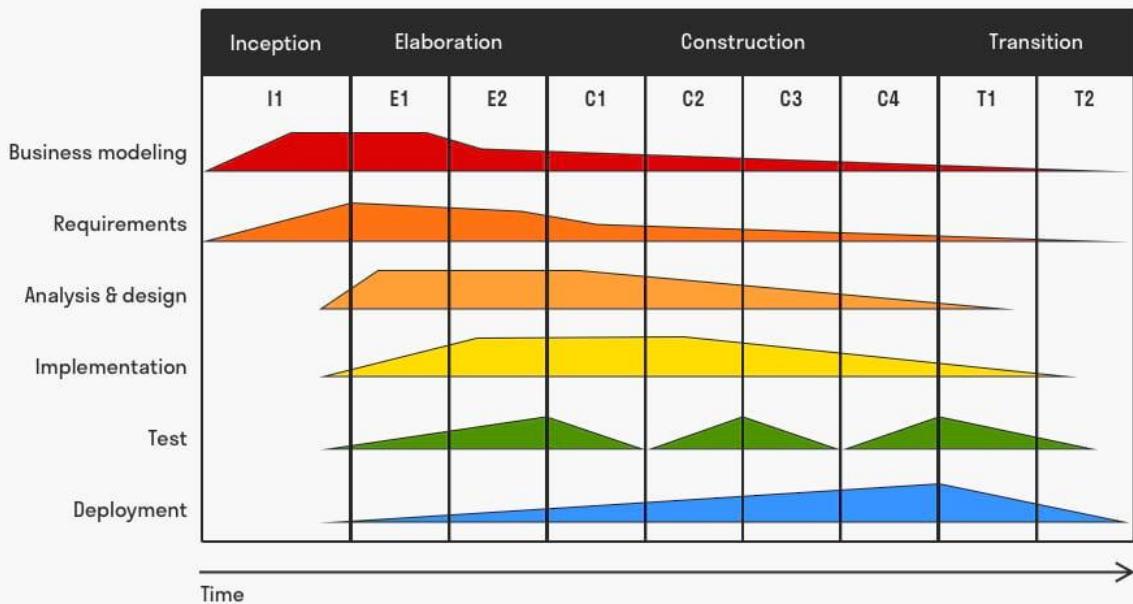
These are the reasons why agile methodology has not been implemented.

7.9.3. Rational Unified Process (RUP):

Rational Unified Process is an agile software development process that separates the four phases of the project life cycle. Business modelling, requirements, analysis and design, implementation, testing, and deployment are the six essential development disciplines that are used in each phase. Though some operations are more important and need more time at each stage than others. For instance, business modelling frequently takes place during the conception and elaboration phases. Before going on to the project's next stage, each of the four stages' main objectives must be accomplished. The main goal of RUP is to create high-quality software on a manageable timetable and budget. The life cycle phases can all be repeated as needed to complete the initial objectives. Once the transition phase has been completed, the project is complete (Bugajenko, 2021).

Rational Unified Process (RUP)

toolhero



www.toolhero.com

Figure 322 Rational Unified Process (RUP) (Janse, 2019)

Reasons for not this methodology:

- i) Implementing software development using this methodology is difficult since it uses a complicated process. Therefore, I didn't use this particular methodology in this project.
- ii) I can't apply this methodology for this project because it takes a lot of time and is expensive.

These are the reasons why Rational Unified Process has not been implemented.

7.10. Survey results analysis:

7.10.1. Pre-Survey results:

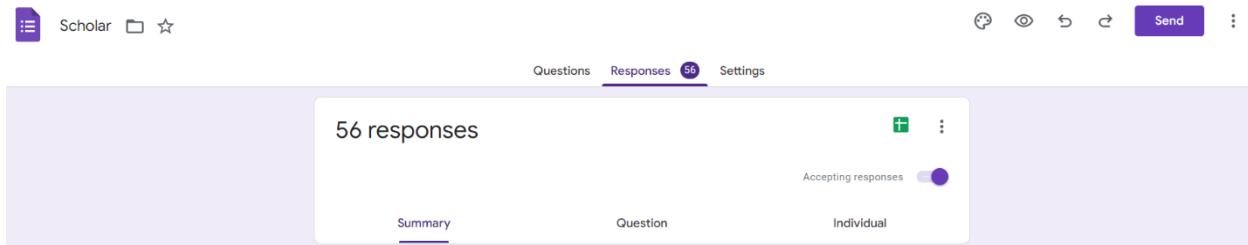


Figure 323 Survey Summary

A survey had been conducted for my application ‘Scholar’ for respondents’ decisions and opinions as well spread information on the use of technology properly in the educational sector to a level.

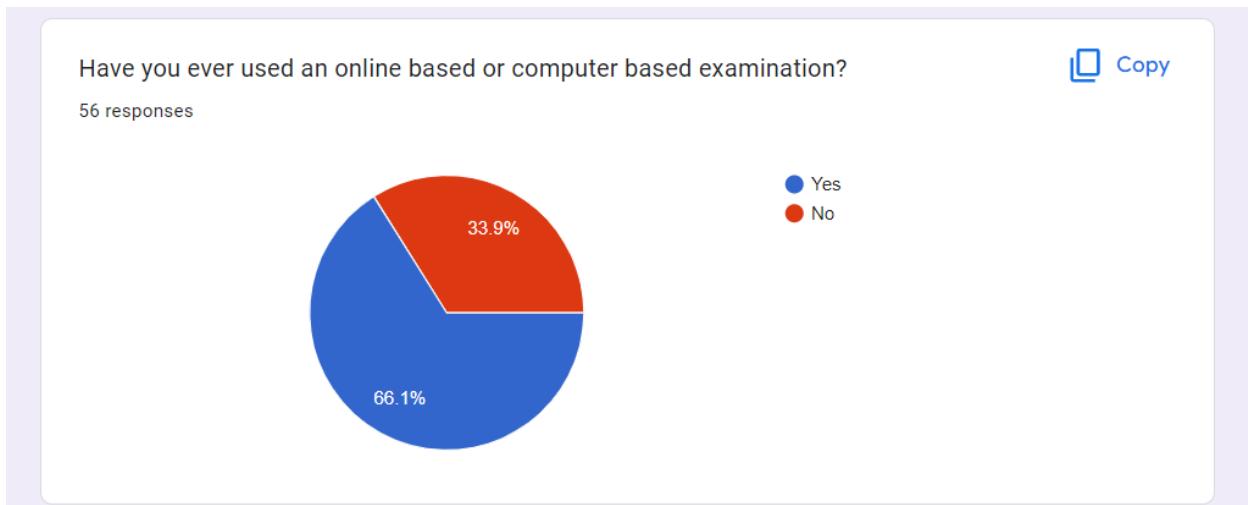


Figure 324 Have you ever used an online based or computer-based examination?

The above figure shows that many people have used online or computer-based examinations.

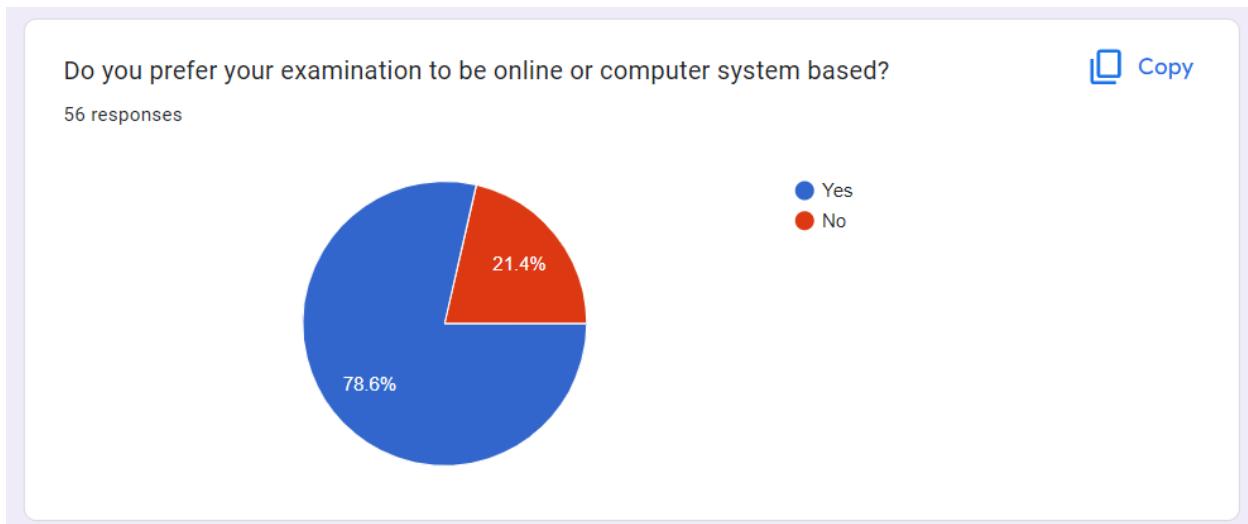


Figure 325 Do you prefer your examination to be online or computer system based?

The above figure shows that a greater number of people prefer online or computer-based examinations in comparison to traditional paper-based examinations.

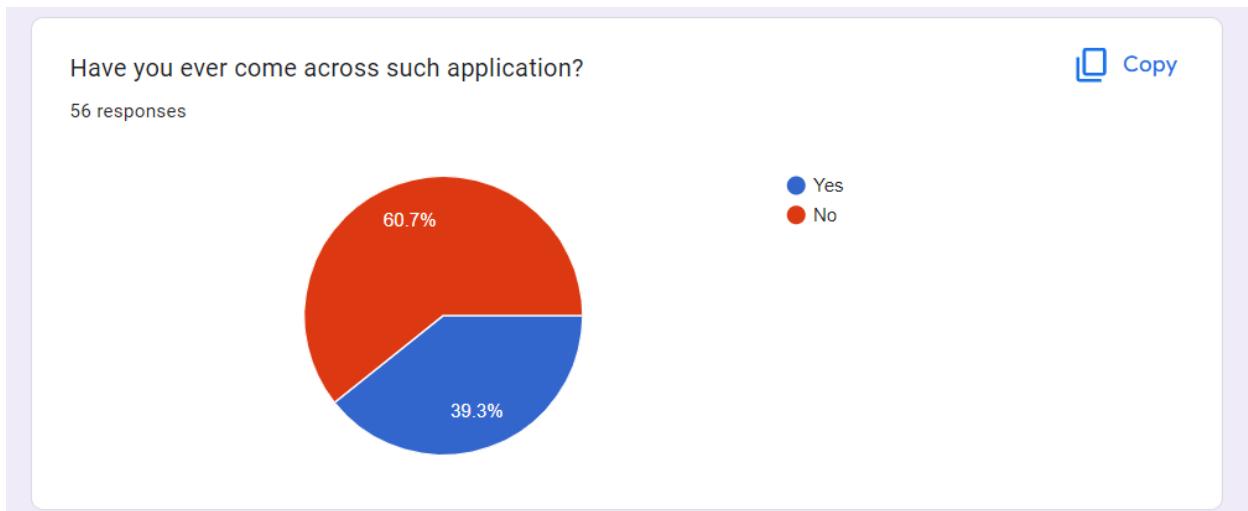


Figure 326 Have you ever come across such application?

The figure above shows that fewer people have come across an application like Scholar (an online examination system).

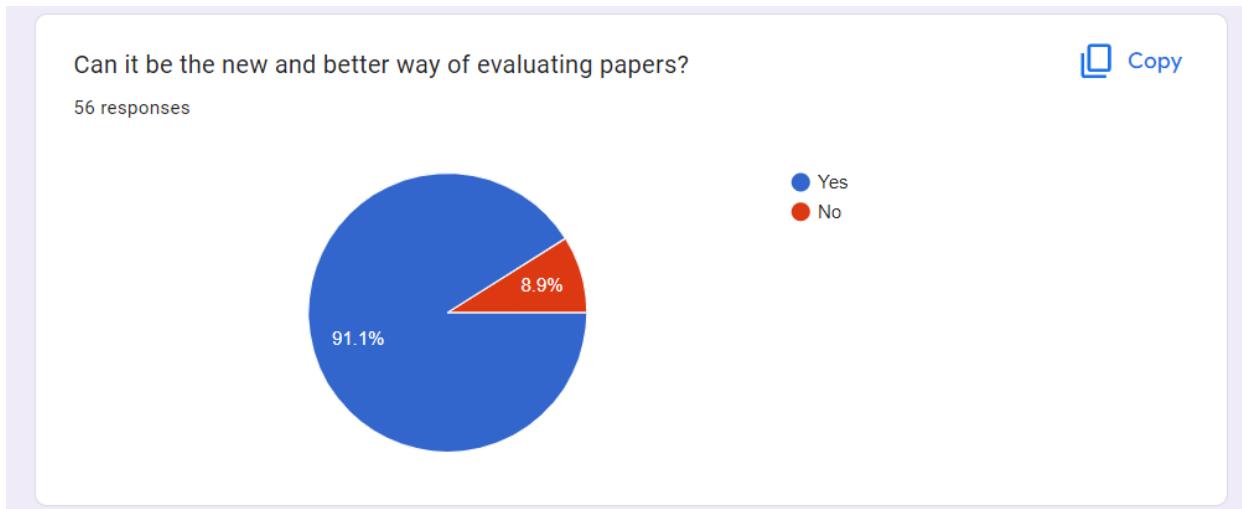


Figure 327 Can it be the new and better way of evaluating papers?

The above figure shows that the maximum number of people prefer the evaluation of their papers to be done through an online or computer-based system.

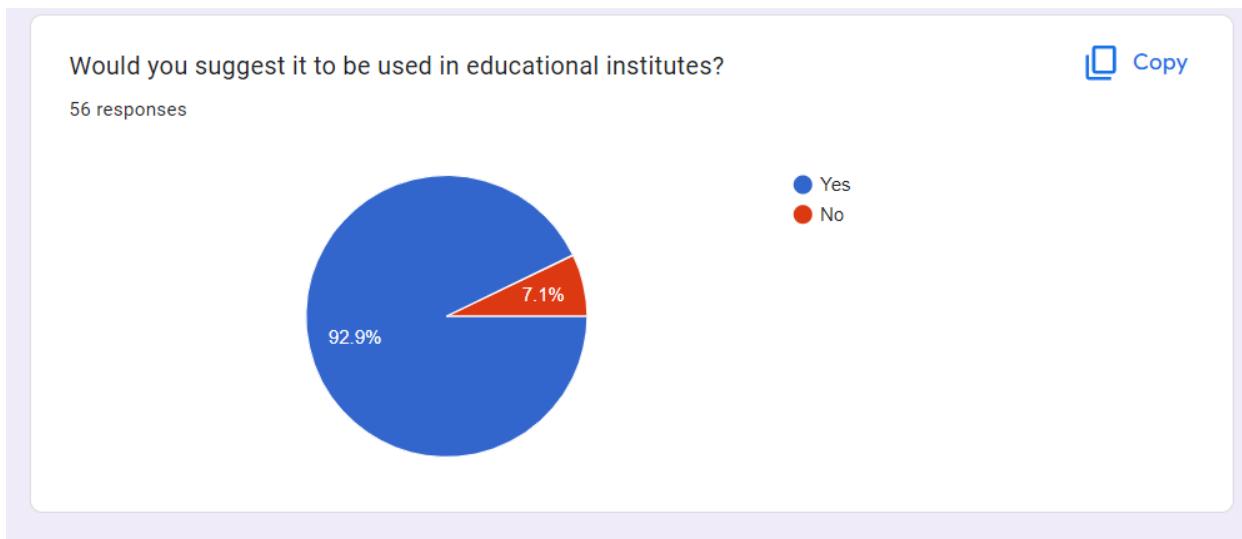


Figure 328 Would you suggest it to be used in educational institutes?

The above figure shows that the maximum number of people suggest that educational institutes implement applications like Scholar (online or computer-based examination).

Online examination is better than the traditional approach of conducting the examination. Why or why not?

38 responses

Online examination is better considering the world we live in today since we have access to technology but regardless traditional approach is also convenient to those area where they lack access to technology a

online examination is better because it is time efficient, environmental friendly and scalable with worldwide reach

Online examination might not be a better way of conductinh exams as it is difficult to monitor and manage time.

No ..due to the different unfortunate circumstance like network connecton, device problem etc.

Yes, because it allows us to analyze the data instead of memorizing it. I believe 21st century is an age of technology and information. You can get info anywhere but its up to us how we use those info. Online exam allows us such analytical approach of evaluating students' capacity.

Because it allows us to write and think freely on our own space.

It's more comfortable

Figure 329 Online examination is better than the traditional approach of conducting the examination. Why or why not? (Series 1)

The above figure shows the respondents' answers to why or why not online examination is better than the traditional approach of conducting an examination.

Online examination is better than the traditional approach of conducting the examination. Why or why not?

38 responses

However, if everything goes well and this problem is solved in future, online exam can be a better option at that time, but not in the current situation of the country.

Very nice

Advantages of Online Exams hlp4othr@gmail.com Online examination is better than the traditional approach of conducting the examination. Why There are several reasons why online examinations may be better than traditional examinations in certain situations: Convenience: Online examinations can be taken from any location with an internet connection, which is more convenient for students who may not be able to travel to a specific location to take an exam. Flexibility: Online exams can be scheduled at a time that is convenient for the student, which may not be possible with traditional exams that are held at specific times and locations. Reduced cheating: Online exams can be designed to minimize the opportunity for cheating, such as by using proctoring software that monitors the student's webcam and screen during the exam. Increased accessibility: Online exams can be made more accessible for students with disabilities by providing accommodations such as text-to-speech or larger font sizes. Cost savings: Online exams can be more cost-effective for institutions to administer, as they do not require the use of physical space or resources like paper and ink. Of course, there are also potential drawbacks to online exams, such as the need for reliable internet access and potential technical issues. It is important to carefully consider the specific needs and goals of a particular examination and determine the best approach based on those factors.

Figure 330 Online examination is better than the traditional approach of conducting the examination. Why or why not? (Series 2)

Some more answers on the respondents' answers to why or why not online examination is better than the traditional approach of conducting the examination.

According to the pre-survey conducted by Scholar,

- i) People have used online measures to conduct and take examinations. But on the other hand, they also have not come across such applications. This means even though they have appeared for online examinations, they have not used a proper application to conduct such online or computer-based examinations. They may have used some other ways to conduct the examinations online but not through a proper application.
- ii) Maximum people prefer the examination to be conducted through an online or computer-based system.
- iii) Maximum people suggest online, or computer-based examination would be better than the traditional way of paper-based examination.
- iv) Maximum respondents also suggest educational institutes should have online or computer-based examination systems.
- v) Respondents have given their answers to why or why not online examination is better than the traditional approach of conducting an examination. Most of them support that online examination is better than the traditional approach of conducting the examination.

7.10.2. Post-survey results:

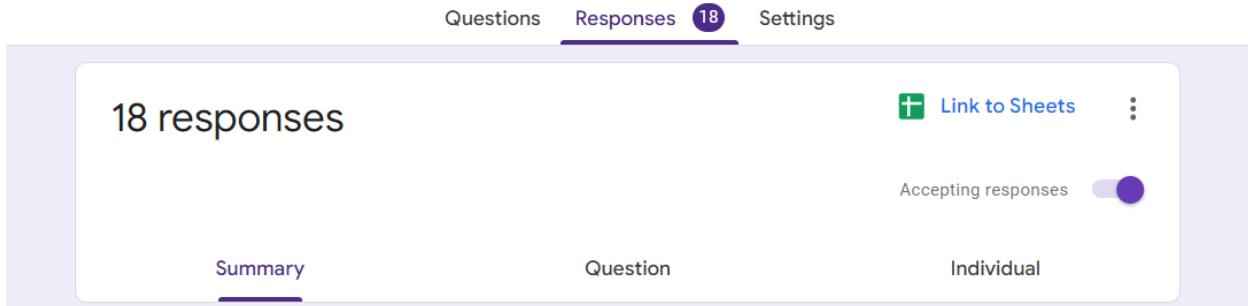


Figure 331 Survey summary



Figure 332 How would you rate your application experience?

The above result shows that users were very satisfied with the application experience.

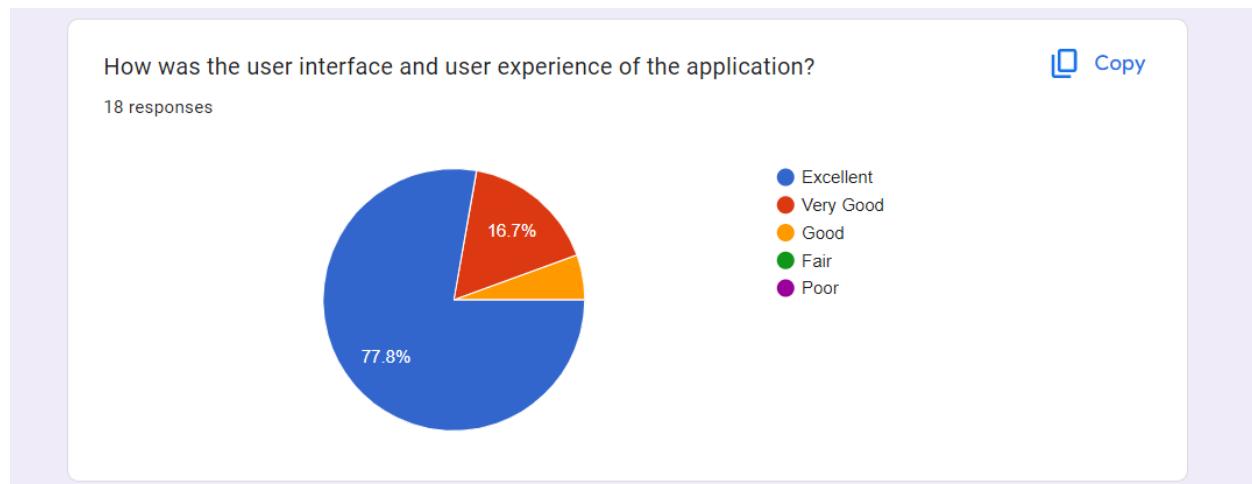


Figure 333 How was the user interface and user experience of the application?

The above figure shows that the end users thought the user interface and user experience was excellent.

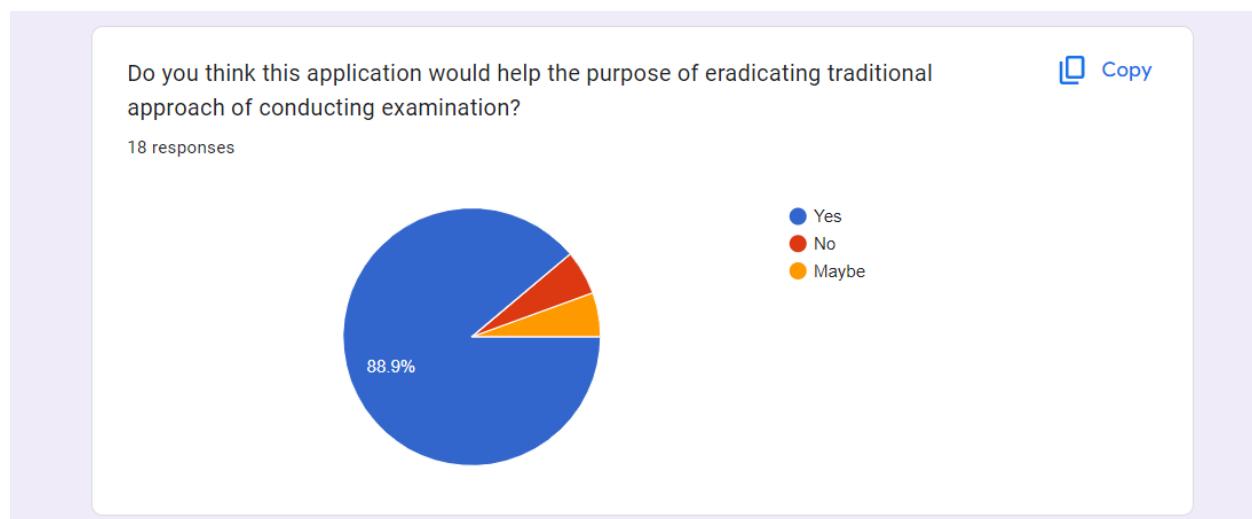


Figure 334 Do you think this application would help the purpose of eradicating traditional approach of conducting examination?

The above figure shows that maximum respondents think this application would help the purpose of eradication traditional approach of conducting examination.

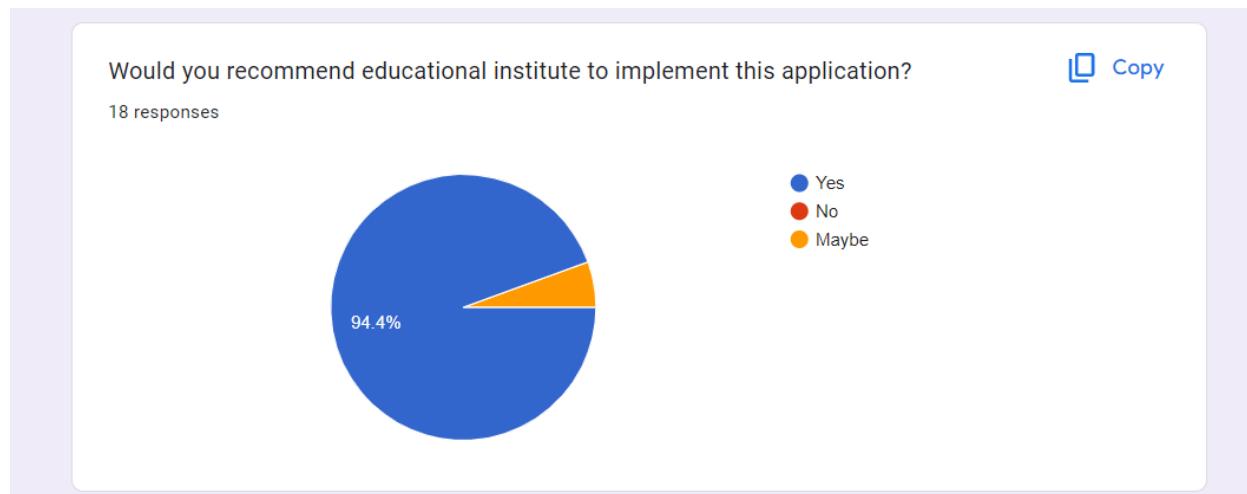


Figure 335 Would you recommend educational institute to implement this application?

The above figure shows that survey respondents would recommend educational institute to implement this application.

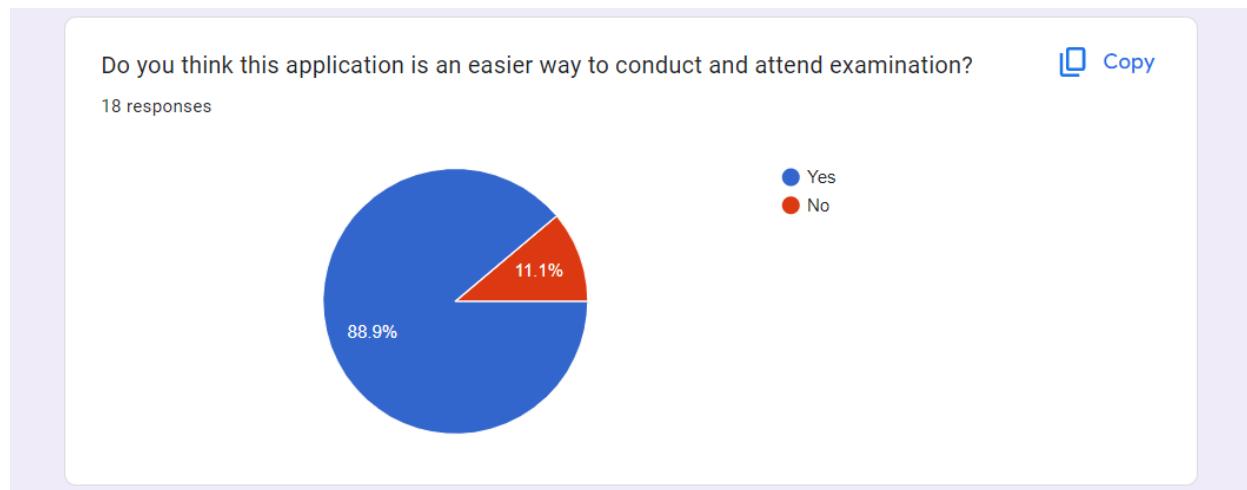


Figure 336 Do you think this application is an easier way to conduct and attend examination?

The above figure shows that maximum respondents think that this application is easier way to conduct and attend examination.

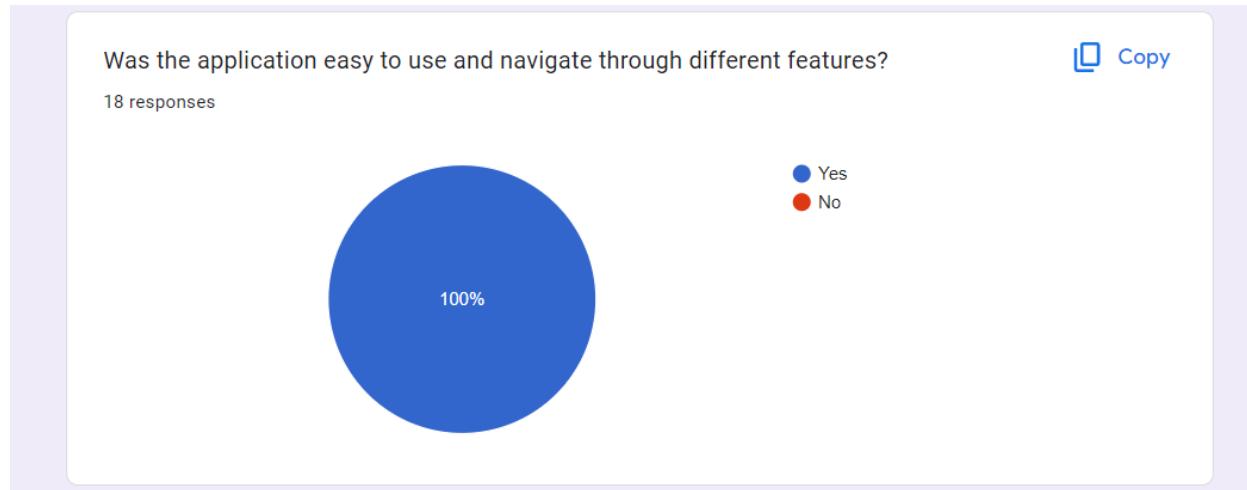


Figure 337 Was the application easy to use and navigate through different features?

The above figure shows that the application was easy to use and navigate through different features for every respondent.

According to the post-survey conducted by Scholar,

- i) Maximum respondents were very satisfied with the application experience.
- ii) Maximum respondents felt the user interface and user experience was excellent.
- iii) Maximum respondents think this application would help the purpose of eradicating traditional approach of conducting examination.
- iv) Maximum respondents would recommend educational institute to implement this application.
- v) Maximum respondents think this application is an easier way to conduct and attend examination.
- vi) All respondents think that the application was easy to use and navigate through different features.

7.11. System Testing:

7.11.1. User Login:

7.11.1.1. Test Case: To Test User Login (Non-existing username is given):

*Login feature is similar for every user (staff, teacher, student).

*Student has been selected for User Login test cases here.

Test No:	1
Objective:	To test user login (non-existing username is given).
Action:	i) Non-existing username is given.
Expected result:	Error message “Either username or password is invalid” would be displayed.
Actual result:	Error message was displayed.
Conclusion:	The test is successful.

Table 41 To test user login (non-existing username is given).

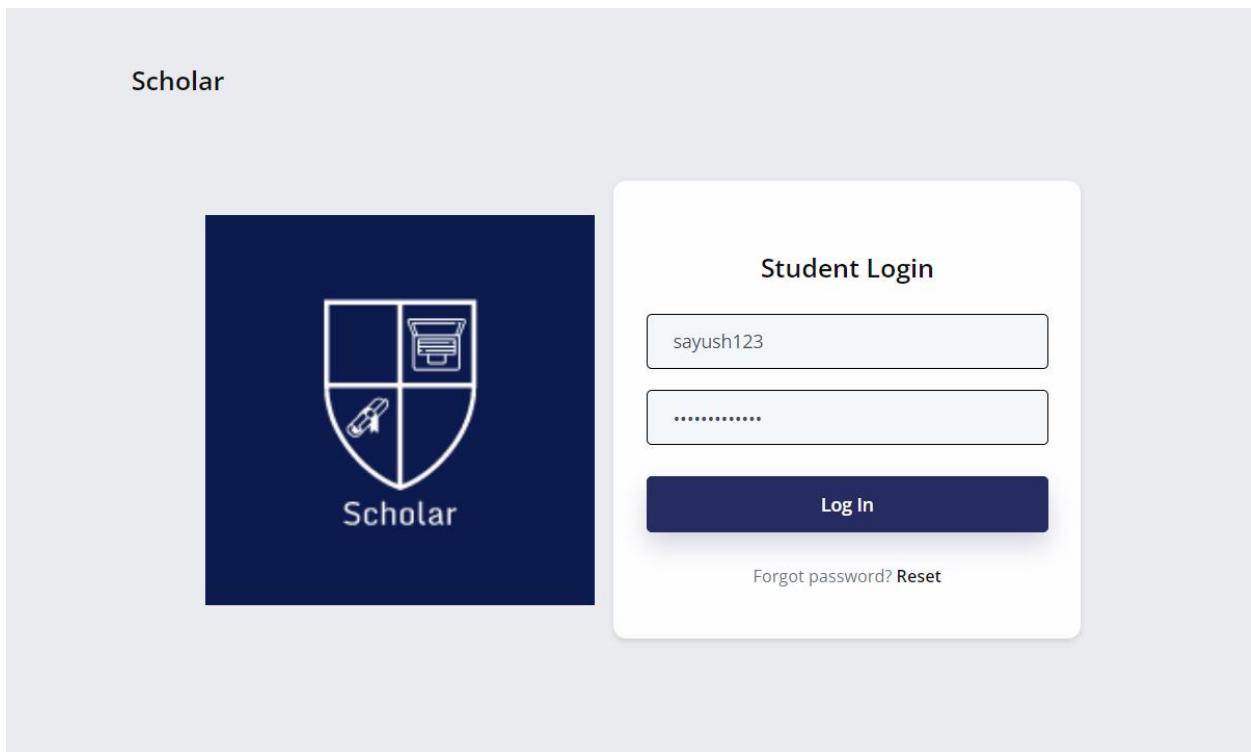


Figure 338 To test user login (non-existing username is given).

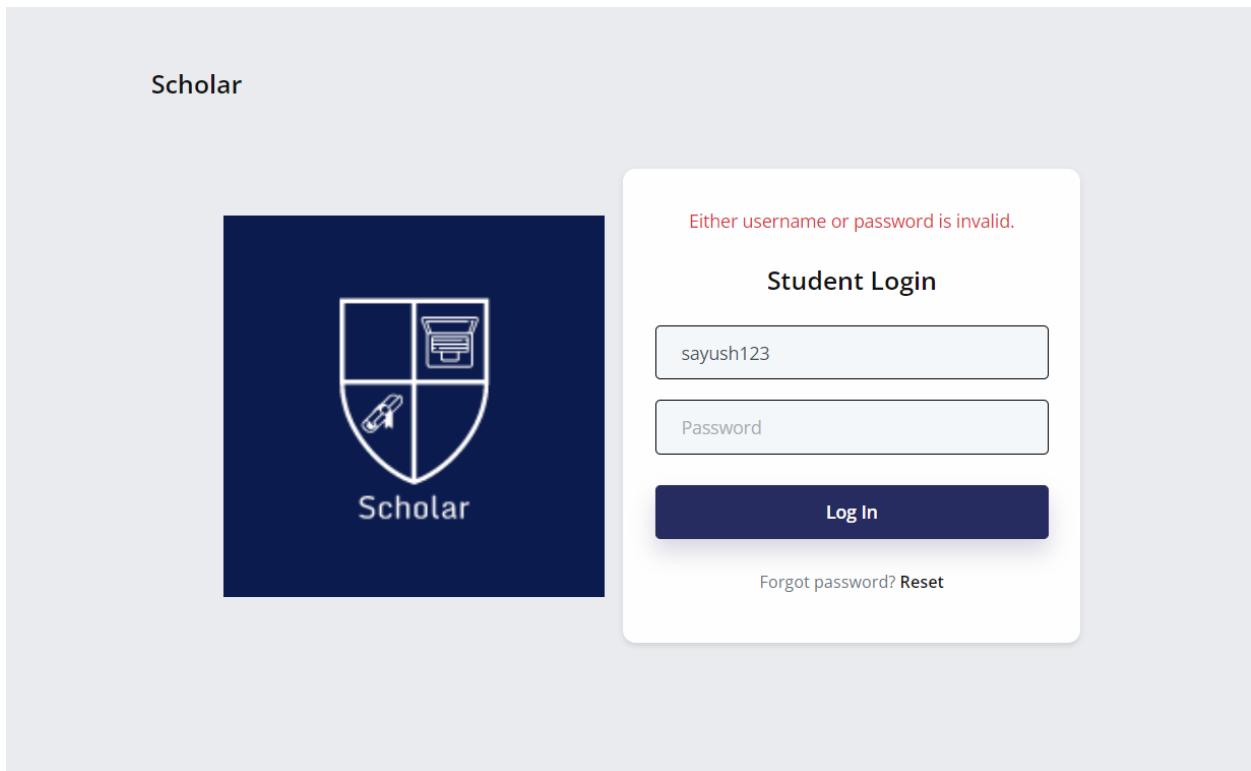


Figure 339 To test user login (non-existing username is given). (Test successful)

7.11.1.2. Test Case: To Test User Login (Invalid password is given):

Test No: 2	
Objective:	To test user login (invalid password is given).
Action:	i. Invalid password is given.
Expected result:	Error message “Either username or password is invalid” would be displayed.
Actual result:	Error message was displayed.
Conclusion:	The test is successful.

Table 42 To test user login (invalid password is given).

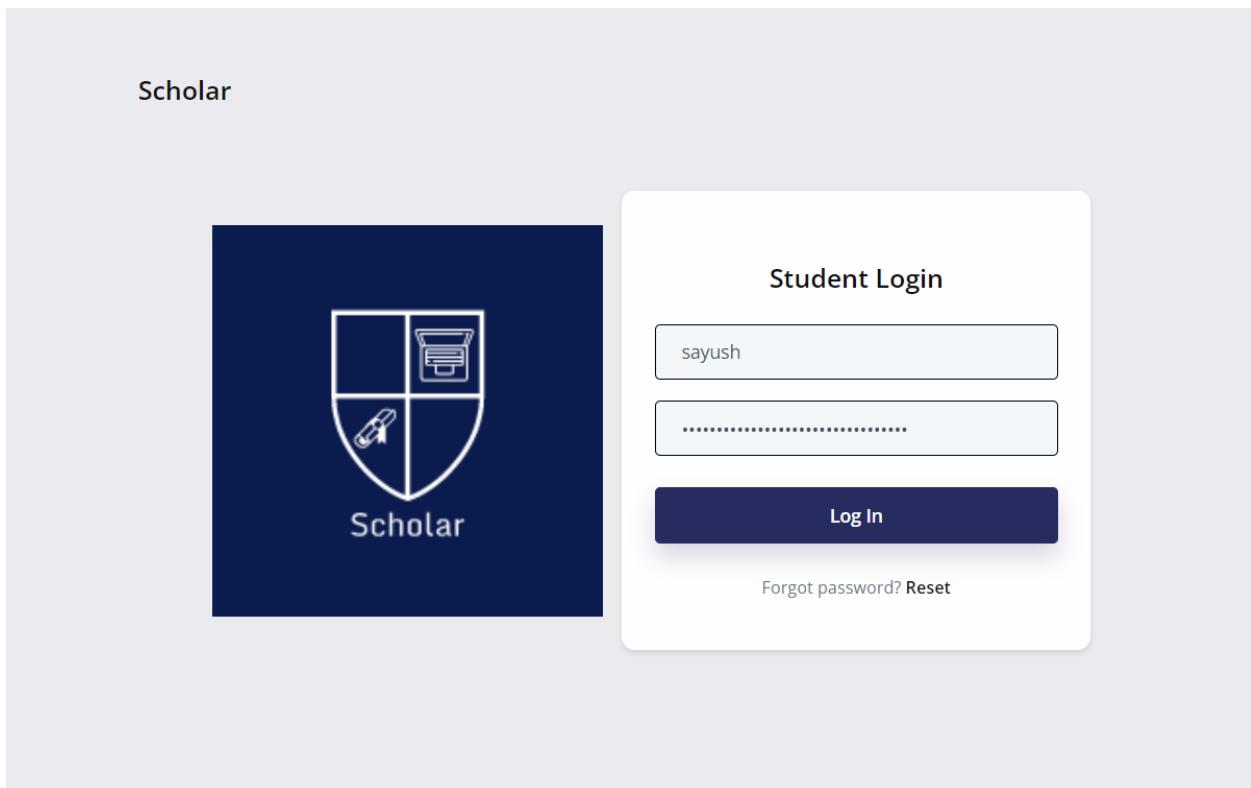


Figure 340 To test user login (invalid password is given).

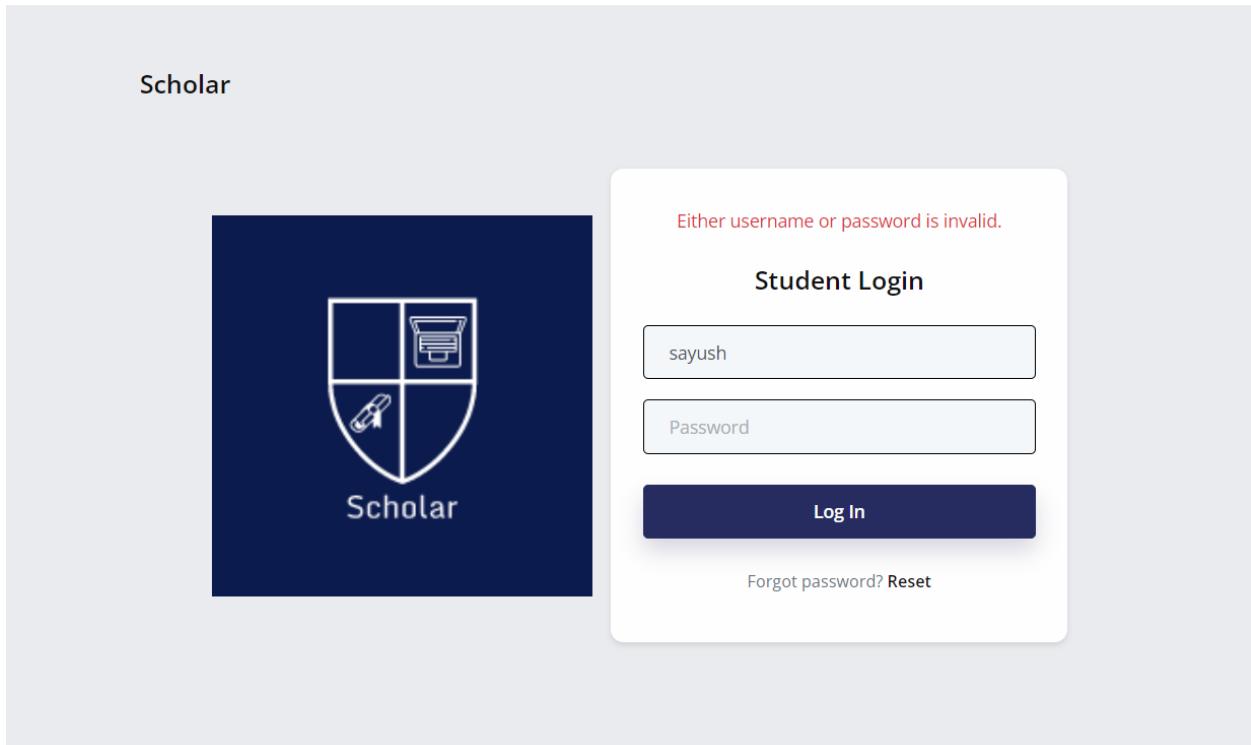


Figure 341 To test user login (invalid password is given). (Test successful)

7.11.2.3. Test Case: To Test User Login (Valid username and password are given):

Test No: 3	
Objective:	To test user login (valid username and password are given).
Action:	i) Valid username is given. ii) Valid password is given.
Expected result:	The user would be logged into the user portal (student).
Actual result:	The user was logged into the user portal (student).
Conclusion:	The test is successful.

Table 43 To test user login (valid username and password are given).

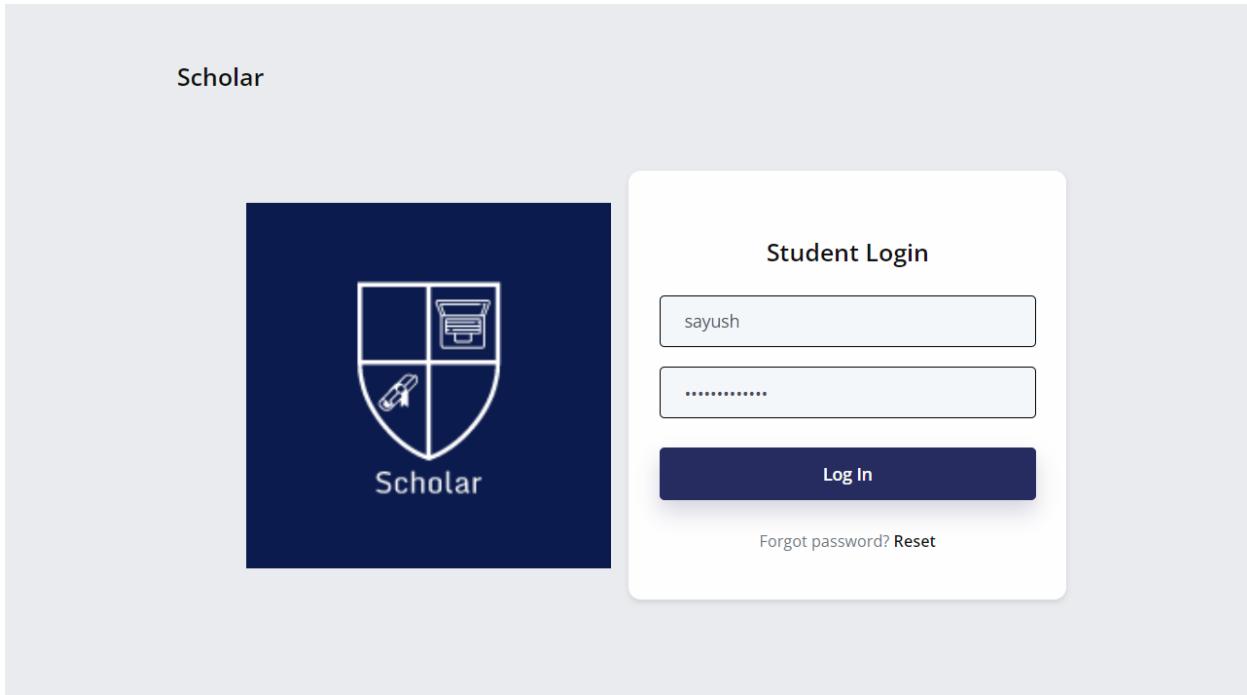


Figure 342 To test user login (valid username and password are given).

A screenshot of the "Student Portal" section of the Scholar application. The sidebar on the left shows the "Student Main" tab is active. The main content area is titled "Student Portal" and displays the message: "You are using the official student portal experience of Scholar Sayush Khadka.". To the right is a "Basic User Information" table. The table contains the following data:

Username:	sayush
First Name:	Sayush
Last Name:	Khadka
Email:	sayushkhadka777@gmail.com
Role:	Student
Class:	7
Section:	A
Subjects:	1. Maths 2. English

Figure 343 To test user login (valid username and password are given). (Test successful)

7.11.2. User registration:

*Teacher has been selected for User Registration test cases here.

7.11.2.1. Test Case: To Test User registration (Invalid email is given):

Test No: 4	
Objective:	To test user registration (invalid email is given).
Action:	i) Invalid email is given.
Expected result:	Error message “Enter a valid email address” would be displayed.
Actual result:	Error message was displayed.
Conclusion:	The test is successful.

Table 44 To test user registration (invalid email is given).

Scholar

Teacher Registration

John

Mayer

john@.com

john

.....

.....

Role

teacher

Class

9

Section

D

Subjects('ctrl' and 'click' to select multiple subjects)

Maths
English

Register

Get back to staff page: [Staff Page](#)

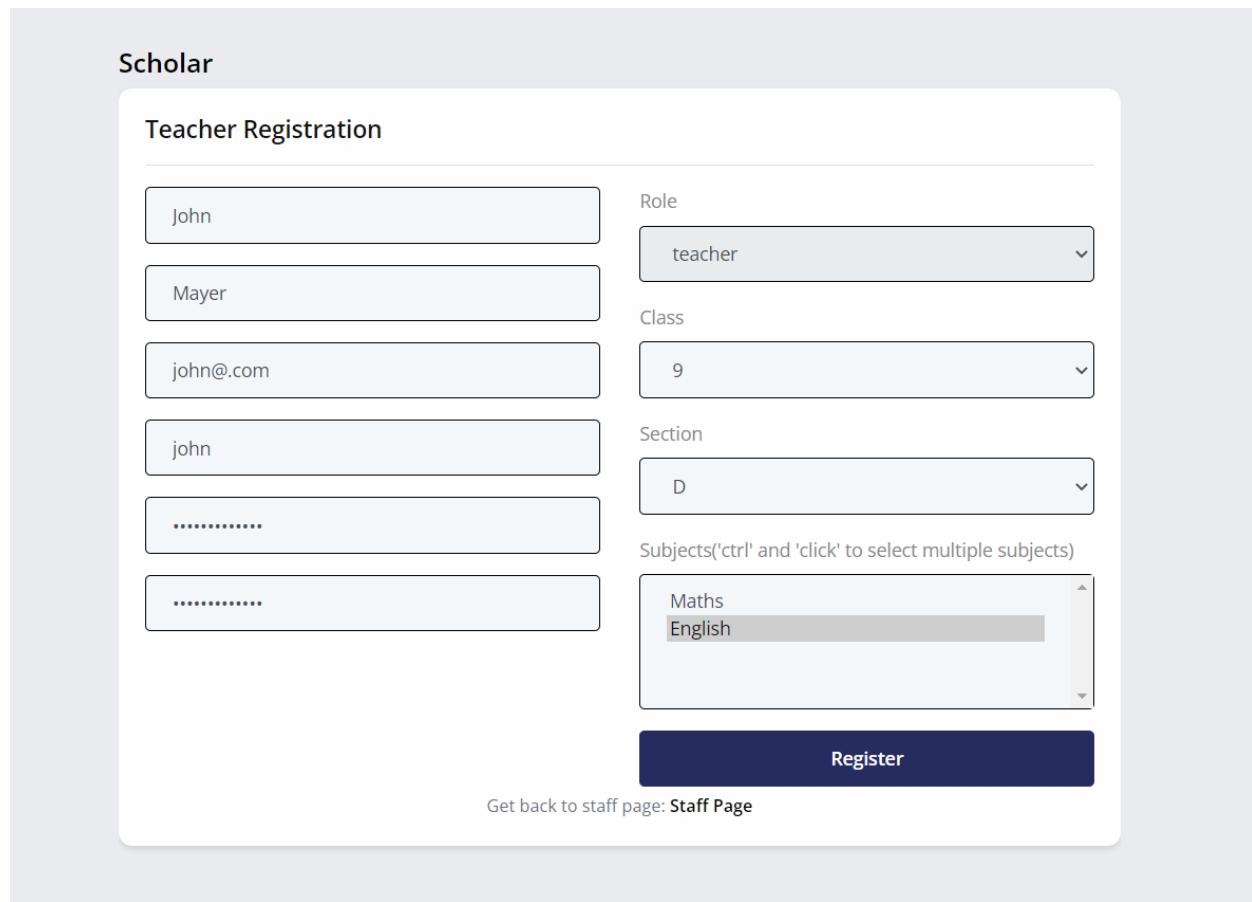
A screenshot of a 'Teacher Registration' form. The form fields include: First Name (John), Last Name (Mayer), Email (john@.com), Username (john), Password (two masked fields), Role (teacher), Class (9), Section (D), and Subjects (Maths, English selected). A note says 'Subjects('ctrl' and 'click' to select multiple subjects)'. A 'Register' button is at the bottom.

Figure 344 To test user registration (invalid email is given).

Scholar

Teacher Registration

John	Role
Mayer	teacher
john@.com	Class
• Enter a valid email address.	
john	Section
Password	Subjects('ctrl' and 'click' to select multiple subjects)
Password Confirmation	Maths English

Register

Get back to staff page: [Staff Page](#)

Figure 345 To test user registration (invalid email is given). (Test successful)

7.11.2.2. Test Case: To Test User Registration (Not fulfilling password requirements):

Test No:	5
Objective:	To test user registration (not fulfilling password requirements).
Action:	i) Short and common password is given.
Expected result:	Error message “This password is too short. It must contain at least 8 characters This password is too common” would be displayed.
Actual result:	Error message was displayed.
Conclusion:	The test is successful.

Table 45 To test user registration (not fulfilling password requirements).

Scholar

Teacher Registration

John	Role teacher
Mayer	Class 9
john@gmail.com	Section D
....	Subjects('ctrl' and 'click' to select multiple subjects) Maths English
....	

Register

Get back to staff page: [Staff Page](#)

The screenshot shows a mobile application interface for 'Teacher Registration'. At the top, it says 'Scholar' and 'Teacher Registration'. There are several input fields: 'Name' (containing 'John Mayer'), 'Role' (set to 'teacher'), 'Email' (containing 'john@gmail.com', which has a red validation error message: 'Enter a valid email address.'), 'Class' (set to '9'), 'Section' (set to 'D'), and a 'Subjects' dropdown containing 'Maths' and 'English'. Below these is a large blue 'Register' button. At the bottom, there's a link to 'Get back to staff page: Staff Page'.

Figure 346 To test user registration (not fluffing password requirements).

Scholar

Teacher Registration

John	Role teacher
Mayer	Class 9
john@gmail.com	Section D
john	
Password	
Password Confirmation	<ul style="list-style-type: none">This password is too short. It must contain at least 8 characters.This password is too common.
Subjects('ctrl' and 'click' to select multiple subjects) Maths English	

Register

Get back to staff page: [Staff Page](#)

Figure 347 To test user registration (not fluffing password requirements). (Test successful)

7.11.2.3. Test Case: To Test User Registration (Valid user information is given):

Test No:	6
Objective:	To test user registration (valid user information is given).
Action:	ii) Valid user information is given.
Expected result:	iii) The user would be registered successfully(teacher). iv) Success message would be displayed.
Actual result:	iii) The user was registered successfully(teacher). iv) Success message was displayed.
Conclusion:	The test is successful.

Table 46 To test user registration (valid user information is given).

john was registered successfully. The login credentials have been delivered to john.

Scholar

Teacher Registration

First Name	Role
Last Name	teacher
Email	Class
Username	Section
Password	Subjects('ctrl' and 'click' to select multiple subjects)
Password Confirmation	Maths English

Register

Get back to staff page: [Staff Page](#)

This screenshot shows a user interface for teacher registration. At the top, a green banner displays the message "john was registered successfully. The login credentials have been delivered to john.". Below this, the title "Scholar" is displayed above a "Teacher Registration" form. The form contains fields for First Name, Last Name, Email, Username, Password, and Password Confirmation. To the right of these fields are dropdown menus for Role (set to "teacher"), Class (set to "-----"), and Section (set to "-----"). A note below the dropdowns says "Subjects('ctrl' and 'click' to select multiple subjects)". A scrollable list of subjects is shown, with "Maths" and "English" selected. A large blue "Register" button is at the bottom of the form. At the very bottom, there is a link "Get back to staff page: Staff Page".

Figure 348 To test user registration (valid user information is given). (Test successful)

7.11.3. Test Case: To View User List:

*Student View List is tested in this test case.

Test No: 7	
Objective:	To view user list.
Action:	i) Account View from navigation bar is clicked. ii) Student option is clicked.
Expected result:	List of existing students would be shown.
Actual result:	List of existing students was shown.
Conclusion:	The test is successful.

Table 47 To test view user list.

S.no	Username	First Name	Last Name	Email	Delete
1	sayush	Sayush	Khadka	sayushkhadka777@gmail.com	Delete
2	kurt	Kurt	Cobain	kurt@gmail.com	Delete

Figure 349 To test view user list. (Student List)

Additional View List for other users (teacher and staff).

S.no	Username	First Name	Last Name	Email	Delete
1	john	John	Mayer	john@gmail.com	Delete

Figure 350 To test view user list. (Teacher List)

S.no	Username	First Name	Last Name	Email	Position	Delete
1	eddie	Eddie	Vedder	eddie@gmail.com	Administrator	Delete

Figure 351 To test view user list. (Staff List)

7.11.4. Test Case: To Remove User:

*Student is removed in this test case.

Test No: 8	
Objective:	To remove user.
Action:	<ul style="list-style-type: none">i) Account View from navigation bar is clicked.ii) Student option is clicked.iii) Delete option for the student to be deleted is clicked.iv) Ok on confirmation alert box is clicked.
Expected result:	The student would be deleted.
Actual result:	The student was deleted.
Conclusion:	The test is successful.

Table 48 To remove user.

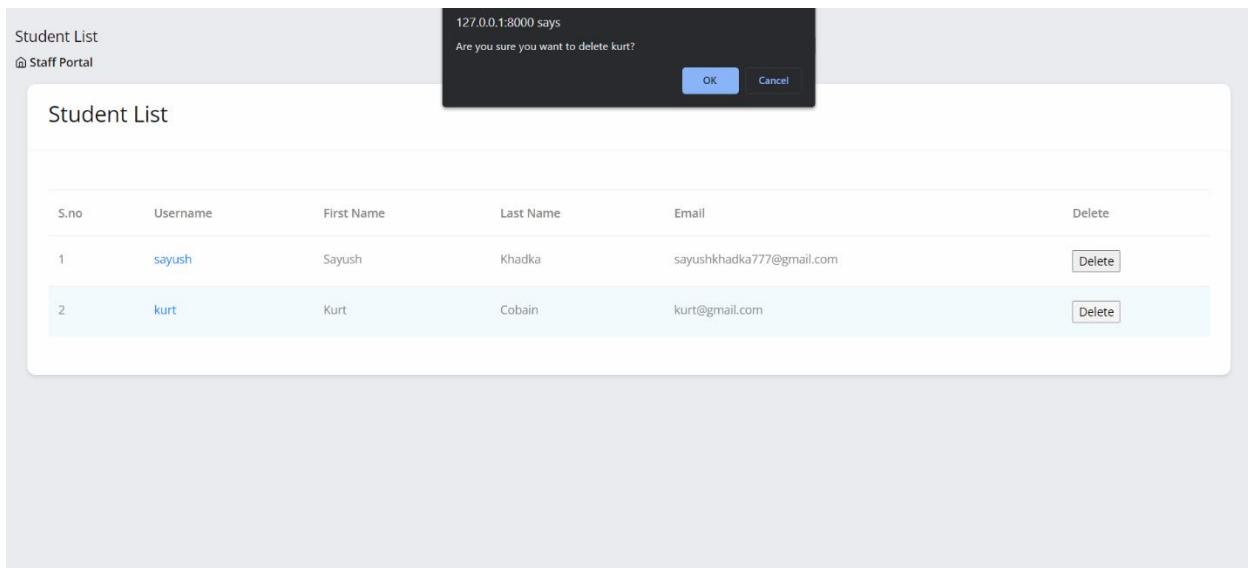


Figure 352 To remove user.

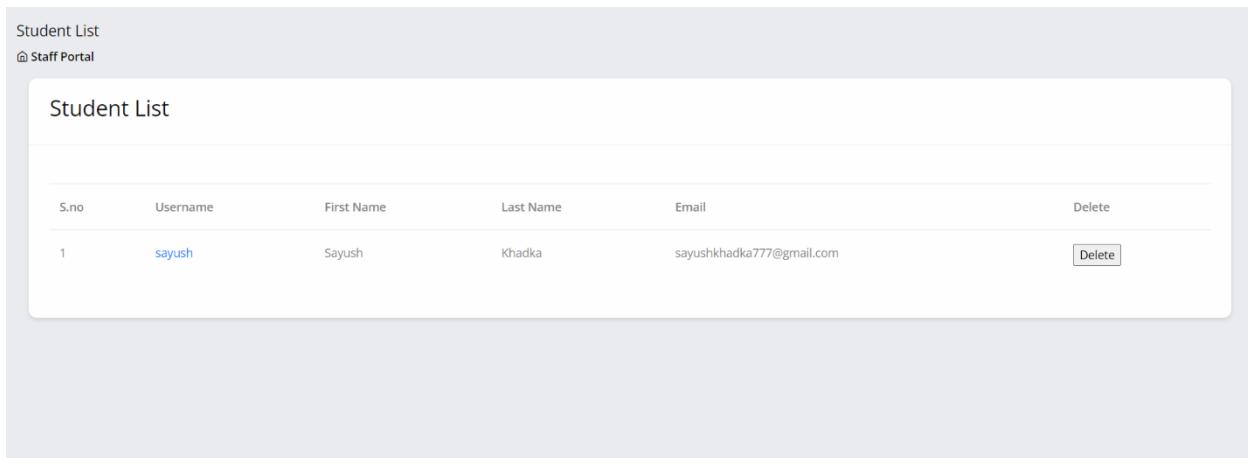


Figure 353 To remove user. (Test Successful)

7.11.5. Exam registration:

7.11.5.1. Test Case: To Register Exam (Negative Number of Question is given):

Test No: 9	
Objective:	To register exam (negative number of questions is given).
Action:	i) Negative number is provided in number of questions field.
Expected result:	Error message “Ensure this value is greater than or equal to 0” would be shown.
Actual result:	Error message was shown.
Conclusion:	The test is successful.

Table 49 To register exam (negative number of questions is given).

Scholar

Exam Registration

Class

9

Section

D

Subject

Maths

First Term

-4

- Ensure this value is greater than or equal to 0.

10

40

Register

Get back to staff page: [Staff Page](#)

The screenshot shows a web-based exam registration form titled 'Exam Registration' under the 'Scholar' section. The form includes fields for Class (9), Section (D), Subject (Maths), and Term (First Term). A question count field contains the value '-4', which is highlighted in red with a validation error message: '• Ensure this value is greater than or equal to 0.' Below the form is a dark blue 'Register' button and a link to the 'Staff Page'.

Figure 354 To register exam (negative number of questions is given). (Test successful)

7.11.5.2. Test Case: To Register Exam (Negative Time is given):

Test No:		10
Objective:	To register exam (negative time is given).	
Action:	i) Negative number is provided in time field.	
Expected result:	Error message “ Ensure this value is greater than or equal to 0 ” would be shown.	
Actual result:	Error message was shown.	
Conclusion:	The test is successful.	

Table 50 To register exam (negative time is given).

Scholar

Exam Registration

Class

9

Section

D

Subject

Maths

First Term

4

-10

- Ensure this value is greater than or equal to 0.

40

Register

Get back to staff page: [Staff Page](#)

Figure 355 To register exam (negative time is given). (Test successful)

7.11.5.3. Test Case: To Register Exam (Negative Score to Pass is given):

Test No: 11	
Objective:	To register exam (negative score to pass is given).
Action:	i) Negative number is provided in score to pass field.
Expected result:	Error message “ Ensure this value is greater than or equal to 0 ” would be shown.
Actual result:	Error message was shown.
Conclusion:	The test is successful.

Table 51 To register exam (negative score to pass is given).

Scholar

Exam Registration

Class

9

Section

D

Subject

Maths

First Term

4

10

-40

- Ensure this value is greater than or equal to 0.

Register

Get back to staff page: [Staff Page](#)

The screenshot shows a web-based application for exam registration. At the top, it says "Scholar" and "Exam Registration". Below that, there are dropdown menus for "Class" (set to "9") and "Section" (set to "D"). Under "Subject", there's a dropdown menu set to "Maths" and a text input field set to "First Term". There are also two empty text input fields, one with "4" and another with "10". Below these is a text input field containing "-40". A red error message "• Ensure this value is greater than or equal to 0." is displayed next to the "-40" field. At the bottom is a large blue "Register" button. At the very bottom of the page, there's a link "Get back to staff page: Staff Page".

Figure 356 To register exam (negative score to pass is given). (Test successful)

7.11.5.4. Test Case: To Register Exam (Valid Exam Information is given):

*Exam can be registered by staff and teacher.

Test No: 12	
Objective:	To register exam (valid exam information is given).
Action:	ii) Valid exam information is given.
Expected result:	iii) The exam would be registered successfully. iv) Success message would be displayed.
Actual result:	iii) The exam was registered successfully. iv) Success message was displayed.
Conclusion:	The test is successful.

Table 52 To register exam (valid exam information is given).

Exam name: 'First Term' was registered successfully. Subject: 'Maths'

Scholar

Exam Registration

Class

Section

Subject

Exam name

No of Questions(*Cannot be changed later*)

Time (In minutes)

Score To Pass

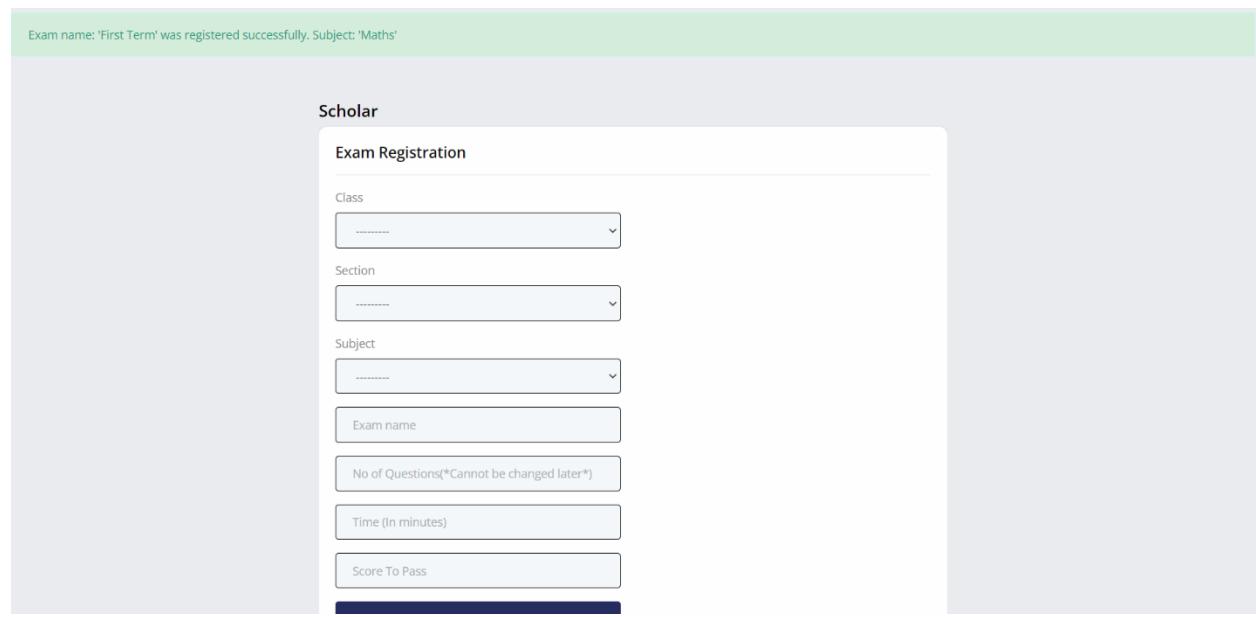


Figure 357 To register exam (valid exam information is given). (Test successful)

7.11.6. Test case: To View exam list:

Test No: 13	
Objective:	To view exam list.
Action:	i) Exam Material from navigation bar is clicked. ii) Exam List is clicked.
Expected result:	List of existing exams would be shown.
Actual result:	List of existing exams was shown.
Conclusion:	The test is successful.

Table 53 To view exam list.

Exam List						
Staff Portal						
S.no	Title	Subject	Level	Level Section	Update	Delete
1	Maths: First Term	Maths	9	D	Update Maths: First Term	Delete
2	English: Class Test 1	English	7	A	Update English: Class Test 1	Delete
3	Maths: Class Test 2	Maths	7	B	Update Maths: Class Test 2	Delete

Figure 358 To view exam list. (Test successful)

7.11.7. Test Case: To View Exam Result List:

Test No: 14	
Objective:	To view exam result list.
Action:	i) Exam Material from navigation bar is clicked. ii) Exam Result List is clicked.
Expected result:	List of existing results of the exams given by the students would be shown.
Actual result:	List of existing results of the exams given by the students was shown.
Conclusion:	The test is successful.

Table 54 To view exam result list.

S.no	Username	First Name	Last Name	Email	Class	Section	Subject	Exam Title	Exam Name	Number of Questions	Date Created	Tim
1	sayush	Sayush	Khadka	sayushkhadka777@gmail.com	9	D	Maths	First Term	Maths: First Term	4	April 16, 2023, 8:57 p.m.	1
2	sayush	Sayush	Khadka	sayushkhadka777@gmail.com	9	D	Maths	First Term	Maths: First Term	4	April 16, 2023, 8:57 p.m.	1
3	sayush	Sayush	Khadka	sayushkhadka777@gmail.com	9	D	Maths	First Term	Maths: First Term	4	April 16, 2023, 8:57 p.m.	1
4	sayush	Sayush	Khadka	sayushkhadka777@gmail.com	9	D	Maths	First Term	Maths: First Term	4	April 16, 2023, 8:57 p.m.	1
5	sayush	Sayush	Khadka	sayushkhadka777@gmail.com	9	D	Maths	First Term	Maths: First Term	4	April 16, 2023, 8:57 p.m.	1
6	sayush	Sayush	Khadka	sayushkhadka777@gmail.com	9	D	Maths	First Term	Maths: First Term	4	April 16, 2023, 8:57 p.m.	1
7	sayush	Sayush	Khadka	sayushkhadka777@gmail.com	9	D	Maths	First Term	Maths: First Term	4	April 16, 2023, 8:57 p.m.	1
8	jimmy	Jimmy	Page	jimmy@gmail.com	9	D	Maths	First Term	Maths: First Term	4	April 16, 2023, 8:57 p.m.	1

Figure 359 To view exam result list. (Test successful)

Email	Class	Section	Subject	Exam Title	Exam Name	Number of Questions	Date Created	Time (In mins)	Score To Pass	Score obtained
sayushkhadka777@gmail.com	9	D	Maths	First Term	Maths: First Term	4	April 16, 2023, 8:57 p.m.	1	40	0.0
sayushkhadka777@gmail.com	9	D	Maths	First Term	Maths: First Term	4	April 16, 2023, 8:57 p.m.	1	40	50.0
sayushkhadka777@gmail.com	9	D	Maths	First Term	Maths: First Term	4	April 16, 2023, 8:57 p.m.	1	40	50.0
sayushkhadka777@gmail.com	9	D	Maths	First Term	Maths: First Term	4	April 16, 2023, 8:57 p.m.	1	40	25.0
sayushkhadka777@gmail.com	9	D	Maths	First Term	Maths: First Term	4	April 16, 2023, 8:57 p.m.	1	40	25.0
sayushkhadka777@gmail.com	9	D	Maths	First Term	Maths: First Term	4	April 16, 2023, 8:57 p.m.	1	40	25.0
sayushkhadka777@gmail.com	9	D	Maths	First Term	Maths: First Term	4	April 16, 2023, 8:57 p.m.	1	40	0.0
jimmy@gmail.com	9	D	Maths	First Term	Maths: First Term	4	April 16, 2023, 8:57 p.m.	1	40	50.0

Figure 360 To view exam result list. (continued) (Test Successful)

7.11.8. Class Registration:

7.11.8.1. Test Case: To test Class Registration (Negative Class is given):

Test No: 15	
Objective:	To test class registration (negative class is given).
Action:	i) Negative number is provided in class field.
Expected result:	Error message “Ensure this value is greater than or equal to 0” would be shown.
Actual result:	Error message was shown.
Conclusion:	The test is successful.

Table 55 To test class registration (negative class is given).

Scholar

Class Registration

• Ensure this value is greater than or equal to 0.

Note: Enter positive integer number only.

Register

Get back to staff page: [Staff Page](#)

Figure 361 To test class registration (negative class is given). (Test successful)

7.11.8.2. Test Case: To Test Class Registration (Valid Class Information is given):

Test No: 16	
Objective:	To test class registration (valid class information is given).
Action:	i) Valid class information is given.
Expected result:	i) The class would be registered successfully. ii) Success message would be displayed.
Actual result:	i) The class was registered successfully. ii) Success message was displayed.
Conclusion:	The test is successful.

Table 56 To test class registration (valid class information is given).

Class 5 was registered successfully.

Scholar

Class Registration

Note: Enter positive integer number only.

Get back to staff page: [Staff Page](#)

A screenshot of a web-based application titled "Scholar". The main heading is "Class Registration". Below it is a text input field with the placeholder "Class". A note below the field says "Note: Enter positive integer number only.". A blue "Register" button is centered below the input field. At the bottom right of the form area, there is a link "Get back to staff page: Staff Page". Above the form, a green horizontal bar displays the message "Class 5 was registered successfully.".

Figure 362 To test class registration (valid class information is given). (Test successful)

7.11.9. Section Registration:

7.11.9.1. Test Case: To Test Section Registration (Class is not selected):

Test No: 17	
Objective:	To test section registration (class is not selected).
Action:	i) Class is not selected
Expected result:	Error message “Please select an item in the list” would be shown.
Actual result:	Error message was shown.
Conclusion:	The test is successful.

Table 57 To test section registration (class is not selected).

Scholar

Section Registration

Class

Section ! Please select an item in the list.

Register

Get back to staff page: [Staff Page](#)

The screenshot shows a 'Section Registration' form within a 'Scholar' interface. The 'Class' dropdown is empty, displaying a placeholder '-----'. A validation message 'Please select an item in the list.' is shown below the dropdown, accompanied by a yellow exclamation mark icon. A large blue 'Register' button is at the bottom of the form. At the bottom right, there is a link 'Get back to staff page: Staff Page'.

Figure 363 To test section registration (class is not selected). (Test successful)

7.11.9.2. Test Case: To Test Section Registration (Valid Section Information is given):

Test No: 18	
Objective:	To test section registration (valid section information is given).
Action:	i) Valid section information is given.
Expected result:	i) The section would be registered successfully. ii) Success message would be displayed.
Actual result:	i) The section was registered successfully. ii) Success message was displayed.
Conclusion:	The test is successful.

Table 58 To test section registration (valid class information is given).

Scholar

Section Registration

Class

5

C

Register

Get back to staff page: [Staff Page](#)

This screenshot shows a section registration form. At the top, it says 'Section Registration'. Below that is a 'Class' dropdown menu containing the value '5'. Underneath the dropdown is a text input field containing the value 'C'. At the bottom of the form is a dark blue 'Register' button. Below the form, there is a link to 'Get back to staff page: Staff Page'.

Figure 364 To test section registration (valid section information is given).

Section C was registered successfully.

Scholar

Section Registration

Class

.....

Section

Register

Get back to staff page: [Staff Page](#)

This screenshot shows the same section registration form as Figure 364, but with a green header bar at the top stating 'Section C was registered successfully.' The form itself is identical, with a 'Class' dropdown set to '.....' and a 'Section' input field set to 'Section'. A dark blue 'Register' button is at the bottom. Below the form is a link to 'Get back to staff page: Staff Page'.

Figure 365 To test section registration (valid class information is given). (Test successful)

7.11.10. Test Case: Subject Registration (Valid subject information is given):

Test No: 19	
Objective:	To test subject registration (valid subject information is given).
Action:	i) Valid subject information is given.
Expected result:	i) The subject would be registered successfully. ii) Success message would be displayed.
Actual result:	i) The subject was registered successfully. ii) Success message was displayed.
Conclusion:	The test is successful.

Table 59 To test subject registration (valid subject information is given).

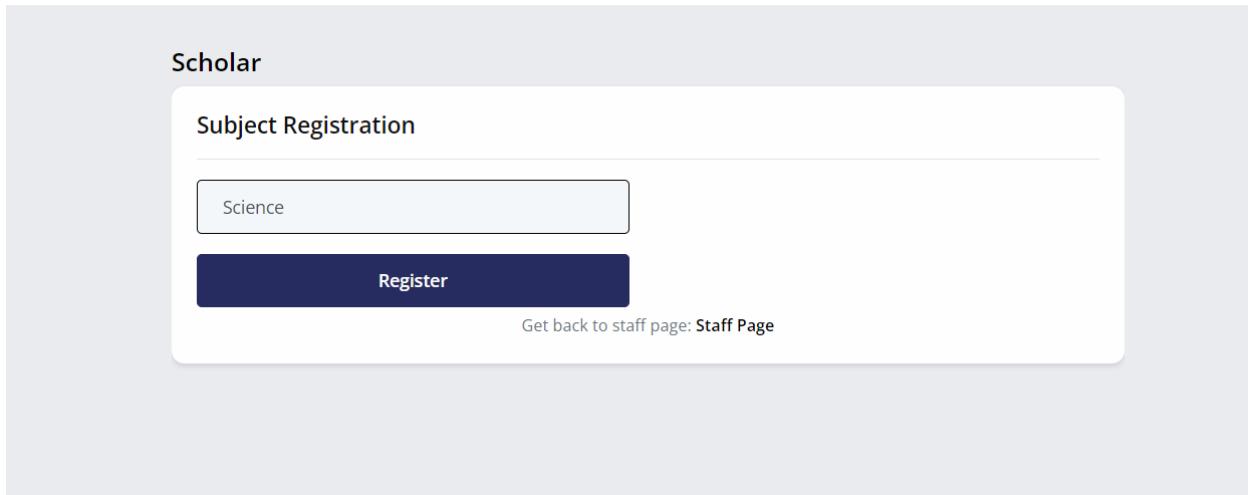


Figure 366 To test subject registration (valid subject information is given).

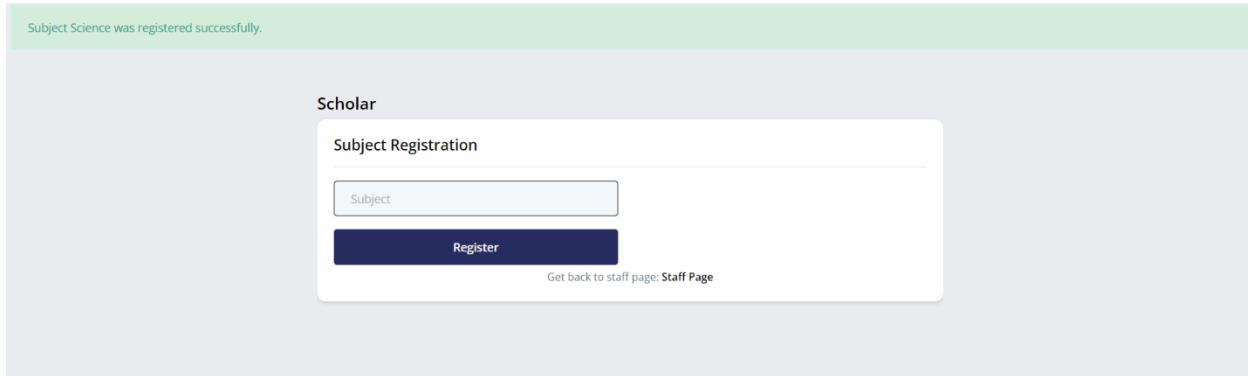


Figure 367 To test subject registration (valid subject information is given). (Test successful)

7.11.11. Test Case: Staff Position Registration (Valid Staff Position Information is given):

Test No: 20	
Objective:	To test staff position registration (valid staff position information is given).
Action:	i) Valid staff position information is given.
Expected result:	i) The staff position would be registered successfully. ii) Success message would be displayed.
Actual result:	i) The staff position was registered successfully. ii) Success message was displayed.
Conclusion:	The test is successful.

Table 60 To test staff position registration (valid staff position information is given).

Scholar

Staff Position Registration

Register

Get back to staff page: [Staff Page](#)

This screenshot shows the 'Staff Position Registration' form. At the top, it says 'Scholar'. Below that is the title 'Staff Position Registration'. There is a text input field containing the value 'Coordinator'. Below the input field is a dark blue button labeled 'Register'. At the bottom right of the form, there is a link 'Get back to staff page: Staff Page'. A green banner at the top of the page displays the message 'Staff position 'Coordinator' was registered successfully.'

Figure 368 To test staff position registration (valid staff position information is given).

Staff position 'Coordinator' was registered successfully.

Scholar

Staff Position Registration

Register

Get back to staff page: [Staff Page](#)

This screenshot shows the 'Staff Position Registration' form. At the top, it says 'Scholar'. Below that is the title 'Staff Position Registration'. There is a text input field containing the value 'Position'. Below the input field is a dark blue button labeled 'Register'. At the bottom right of the form, there is a link 'Get back to staff page: Staff Page'. A green banner at the top of the page displays the message 'Staff position 'Coordinator' was registered successfully.'

Figure 369 To test staff position registration (valid staff position information is given). (Test successful)

7.11.12. Test Case: To Post Notice (Valid Notice Information is given):

*Staff and teacher can post notice.

Test No: 21	
Objective:	To test post notice (valid notice information is given).
Action:	i) Valid notice information is given.
Expected result:	i) The notice would be posted successfully. ii) Notice would be displayed on notice board.
Actual result:	i) The notice was posted successfully. ii) Notice was displayed on notice board.
Conclusion:	The test is successful.

Table 61 To test post notice (valid notice information is given).

Notices

Add Notice

Eddie
• This field is required.

eddie@gmail.com
• This field is required.

FYP
• This field is required.

FYP submission on Wednesday.
• This field is required.

Post

Go back to staff notice board? [Staff notice board](#)

This screenshot shows a 'Add Notice' form. It has four input fields: 'Eddie' (name), 'eddie@gmail.com' (email), 'FYP' (content), and 'FYP submission on Wednesday.' (description). Each field has a red error message below it stating '• This field is required.' A large blue 'Post' button is at the bottom. Below the form is a link to go back to the 'Staff notice board'.

Figure 370 To test post notice (valid notice information is given).

Staff Notice Board

Return back to: [Staff Main Page](#)

Total number of notice: 1

[Add more](#)

FYP

FYP submission on Wednesday.

By: Eddie
email- eddie@gmail.com

Views from other users

[Add more](#)

This screenshot shows the 'Staff Notice Board'. It displays one notice titled 'FYP' with the content 'FYP submission on Wednesday.'. Below the notice, it shows it was posted by 'Eddie' with the email 'eddie@gmail.com'. There is a link to 'Add more' notices and another link to 'Add more' views.

Figure 371 To test post notice (valid notice information is given). (Test successful)

7.11.13. Test Case: To view notice:

*Student, teacher, and staff can all view notice from their respective portal.

Test No: 22	
Objective:	To view notice.
Action:	i) View Notice from navigation bar is clicked.
Expected result:	List of existing notices would be shown.
Actual result:	List of existing notices was shown.
Conclusion:	The test is successful.

Table 62 To view notice.

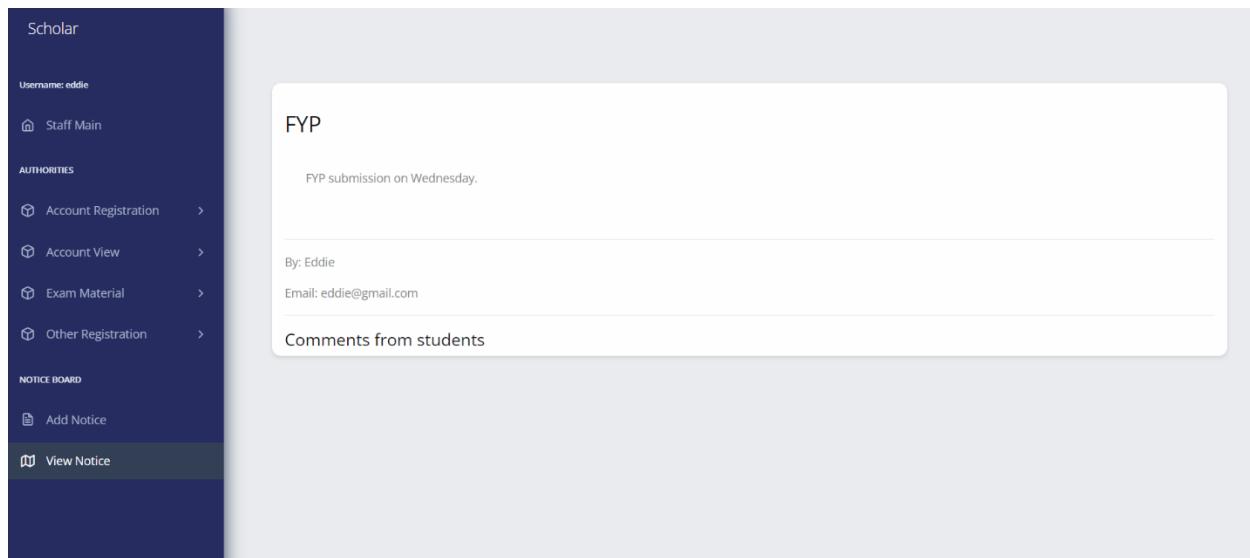


Figure 372 To view notice. (Test successful) (Staff Notice View)

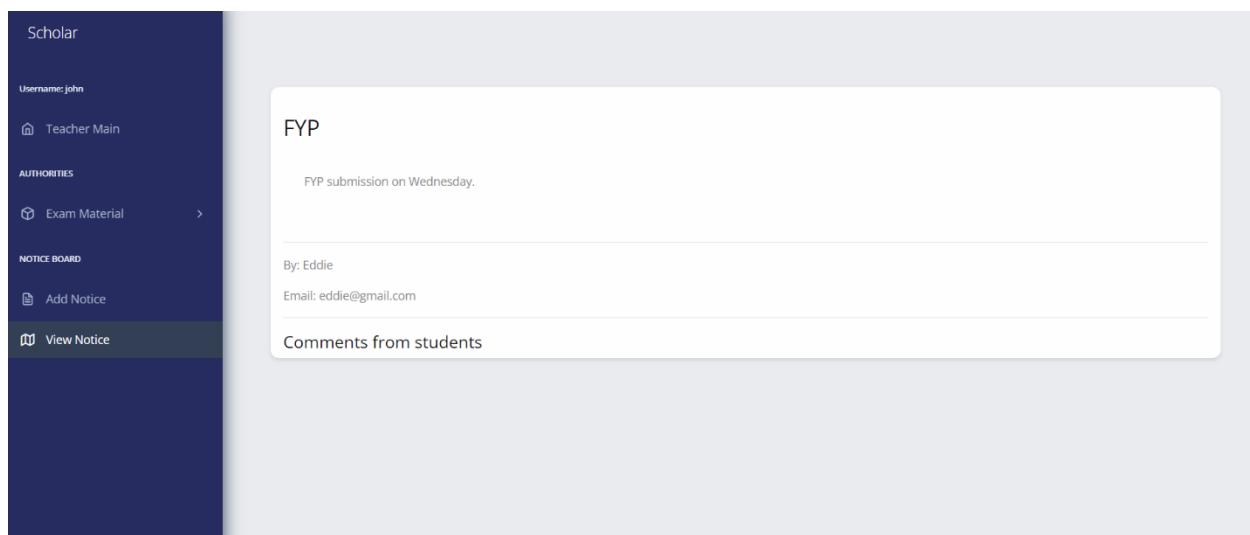


Figure 373 To view notice. (Test successful) (Teacher Notice View)

The screenshot shows the Scholar application interface. On the left is a dark sidebar with the following menu items:

- Scholar
- Username: sayush
- Student Main
- AUTHORITIES
- Available exams
- NOTICE BOARD
- View Notice** (highlighted in green)

The main content area displays a notice titled "FYP" with the following details:

FYP submission on Wednesday.

By: Eddie
Email: eddie@gmail.com

Comments from students

[Add more](#)

Figure 374 To view notice. (Test successful) (Student Notice View)

7.11.14. To Create Question:

7.11.14.1. Test Case: Test to Add Multiple Answers:

Test No:	23
Objective:	To add multiple answers.
Action:	i) Add answers button is clicked.
Expected result:	i) New answer field would be displayed.
Actual result:	i) New answer field was displayed.
Conclusion:	The test is successful.

Table 63 To add multiple answers.

Maths: First Term

Type of Question * **Multiple Choice**

Title *
5-4?

Add Answers

Answer *	Correct(Write true for the correct answer) *	Delete?
3	False	<input type="checkbox"/>

Add answers

Submit

Figure 375 To add multiple answers

Type of Question * **Multiple Choice**

Title *
5-4?

Add Answers

Answer *	Correct(Write true for the correct answer) *	Delete?
3	False	<input type="checkbox"/>
	False	<input type="checkbox"/>

Add answers

Submit

Figure 376 To add multiple answers (Test successful)

7.11.14.2. Test Case: Test to Create Question:

Test No: 24	
Objective:	To create question.
Action:	i) Question name is given. ii) Answer to the question is provided.
Expected result:	i) The question would be created successfully.
Actual result:	i) The question was created successfully.
Conclusion:	The test is successful.

Table 64 To create question.

Add Questions

[Return to your portal](#)

Exam name *

Maths: First Term

Type of Question * **Multiple Choice**

Title *

1+1?

Add Answers

Answer *	Correct(Write true for the correct answer) *	Delete?
3	False	<input type="checkbox"/>

Figure 377 To create question.

Title *

1+1?

Add Answers

Answer *	Correct(Write true for the correct answer) *	Delete?
3	False	<input type="checkbox"/>
2	true	<input type="checkbox"/>
4	False	<input type="checkbox"/>
5	False	<input type="checkbox"/>

Add answers

Submit

Figure 378 To create question. (Fill answers)

Question List					
S.no	Question	Exam Name	Subject	Update	Delete
1	1+1?	Maths: First Term	Maths	Update 1+1?	Delete

Figure 379 To create question. (Test successful)

380

Sayush Khadka

7.11.15. Test Case: To View Question List:

Test No: 25	
Objective:	To view question list.
Action:	i) Exam Material from navigation bar is clicked. ii) Question List is clicked.
Expected result:	List of existing questions would be shown.
Actual result:	List of existing questions was shown.
Conclusion:	The test is successful.

Table 65 To view question list.

S.no	Question	Exam Name	Subject	Update	Delete
1	1+1?	Maths: First Term	Maths	Update 1+1?	<button>Delete</button>
2	2+2?	Maths: First Term	Maths	Update 2+2?	<button>Delete</button>
3	2*2?	Maths: First Term	Maths	Update 2*2?	<button>Delete</button>

Figure 380 To view question list. (Test successful)

7.11.16. Attend Examination:

7.11.16.1. Test Case: To View Available Exams:

Test No: 26	
Objective:	To view available exams.
Action:	i) Available exams from navigation bar are clicked.
Expected result:	List of available exams would be shown.
Actual result:	List of available exams was shown.
Conclusion:	The test is successful.

Table 66 To view available exams.



Figure 381 To view available exams. (Test successful)

7.11.16.2. Test Case: Click on The Exam Card to attend that exam:

Test No: 27	
Objective:	To attend the particular exam.
Action:	i) The exam card of that particular exam is clicked. ii) Yes, on pop-up message to begin exam is clicked.
Expected result:	i) A pop-up message would show to begin the exam. ii) The questions would be displayed for the particular exam.
Actual result:	i) A pop-up message was shown to begin the exam. ii) The questions were displayed for the particular exam.
Conclusion:	The test is successful.

Table 67 To attend the particular exam.

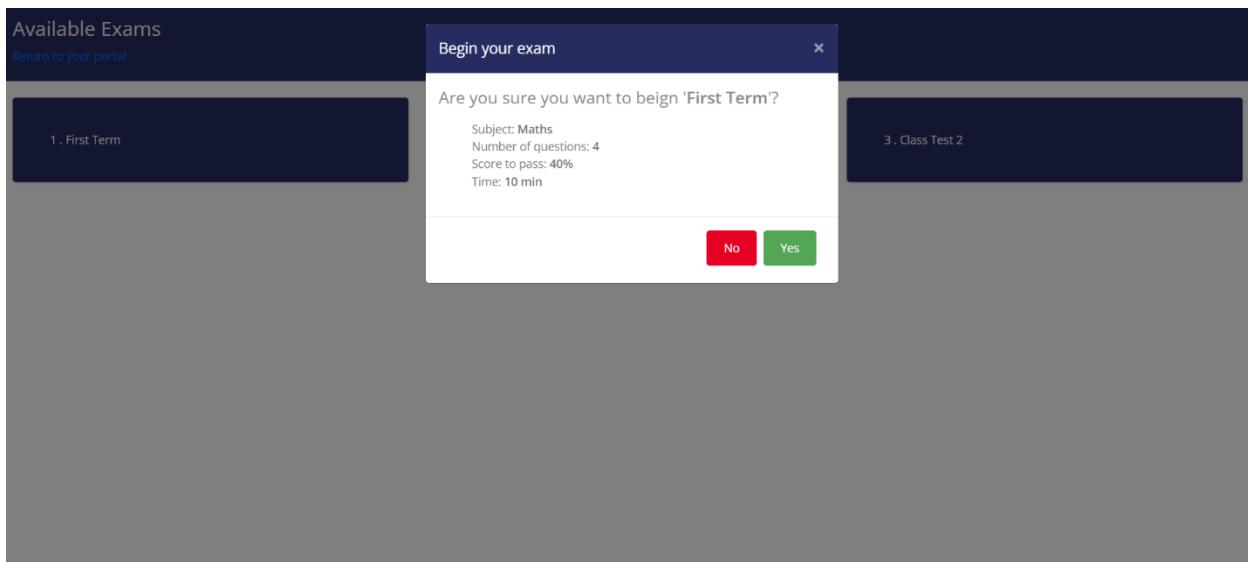


Figure 382 To attend the particular exam. (Test successful)

A screenshot of a math exam interface. At the top, it shows exam details: Subject: Maths, Exam title: First Term, Number of Questions: 4, Score to pass: 40%, and a Timer: 09:58. Below this is a "Return to available exams" link. The exam consists of three questions: 1. $1+1?$ with options 2, 4, 5, and 3; 2. $2+2?$ with options 5, 7, 4, and 3; and 3. $2^2?$ with options 5, 4, 3, and 1.

Figure 383 To attend the particular exam. (Test successful)

5
 3

2+2?
 5
 7
 4
 3

2*2?
 5
 4
 3
 1

5-4?
 5
 1
 2
 4

Submit your paper

Figure 384 To attend the particular exam. (continued) (Test successful)

7.11.16.3. Test Case: To Test the working of the Countdown Timer (Begin Timer):

Test No:	28
Objective:	To test the working of the countdown timer (begin timer).
Action:	i) The exam card of that particular exam is clicked. ii) Yes, on pop-up message to begin exam is clicked.
Expected result:	i) Timer would begin.
Actual result:	i) Timer begun.
Conclusion:	The test is successful.

Table 68 To test the working of the countdown timer (begin timer).

Timer: 09:58
Subject: Maths
Exam title: First Term
Number of Questions: 4
Score to pass: 40%
[Return to available exams](#)

1+1?

2
 4
 5
 3

2+2?

5
 7
 4
 3

2*2?

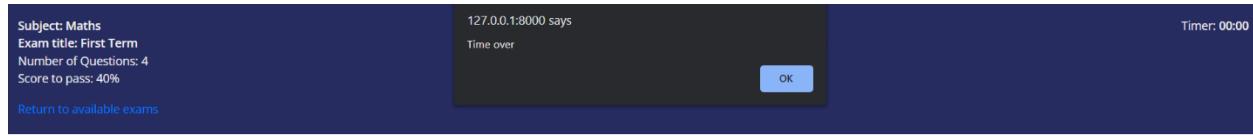
5
 4
 3
 1

Figure 385 To test the working of the countdown timer (begin timer). (Test successful)

7.11.16.4. Test Case: To Test the working of the Countdown Timer (End Timer):

Test No: 29	
Objective:	To test the working of the countdown timer (end timer).
Action:	<ul style="list-style-type: none">i) The exam card of that particular exam is clicked.ii) Yes, on pop-up message to begin exam is clicked.iii) Answers are clicked.
Expected result:	<ul style="list-style-type: none">i) Timer would stop after the time provided during the registration of the examination.ii) Result would be displayed according to the answered questions.
Actual result:	<ul style="list-style-type: none">i) Timer stopped after the time provided during the registration of the examination.ii) Result was displayed according to the answered questions.
Conclusion:	The test is successful.

Table 69 To test the working of the countdown timer (end timer).



1+1?

- 2
- 4
- 5
- 3

2+2?

- 5
- 7
- 4
- 3

2*2?

- 5
- 4
- 3
- 1

Figure 386 To test the working of the countdown timer (end timer). (Pop-up message) (Test successful)



Figure 387 To test the working of the countdown timer (end timer). (Result is displayed) (Test successful)

7.11.16.5. Test Case: To Submit the Exam Paper:

Test No: 30	
Objective:	To submit exam paper.
Action:	<ul style="list-style-type: none">i) The exam card of that particular exam is clicked.ii) Yes, on pop-up message to begin exam is clicked.iii) Answers are clicked.iv) Click on Submit your paper button.
Expected result:	<ul style="list-style-type: none">i) The paper would be submitted.
Actual result:	<ul style="list-style-type: none">i) The paper was submitted.
Conclusion:	The test is successful.

Table 70 To submit exam paper.

Subject: Maths
Exam title: First Term
Number of Questions: 4
Score to pass: 40%
[Return to available exams](#)

Timer: 00:46

1+1?

- 2
- 4
- 5
- 3

2+2?

- 5
- 7
- 4
- 3

2*2?

- 5
- 4
- 3
- 1

Figure 388 To submit exam paper. (Test successful)

-
- 5
 - 3

2+2?

- 5
- 7
- 4
- 3

2*2?

- 5
- 4
- 3
- 1

5-4?

- 5
- 1
- 2
- 4

[Submit your paper](#)

Figure 389 To submit exam paper. (continued) (Test successful)

Subject: Maths
Exam title: First Term
Number of Questions: 4
Score to pass: 40%
[Return to available exams](#)

Timer: 00:08

Failed. Your result is 25.00.

1+1? Correct Answer: 2 Answered: 3
2+2? Correct Answer: 4 Answered: 3
2*2? Correct Answer: 4 Answered: 1
5-4? Answered: 1

Figure 390 To submit exam paper. (Test successful)

7.11.16.6. Test Case: To view results (For Student):

Test No: 31	
Objective:	To view results (for student).
Action:	<ul style="list-style-type: none">i) The exam card of that particular exam is clicked.ii) Yes, on pop-up message to begin exam is clicked.iii) Answers are clicked.iv) Click on Submit your paper button.
Expected result:	<ul style="list-style-type: none">i) The paper would be submitted.ii) The result would be displayed.
Actual result:	<ul style="list-style-type: none">i) The paper was submitted.ii) The result was displayed.
Conclusion:	The test is successful.

Table 71 To view results (for student).

Subject: Maths
Exam title: First Term
Number of Questions: 4
Score to pass: 40%

[Return to available exams](#)

Timer: 00:08

Failed. Your result is 25.00.

1+1? | Correct Answer: 2 | Answered: 3

2+2? | Correct Answer: 4 | Answered: 3

2*2? | Correct Answer: 4 | Answered: 1

5-4? | Answered: 1

Figure 391 To view results (for student). (Test successful)

7.11.17. To Change Password:

*Test actor: Student

7.11.17.1. Test Case: To Change Password (incorrect old password is entered):

Test No: 32	
Objective:	To change password.
Action:	<ul style="list-style-type: none">i) Student clicks on the Change Password option from settings.ii) Then the password change form is displayed.iii) Student enters details to change password (incorrect old password, new password and confirm password).
Expected result:	<ul style="list-style-type: none">i) System would show error message.
Actual result:	<ul style="list-style-type: none">i) System showed error message.
Conclusion:	The test is successful.

Table 72 To Change Password (incorrect old password is entered)

Password Change

Password Change Form

• Your old password was entered incorrectly. Please enter it again.

Old password:

New 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.

New password confirmation:

Change Password

Figure 392 To Change Password (incorrect old password is entered) (Test successful)

7.11.17.2. Test Case: To Change Password (new password confirmation doesn't match the new password):

Test No:	33
Objective:	To change password.
Action:	iv) Student clicks on the Change Password option from settings. v) Then the password change form is displayed. vi) Student enters details to change password (old password, new password and different confirm password).
Expected result:	i) System would show error message
Actual result:	i) System showed error message.
Conclusion:	The test is successful.

Table 73 To Change Password (new password confirmation doesn't match the new password)

Password Change

Password Change Form

Old password:

New 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.

• **The two password fields didn't match.**

New password confirmation:

Change Password

Figure 393 To Change Password (new password confirmation doesn't match the new password) (Test successful)

7.11.17.3. Test Case: To Change Password (If the entered details are valid):

Test No: 34	
Objective:	To change password.
Action:	<ul style="list-style-type: none">i) Student clicks on the Change Password option from settings.ii) Then the password change form is displayed.iii) Student enters details to change password (old password, new password and confirm password).
Expected result:	<ul style="list-style-type: none">i) System would redirect to change password successful confirmation page.
Actual result:	<ul style="list-style-type: none">i) System redirected to change password successful confirmation page.
Conclusion:	The test is successful.

Table 74 To Change Password (If the entered details are valid):

Your password has been changed. You may go ahead and [Log In](#) from our home page now.

Scholar

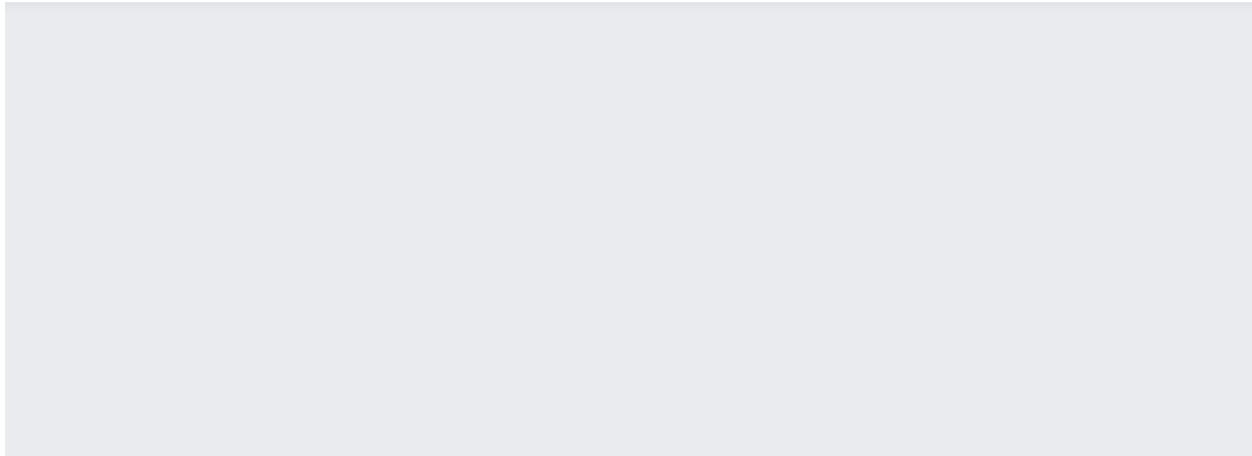


Figure 394 To Change Password (If the entered details are valid):