

Automated CQI



Session: 2020 – 2024

Submitted by:

Fatima Zafar 2020-CS-664

Abdul Moiz Arif 2020-CS-674

Munazza Rafique 2020-CS-687

Supervised by:

Dr. Zeeshan Ramzan

Department of Computer Science, New Campus
University of Engineering and Technology
Lahore, Pakistan

Automated CQI

(Session 2020 Computer Science)

The thesis is to be submitted to the Department of Computer Science, New Campus, University of Engineering and Technology, Lahore for the partial fulfillment of the requirement for Bachelor of Science in Computer Science.

Supervisor

Signature:

Name: **Dr. Zeeshan Ramzan**

Designation: **Lecturer**

Chairman

Signature:

Dean

Signature:

Department of Computer Science, New Campus

**University of Engineering and Technology
Lahore, Pakistan**

Declaration

I declare that the work contained in this thesis is my own, except where explicitly stated otherwise. In addition this work has not been submitted to obtain another degree or professional qualification.

Members Signature:

Date: _____

Acknowledgments

In the name of Allah, the most beneficent and the most merciful. Alhamdulillah, we express our gratitude to Allah for His strength and blessings, which enabled us to successfully complete this project. We would like to extend special appreciation to our supervisor, Dr. Zeeshan Ramzan, for his invaluable guidance and unwavering support throughout the project. His assistance was instrumental in facilitating our work. Additionally, we are grateful to all our project members who contributed directly to the research and development efforts. Thank you very much to everyone involved in making this project a reality.

Dedicated to our families and respected teachers . . .

Contents

Acknowledgments	iii
List of Figures	xii
List of Tables	xiv
Abbreviations	xvii
Abstract	xviii
1 Introduction	1
1.1 Introduction	1
1.2 Objectives	2
1.3 Problem Statement	3
1.4 Assumptions and Constraints	3
1.4.1 Assumptions	3
1.4.2 Constraints	4
1.5 Project Scope and Features	4
1.5.1 Project Scope	4
1.5.2 Features	5
2 Background Study	6
2.1 Literature Review	6
2.1.1 Anthology	6
2.1.2 Google Forms	7
2.1.3 Continuous Quality Improvement in Higher Education	8
2.1.4 Effect of OBE systems	8
2.1.5 Swedish and American Model for CQI	9
2.1.6 Role of Automation in Improvement of Quality	10
3 Requirement Analysis	14
3.1 Requirement Elicitation	14
3.1.1 Functional Requirement	14
3.1.1.1 Security Requirements	14
3.1.1.2 User Requirements	15
3.1.1.3 Admin Requirements	15

3.1.2	Non-functional Requirement	16
3.1.2.1	Performance	16
3.1.2.2	Scalability	16
3.1.2.3	Availability	16
3.1.2.4	Capacity	17
3.1.2.5	Reliability	17
3.1.2.6	Maintainability	17
3.1.2.7	Serviceability	18
3.1.2.8	Interoperability	18
3.1.2.9	Security	18
3.1.2.10	Regulatory	18
3.1.2.11	Manageability	19
3.1.2.12	Data Integrity	19
3.1.2.13	Usability	19
3.1.3	Requirement Traceability Matrix	19
3.2	Use-case Description	20
3.2.1	Login	21
3.2.2	Create Users	22
3.2.3	Create Admins	23
3.2.4	Store Teachers Information	24
3.2.5	Select Form	25
3.2.6	Fill Form	26
3.2.7	Submit Form	27
3.2.8	Add New Form	28
3.2.9	Delete Form	29
3.2.10	Edit Form	30
3.2.11	Manage Profile	31
3.2.12	Edit Username	32
3.2.13	Edit Password	33
3.2.14	View Stats	34
3.2.15	View Results	35
3.2.16	Download Results	36
3.2.17	Forgot Password	37
3.3	Use-Case Diagram	38
4	System Design	39
4.1	Work BreakDown Structure	39
4.2	Gantt chart	40
4.3	Wireframes	41
4.3.1	Log in	41
4.3.2	Fill Form	41
4.3.3	Manage Profile	42
4.3.4	Manage Form	42
4.3.5	Download Results	43

4.3.6	Create new Form	43
4.3.7	Create new Subject Registration Form	44
4.3.8	Add User/Admin/Teacher	44
4.3.9	View Stats	45
4.4	Activity Diagram	46
4.4.1	Admin-side Diagram	46
4.4.2	User-side Diagram	47
4.5	State Diagram	48
4.5.1	Admin-side Diagram	48
4.5.2	User-side Diagram	49
4.6	Sequence Diagram	50
4.6.1	Login	50
4.6.2	Fill Form	51
4.6.3	Add Admins	52
4.6.4	Add Teachers	53
4.6.5	Add Students	54
4.6.6	Create Form	55
4.6.7	Delete Form	56
4.6.8	Edit Form	57
4.6.9	Manage Profile	58
4.6.10	Edit Username	59
4.6.11	Edit Password	60
4.6.12	View Stats	61
4.6.13	Download result	62
4.7	Collaboration Diagram	63
4.7.1	User-side Diagram	63
4.7.2	Admin-side Diagram	64
4.8	Class Diagram	65
4.9	Component Diagram	66
4.10	Deployment Diagram	67
4.11	ER Diagram	67
4.12	Software Architecture	68
5	System Implementation	69
5.1	System Implementation	69
5.1.1	Software Requirements	69
5.1.2	Hardware Requirements	70
5.1.3	Sign-in	70
5.1.4	Create Form and Survey	72
5.1.5	Edit Form	72
5.1.6	Delete Form	73
5.1.7	Add User (Backend Implementation)	73
5.1.8	Add Admin (Backend Implementation)	74
5.1.9	Add Teacher (Backend Implementation)	74

5.1.10	Add User, Admin, Teacher manually Front-end	75
5.1.11	Add User, Admin, Teacher through .csv Front-end	76
5.1.12	View and download results	77
5.1.13	Send Reminders	78
5.1.14	Edit Profile	78
5.1.15	Change Mode	79
6	System Testing and Evaluation	80
6.1	Unit Testing	81
6.1.1	Log-in page Testing	81
6.1.1.1	Email Text Field	81
6.1.1.2	Password Text field	82
6.1.1.3	Log-in Button	82
6.1.2	Top Nav-bar Testing	82
6.1.3	Side Nav-bar Testing	82
6.1.3.1	Back Button	83
6.1.3.2	Profile Display	83
6.1.3.3	Log-out Button	83
6.1.3.4	Dark Mode Toggle	83
6.1.3.5	Dashboard Option	84
6.1.3.6	Surveys Option	84
6.1.3.7	Users Option	84
6.1.3.8	Teachers Option	84
6.1.3.9	Admins Option	85
6.1.3.10	Reminders Option	85
6.1.3.11	Results Option	85
6.1.3.12	Profile Option	85
6.1.4	Admin Dashboard Testing	85
6.1.4.1	Display Survey graphs	86
6.1.4.2	Display Subject Registration graphs	86
6.1.5	Admin page Testing	86
6.1.5.1	Add Admin List through .csv	86
6.1.5.2	Add Admins manually	86
6.1.5.3	Remove Admins from list	87
6.1.6	Student page Testing	87
6.1.6.1	Add Student List through .csv	87
6.1.6.2	Add Students manually	87
6.1.6.3	Remove Students from list	87
6.1.7	Teacher page Testing	88
6.1.7.1	Add Teacher List through .csv	88
6.1.7.2	Add Teachers manually	88
6.1.7.3	Remove Teachers from list	88
6.1.8	Reminder Testing	88
6.1.9	Results Testing	89

6.1.10	Profile Testing	90
6.1.11	Fill Form Testing	91
6.2	Integration Testing	92
6.2.1	Sign-in	92
6.2.2	View Stats	92
6.2.3	Create Student Forms	92
6.2.4	Create Final year Student Forms	93
6.2.5	Create Alumina Forms	93
6.2.6	Create Organization Forms	93
6.2.7	Create Registration Form	94
6.2.8	View and Download Results	94
6.2.9	Add user	94
6.2.10	Add Teacher	94
6.2.11	Add Admin	95
6.2.12	Fill Form	95
6.2.13	Send Reminders	95
6.2.14	Edit Profile	96
6.2.15	Change Mode	96
6.3	System Testing	97
6.3.1	Sign-in Testing	97
6.3.2	Create and Add User Testing	98
6.3.3	Create and Add Teacher Testing	99
6.3.4	Create and Add Admin Testing	100
6.3.5	Create Form Testing	101
6.3.6	View and Download Results Testing	102
6.3.7	Select Category and Send Reminders Testing	103
6.3.8	View and Edit Profile Testing	104
6.3.9	Fill Form Testing	105
6.4	Acceptance Testing	106
6.4.1	Sign-in Testing	106
6.4.2	Create User Testing	107
6.4.3	Create Teacher Testing	108
6.4.4	Create Admin Testing	109
6.4.5	Create Student Form Testing	110
6.4.6	Create Final Year Student Form Testing	111
6.4.7	Create Alumina Form Testing	112
6.4.8	Create Organization Form Testing	113
6.4.9	Download Results Testing	114
6.4.10	Send Reminders Testing	115
6.4.11	Edit Profile Testing	116
6.4.12	Change Mode	117
6.5	Performance Testing	117
6.5.1	Load testing	118
6.5.2	Stress testing	118

6.5.3	Endurance testing	119
6.5.4	Spike testing	119
6.6	Security Testing	120
6.6.1	Integrity	120
6.6.2	Confidentiality	121
6.6.3	Authentication	121
6.6.4	Authorization	121
6.6.5	Availability	121
6.7	Usability Testing	122
6.8	Compatibility Testing	123
7	Results and Outputs	124
7.0.1	Sign-in	124
7.0.2	Admin Dashboard	125
7.0.3	Created Forms	125
7.0.4	Create new Survey	126
7.0.5	Create Subject Registration Form	126
7.0.6	Filling Survey Form	127
7.0.7	Filling Subject Registration Form	127
7.0.8	Filling Subject Registration Form	128
7.0.9	Add User through csv	128
7.0.10	Add User manually	129
7.0.11	Add Admin manually/through csv	129
7.0.12	Add Teacher manually/through csv	130
7.0.13	Send Reminders	130
7.0.14	User Added Email	131
7.0.15	Fill Form Email	131
7.0.16	View/Edit Profile	132
7.0.17	Change Username Email	132
7.0.18	Change Password Email	133
7.0.19	View and Download Results	133
7.0.20	Registered Subjects	134
7.0.21	Change Mode	134
8	Conclusions and Future Direction	135
8.1	Conclusion	135
8.2	Future Direction	135
8.2.1	Enhanced Data Analytics	136
8.2.2	Integration with Learning Management Systems	136
8.2.3	Feedback Mechanism Refinement	136
8.2.4	Scalability and Expansion	136

References	137
-------------------	------------

List of Figures

2.1	Anthology Designed form for University of Mississippi	7
2.2	Google forms Templates	7
2.3	The proposed linkage of CitationSpady's (1994) top to bottom levels of outcomes development with the incorporation of long-term PEOs.	9
3.1	UseCase Diagram	38
4.1	Work BreakDown Structure	39
4.2	Gantt Chart	40
4.3	Login Wireframe	41
4.4	Fill Form Wireframe	41
4.5	Manage Profile Wireframe	42
4.6	Manage Form Wireframe	42
4.7	Download Results Wireframe	43
4.8	Create new Form Wireframe	43
4.9	Create new Registration Form Wireframe	44
4.10	Add User/Admin/Teacher	44
4.11	View Stats Wireframe	45
4.12	Admin-side Diagram	46
4.13	User-side Diagram	47
4.14	Admin-side State Diagram	48
4.15	User-side State Diagram	49
4.16	Login Sequence Diagram	50
4.17	Fill Form Sequence Diagram	51
4.18	Add Admins Sequence Diagram	52
4.19	Add Teachers Sequence Diagram	53
4.20	Add Students Sequence Diagram	54
4.21	Create Form Sequence Diagram	55
4.22	Delete Form Sequence Diagram	56
4.23	Edit Form Sequence Diagram	57
4.24	Manage Profile Sequence Diagram	58
4.25	Edit Username Sequence Diagram	59
4.26	Edit Password Sequence Diagram	60
4.27	View Stats Sequence Diagram	61
4.28	Download result Sequence Diagram	62
4.29	User-side Collaboration Diagram	63

4.30 Admin-side Collaboration Diagram	64
4.31 Class Diagram	65
4.32 Component Diagram	66
4.33 Deployment Diagram	67
4.34 ER Diagram	67
4.35 ER Diagram	68
5.1 Code Snippet for Token Generation	70
5.2 Code Snippet for Token Returning	71
5.3 Code Snippet for Token Checking	71
5.4 Code Snippet for Add Form	72
5.5 Code Snippet for Edit Form	72
5.6 Code Snippet for Delete Form	73
5.7 Code Snippet for Add User	73
5.8 Code Snippet for Add Admin	74
5.9 Code Snippet for Add Teacher	74
5.10 Code Snippet for Add User,Teacher,Admin manually	75
5.11 Code Snippet for Add User,Teacher,Admin csv	76
5.12 Code Snippet for View and Download Results	77
5.13 Code Snippet for Send Reminders	78
5.14 Code Snippet for Edit Profile	78
5.15 Code Snippet for Change Mode	79
7.1 Sign-in	124
7.2 Admin Dashboard	125
7.3 Created Forms	125
7.4 Create new Survey	126
7.5 Create Subject Registration Form	126
7.6 Filling Survey Form	127
7.7 Filling Subject Registration Form	127
7.8 Filling Subject Registration Form	128
7.9 Add User through csv	128
7.10 Add User manually	129
7.11 Add Admin manually/through csv	129
7.12 Add Teacher manually/through csv	130
7.13 Send Reminders	130
7.14 User Added Email	131
7.15 Fill Form Email	131
7.16 View/Edit Profile	132
7.17 Change Username Email	132
7.18 Change Password Email	133
7.19 View/Download Results	133
7.20 Registered Subjects	134
7.21 Change Mode	134

List of Tables

2.1	Related Systems Analysis	11
3.1	Security Requirements (FR-01-00)	14
3.2	User Requirements (FR-02-00)	15
3.3	Admin Requirements (FR-03-00)	15
3.4	Performance (NFR-01-00)	16
3.5	Scalability (NFR-02-00)	16
3.6	Availability (NFR-03-00)	16
3.7	Capacity (NFR-04-00)	17
3.8	Reliability (NFR-05-00)	17
3.9	Maintainability (NFR-06-00)	17
3.10	Serviceability (NFR-07-00)	18
3.11	Interoperability (NFR-08-00)	18
3.12	Security (NFR-09-00)	18
3.13	Regulatory (NFR-10-00)	18
3.14	Manageability (NFR-11-00)	19
3.15	Data Integrity (NFR-12-00)	19
3.16	Usability (NFR-13-00)	19
3.17	Requirement Traceability Matrix	20
3.18	User Login	21
3.19	Add User	22
3.20	Add Admin	23
3.21	Store Teachers Information	24
3.22	Select Form	25
3.23	Fill Form	26
3.24	Submit Form	27
3.25	Add New Form	28
3.26	Delete Form	29
3.27	Edit Form	30
3.28	Manage Profile	31
3.29	Edit Username	32
3.30	Edit Password	33
3.31	View Stats	34
3.32	View Results	35
3.33	Download Results	36

3.34 Forgot Password	37
5.1 Software Requirements	69
5.2 Hardware Requirements	70
6.1 Email Text Field Testing	81
6.2 Password Text Field Testing	82
6.3 Log-in Button Testing	82
6.4 Back Button Testing	83
6.5 Profile Display Testing	83
6.6 Log-out Button Testing	83
6.7 Dark Mode Toggle Button Testing	83
6.8 Dashboard Button Testing	84
6.9 Survey Button Testing	84
6.10 Students Button Testing	84
6.11 Teachers Button Testing	84
6.12 Admin Button Testing	85
6.13 Reminder Button Testing	85
6.14 Results Button Testing	85
6.15 Profile Button Testing	85
6.16 Survey Graph Testing	86
6.17 Subject Registration Graph Testing	86
6.18 Add Admin List through .csv Testing	86
6.19 Add Admins manually Testing	86
6.20 Remove Admin Testing	87
6.21 Add Student List through .csv Testing	87
6.22 Add Admins manually Testing	87
6.23 Remove Student Testing	87
6.24 Add Teacher List through .csv Testing	88
6.25 Add Teachers manually Testing	88
6.26 Remove Student Testing	88
6.27 Selected Category Testing	89
6.28 Reminder button Testing	89
6.29 View result Testing	89
6.30 Download result button Testing	89
6.31 View/edit Profile picture Testing	90
6.32 View/edit Email Testing	90
6.33 View/edit Password Testing	90
6.34 Save changes button Testing	91
6.35 Progress bar Testing	91
6.36 Text Field Testing	91
6.37 Send Response Button Testing	91
6.38 Sign in Integration Testing	92
6.39 View Stats Integration Testing	92
6.40 Create Student Forms Integration Testing	93

6.41 Create Final year Student Forms Integration Testing	93
6.42 Create Alumina Forms Integration Testing	93
6.43 Create Organization Forms Integration Testing	93
6.44 Create Registration Forms Integration Testing	94
6.45 Results Integration Testing	94
6.46 Add User Integration Testing	94
6.47 Add Teacher Integration Testing	95
6.48 Add Admin Integration Testing	95
6.49 Fill form Integration Testing	95
6.50 Send reminders Integration Testing	96
6.51 Edit Profile Integration Testing	96
6.52 Change mode Integration Testing	96
6.53 Sign in System Testing	97
6.54 Create user System Testing	98
6.55 Create Teacher System Testing	99
6.56 Create Admin System Testing	100
6.57 Create Forms System Testing	101
6.58 Download result System Testing	102
6.59 Send Reminders System Testing	103
6.60 Edit Profile System Testing	104
6.61 Fill Form System Testing	105
6.62 Sign in Acceptance Testing	106
6.63 Create user Acceptance Testing	107
6.64 Create Teacher Acceptance Testing	108
6.65 Create Admin Acceptance Testing	109
6.66 Create Forms Acceptance Testing	110
6.67 Create Final Year Student Forms Acceptance Testing	111
6.68 Create Alumina Forms Acceptance Testing	112
6.69 Create Organization Forms Acceptance Testing	113
6.70 Download result Acceptance Testing	114
6.71 Send Reminders Acceptance Testing	115
6.72 Edit Profile Acceptance Testing	116
6.73 Change mode Acceptance Testing	117
6.74 Load testing	118
6.75 Stress testing	119
6.76 Endurance testing	119
6.77 Spike testing	120
6.78 Usability Testing	122

Abbreviations

CLO	Course Learning Outcome
CQI	Continuous Quality Improvement
CSS	Cascading Style Sheet
DB	Data Base
ERD	Entity Relationship Diagram
HTML	Hyper Text Markup Language
ISO	International Organization for Standardization
LMS	Learning Management System
MERN	MySQL Express React Node
OBE	Outcome Based Education
PEO	Programme Educational Objectives
PLO	Programme Learning Outcome
RIT	Rochester Institute of Technology
SQL	Structured Query Language
UET	University of Engineering and Technology
UI	User Interface
UX	User EXperience
WBS	Work Breakdown Structure
XML	EXtensible Markup Language

Abstract

The proposed system aims to revolutionize the Continuous Quality Improvement (CQI) process in educational institutes by addressing the limitations of existing solutions like Google Forms or paid platforms such as Anthology. These platforms often lack optimization for CQI purposes or are prohibitively expensive without offering sufficient interactivity. Our proposed system offers a free and interactive alternative that can seamlessly integrate into existing systems. By leveraging the MERN stack, it provides a user-friendly interface for administrators to manage users, create customized forms, and send reminders efficiently. The system's design ensures easy navigation and accessibility, making it suitable for users of all levels. One of the standout features of our system is its robust data visualization capabilities. Administrators can view outcomes in well-represented visualizations, including graphs and charts, allowing for quick and meaningful insights into the data. Moreover, the system allows for data download in various convenient and usable formats, empowering administrators to further analyze and utilize the data as needed. Overall, our proposed system not only addresses the shortcomings of existing CQI solutions but also enhances the user experience, data analysis capabilities, and overall effectiveness of CQI processes in educational institutes. Our system aims to improve quality by fixing issues with tools like Google Forms or costly options like Anthology. Our free system, using the MERN stack, is easy for admins to use. They can manage users, make forms, and send reminders easily. The design is simple and works well for all users. One great thing about our system is how it shows data. Admins can see data in graphs and charts, giving them quick insights. They can also download data in different formats for more analysis. Our system makes improving quality in schools easier, better, and more accessible than before.

Chapter 1

Introduction

1.1 Introduction

Continuous Quality Improvement (CQI) systems are widely used in higher education institutions to assess and improve the quality of education. University of Engineering and Technology has incorporated a system of elective subject registration and outcomes i.e., CLO, PLO, PEO, that can help administration predict if the quality of education is up to the mark or not.

Elective subject registration ensures registration of each student in specific subjects, no student can register more than specified number of subjects. CLO stands for course learning outcome which analyze the quality of course being taught on basis of teaching methodologies and the contents being taught in a course. PLO stands for Program Learning Outcome which is analyzed on the basis of achievement of CLOs. They help analyze if the content throughout the program has taught you well enough. And the last one is PEO which stands for Program Education Outcome, which is analysed through alumni, ex students of the university who are currently working in their respective fields. Purpose of PEO is to analyze if the content taught was industry applicable and relative.

The traditional methods of collecting and analyzing student feedback data are often inefficient , time-consuming and far from being convenient. Therefore, there is a need for a more efficient and effective system that can streamline the process of data collection and analysis.

UET is currently using google forms and LMS for conducting their surveys. The major problems with using google forms as a CQI is first of all it may raise some security concerns, and second reason is it may not be integrated to other systems seamlessly like LMS, also it does not provide any kind of visualization and so does

the current LMS, It is not quite user friendly and does not provide any kind of visualization of the data. On the other hand this system will be a one stop shop for all surveys and would be more user friendly compared to the other options available.

The motivation for this project stems from the need to improve the quality of education by providing a user-friendly and convenient platform for data collection and analysis. The proposed system will enable university administrators to gain insights into student experiences and make data-driven decisions to improve the quality of education.

This proposal outlines the development of a web application using the MySql Database, Express, React, and Node.js that aim to address the challenges of the traditional methods and improve the CQI process. The methodology for developing this system will involve conducting a needs analysis, designing a database schema, creating RESTful APIs using Express, building a user-friendly interface using React, and implementing data visualization tools using Chart.js. The system will also be tested for functionality, usability, and security.

The proposed system has the potential to be used by higher education institutions to improve the quality of education. It will provide a platform for efficient data collection, analysis, and visualization, enabling university administrators to gain insights into student experiences and make data-driven decisions to improve the quality of education..

1.2 Objectives

The objectives behind this project are

- To provide an efficient system for registration of elective subjects.
- To provide an efficient, user friendly and quick system for conducting student surveys
- To provide insights obtained from the surveys in a usable and convenient format for further analysis
- To visualize the data in such manner that it is easy to comprehend and examine the change over time
- To enhance the overall quality of education by providing a platform for continuous quality improvement that maintains a record of Quality Enhancement over time

1.3 Problem Statement

Inefficient , time-consuming and difficult to use traditional methods of collecting and analyzing student feedback data pose a challenge in higher education institutions, leading to delayed insights and decision-making. This highlights the need for a more optimized and user-friendly system that can facilitate efficient data collection, analysis, and visualization. Such a system will enable university administrators to gain insights into student experiences and make informed decisions to improve the quality of education.

1.4 Assumptions and Constraints

Assumptions and Constraints of project are given below.

1.4.1 Assumptions

Following were the assumptions from the beginning of the project

- Communication among team members will be effective and clear.
- Any external resources required for the project will be available and accessible.
- The technology and tools chosen for the project will remain stable and functional throughout.
- All tasks (design, making, testing, and writing) will be done on time.
- Our project supervisor will help us.
- Any unexpected issues or challenges will be addressed effectively.
- User credentials are safe, and everyone's data is private.
- The project team will have necessary skills for this project.
- The project documentation will be comprehensive and up-to-date throughout the project lifecycle.
- We'll achieve our project goals in the end.
- The project plan and timeline will be realistic and achievable.

1.4.2 Constraints

Some of the Constraints in our system are following:

- The User may have internet connection issues.
- Our web app can not be installed on Android or IOS as it is an online web application and can be accessed through internet.
- Time for submitting surveys or subject registration forms may vary due to poor internet connection of user.
- Time to send reminders may vary due to slow internet of Admin.

1.5 Project Scope and Features

In this section the scope and features of the project are discussed.

1.5.1 Project Scope

The Scope of this project extends to

- Designing, developing, and implementing a web application that can collect, visualize, and provide data for further analysis from feedback data by students, faculty and alumni
- Maintain record of previous surveys over time
- Maintain Profile
- Add or remove Admins/Students/Teachers
- Create subject registration forms
- Create custom survey questions
- Create special surveys on occasions if needed
- Allow downloading data in a reusable format

1.5.2 Features

The features of this project includes:

1. Login for Super Admin
2. Verification of admin credentials
3. Super Admin can add/remove Admins
4. Super Admin can add/remove Students
5. Super Admin can add/remove Teachers
6. Super Admin can make entries both by uploading .csv file or making an entry manually
7. Super Admin and Admins can create subject registration forms
8. Super Admin and Admins can create surveys
9. Super Admin and Admins can edit surveys
10. Super Admin and Admins can delete surveys
11. Super Admin and Admins can view survey reports
12. Super Admin and Admins can download survey reports in an Microsoft Excel file
13. Students, teachers, alumni can view and fill their respective surveys
14. Users will receive notifications

Chapter 2

Background Study

2.1 Literature Review

This is a CQI website which enables admin to create and analyze surveys. Some other sites like this do exist in market as per now. Major difference between them and this project is that, this project is much more goal oriented and focuses on one major goal i.e., student reviews and quality enhancement of university, also all of these websites, despite being very interactive are still not very easy to use. Some of them are listed below.

2.1.1 Anthology

Since Anthology[2] is one of the most famous CQI system available for usage that co-relates to our goal of Quality enhancement for Educational Institutes. But it does have some limitations like it can be very difficult to use, and it is a paid app so that is not a good thing for obvious reasons. Limited customizability , is also one of the problems faced by Anthology user, although it does provide customizability to some extent but it does not provide complete customizability .

Also we can see that there form design is very basic which fails to gain user attention. See Figure2.1

The screenshot shows a survey titled "2017 Event Post-Assessment" under the section "Post-Assessment". A blue header bar contains the instruction: "Please answer the following questions about your experience at the event". Below this, there is a question asking for a rating: "Overall, how would you rate the event?" with options: Poor, Fair, Good, Very good, and Excellent. Another question asks: "What did you like the most about the event?" followed by a large text input field.

FIGURE 2.1: Anthology Designed form for University of Mississippi

2.1.2 Google Forms

Google forms[7] is very useful software for conducting surveys since it is completely customizable and very easy to use. But the problem with using Google forms as a CQI system is we cannot have any kind of visualization for the data we received. So it is good for surveys that take place once or twice but not very useful as a system that needs to conduct these surveys regularly, also it might not integrate with existing systems seamlessly.

Even though Google forms provide many customizable and interactive templates, it is not designed to be a part of a system and will not provide data in Microsoft Excel form instead it gives data in Google sheets, but most of the analysis programs are designed to use Microsoft Excel format. See Figure 2.2

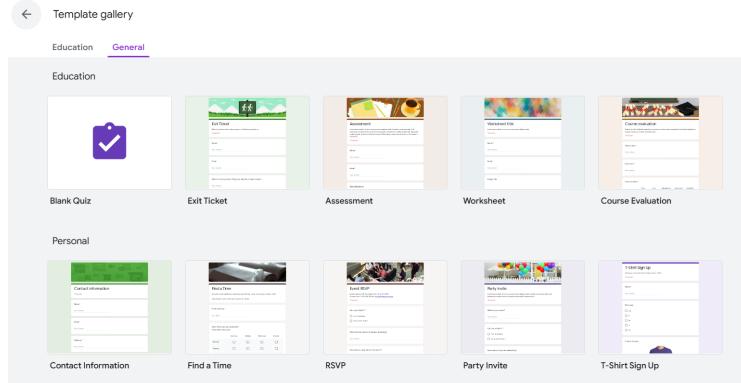


FIGURE 2.2: Google forms Templates

2.1.3 Continuous Quality Improvement in Higher Education

In response to public demands, colleges and universities are increasingly focusing on continuous quality improvement, particularly in their curricula. For instance, the Rochester Institute of Technology's (RIT) College of Business experienced a higher attrition rate around the junior year[29]. To address this, they surveyed students who had switched majors or transferred institutions to understand the reasons behind this trend. Many students expressed feelings of alienation, primarily because they didn't take any required business courses until their junior year and thus didn't feel integrated into the college.

Responding to these concerns, RIT's College of Business developed a five-quarter sequence of courses called "Business Concepts," mandatory for all freshmen and sophomores. These courses focus on team building, problem-solving, oral presentations, and theoretical aspects of continuous quality improvement. The feedback from current students in the "Business Concepts" sequence has been overwhelmingly positive.

Additionally, RIT's College of Business requires two quarters of cooperative education (internship) credit for every undergraduate business major. The Placement Office assists students in finding suitable internships. These curricular changes, including the "Business Concepts" sequence and the internship requirement, have effectively met or exceeded student needs and expectations.

It's recognized that curriculum and instruction are the areas most likely to see change through continuous quality improvement efforts.

2.1.4 Effect of OBE systems

Globally, it is widely acknowledged that outcome-based education systems offer several advantages. Unlike traditional systems that often emphasize memorization, outcome-based education focuses on helping students acquire knowledge and improve their performance, particularly in real-world settings like industry.

Outcome-Based Education (OBE)[31] centers on two main aspects of student achievement within an academic program. One is the Programme Outcomes, which are assessed at the time of graduation, and the other is the Programme Educational Objectives (PEOs), evaluated over a longer period, typically around 4–5 years post-graduation. These PEOs are aligned with the guidelines set by the Engineering Council to meet the requirements of the Engineering Accreditation Council in Malaysia.

A study was conducted to map these PEOs and formulate an anonymous online questionnaire survey to gauge the attainment of these objectives. The findings of this study indicate that graduates generally express satisfaction with their achievements across all eight PEOs. However, strategies have been proposed to enhance attainment levels in four PEOs with relatively lower attainment rates, as part of the continuous quality improvement (CQI) process adopted by the department. See Figure 2.3



FIGURE 2.3: The proposed linkage of CitationSpady's (1994) top to bottom levels of outcomes development with the incorporation of long-term PEOs.

2.1.5 Swedish and American Model for CQI

The Swedish model for Quality Assurance and Enhancement in Higher Education[30] is akin to the self-regulation concepts discussed by Graham et al. as cited by Bowden and Marton[28]..

All higher education institutions are required to develop their own quality assurance systems and report on quality measures in triennial reports to the government in connection with the appropriation proposal for the next 3-year cycle.

In the United States, accreditation of dental schools has led to the establishment of procedures for continuous quality improvement. During the self-study process, recommendations for improvements are generated. As part of the site visit, an evaluation is conducted to assess the implementation of these recommendations.

The review of existing CI implementation models by B. A. Lameijera, H. Boerb, J. Antony and R. J. M. M. Does [26] reveals several findings:

- Limited and biased guidance:
 - There is a dominant focus on implementation readiness factors rather than sustainability factors that ensure ongoing results from implementation activities.

- Anecdotal and expert-opinion-based support:
 - Anecdotal and expert-opinion-based support: Most models rely on anecdotal or expert opinions rather than empirical evidence.
- Unclear relationship between activities and performance effects:
 - The connection between implementing guidance and organizational performance effects remains unclear in many models.
- Lack of consideration for firm contextual factors:
 - Factors like organizational size, industry, or national context are not adequately addressed in the guidance provided by existing models.

In the domain of quality enhancement and progress, the creation of a Continuous Quality Improvement (CQI)^[3] [4] system signifies a pivotal stride towards augmenting organizational processes and service delivery. This multifaceted system seeks to streamline the evaluation process through efficient survey management, empowering administrators with a robust toolkit for data manipulation and analysis. As we delve into the functional and non-functional prerequisites of this cutting-edge CQI system, meticulous consideration of its capabilities and attributes is imperative to ensure seamless functionality, reliability, security, and user contentment.

The examples above demonstrate practical applications of CQI in a systematic and transparent way, which can be adopted as a standard approach.

2.1.6 Role of Automation in Improvement of Quality

According to ISO 9001-2011 of this article [27], how a business processes involves using resources to transform inputs into outputs and is crucial for successful work and maximizing profits.

This article "The role of automation in improving the quality of enterprise business processes"^[27] introduces the concept of a "process approach" as defined by ISO 9001-2011, emphasizing the importance of managing processes to achieve desired results continuously. The article discusses automating individual business processes within an enterprise using information technology. An automated application is proposed in the study for tasks such as automatic search for non-compliant part sizes, construction of control charts, and calculation of process reproducibility indices.

For weaknesses and their solutions of above systems view Table 2.1

TABLE 2.1: Related Systems Analysis

Related System	Weakness	Proposed Project Solution
Anthology[2]	A paid website with very non-interactive and non appealing forms least customisable	A free alternative with interactive, appealing and fully customisable forms.
Google Forms[7]	Doesn't provide visualization of data also it does not store data in our system instead stores it on Google sheet.	Data should be visualized and stored on our system i.e., in Excel format.
Survey Monkey[15]	Displays a simple interface to user and consist of complex set of tools to make surveys also it does not provide free sign-up.	An attractive and easy to use and understand interface displayed to user.
Metrics for learning[10]	It does not capture the quality of user engagement, such as whether or not the content is engaging and understandable.	Already created interesting editable surveys will be present.
Qualtrics[11]	It has the risk of data breaches which compromises the confidentiality of study participants.	Data should be secured and user's privacy should be kept along with data confidentiality.
Typeform[16]	Limited customization options for form design and branding.	Introduce more flexible pricing plans and expand customization capabilities.
Alchemer[1]	Complexity in interface and navigation for new users.	Provide more user-friendly resources to improve usability.
Jotform[9]	Limited integration options with complex business systems.	Enhance integration capabilities wider range of third-party applications to improve workflow automation.

Continued on next page

Table 2.1 – continued from previous page

Related System	Weakness	Proposed Project Solution
SurveySparrow[14]	Limited advanced survey logic and branching capabilities.	Enhance survey platform with more robust logic options and features.
ZohoSurvey[17]	Limited customization and advanced survey logic.	Enhance with customizable templates and advanced logic features.
Formstack[6]	Complexity in form building for new users.	Provide intuitive tutorials and templates to simplify form creation process.
Microsoft Forms[19]	Limited customization options for complex survey designs.	Introduce advanced customization features for more intricate survey structures.
Survey Anyplace[23]	Limited integration options with other business systems.	Expand integration capabilities to seamlessly connect with range of apps.
SoGoSurvey[22]	Lack of advanced data analysis tools.	Enhance with robust analytics features for deeper insights.
QuestionPro[20]	Complexity in interface for new users.	Provide user-friendly onboarding and tutorials to improve usability.
Survey Legend[24]	Limited integration options with third-party tools.	Enhance integration capabilities to connect with external systems.
Survey Planet[25]	Lack of advanced analytics features.	Introduce advanced analytics tools for deeper insights.
Get FeedBack[18]	Limited customization options for complex survey designs.	Enhance with more advanced customization features.
Smart Survey[21]	Steeper learning curve due to advanced features.	Provide comprehensive tutorials and support resources.

Continued on next page

Table 2.1 – continued from previous page

Related System	Weakness	Proposed Project Solution
QuickTap Survey[12]	Limited online functionality and dependence on mobile devices.	Enhance web-based features for broader accessibility and usability.
Crowd Signal[5]	Limited customization options for survey design.	Introduce more customizable templates and advanced design.

Chapter 3

Requirement Analysis

3.1 Requirement Elicitation

We have gathered requirements of our project from Department Teachers, from their reviews on different websites like Anthology[2], Survey Monkey[15], Qualtrics[11], Metrics for Learning[10] etc ,we learned the issues they faced with these websites and covered them in our project.

3.1.1 Functional Requirement

Functional requirements of the CQI system should align to the model that provides a usable feedback. The model should follow a proven structure that seems to work adequately and improve over it, Like Google Forms[7], Anthology[2], Survey Monkey[15], Qualtrics[11], Metrics for Learning[10].

3.1.1.1 Security Requirements

Security requirements are referred by Table 3.1 2.1

TABLE 3.1: Security Requirements (FR-01-00)

Index	Priority	Requirements
FR-01-01	1	The system shall support a authorized login system.
FR-01-02	2	The system shall allow user to fill form only once.
FR-01-03	1	The system shall allow only Admins to login to system.

3.1.1.2 User Requirements

User requirements are referred by Table 3.2

TABLE 3.2: User Requirements (FR-02-00)

Index	Priority	Requirements
FR-02-01	1	The system shall display surveys in a user-friendly manner to encourage user participation and engagement.
FR-02-02	1	The system shall allow the user to take surveys.
FR-02-03	2	The system shall allow the user to check which surveys they have submitted.

3.1.1.3 Admin Requirements

Admin requirements are referred by Table 3.3

TABLE 3.3: Admin Requirements (FR-03-00)

Index	Priority	Requirements
FR-03-01	1	The system shall allow the superadmin to create multiple administrators.
FR-03-02	1	The system shall provide administrator privileges that enable form creation.
FR-03-03	2	The system shall allow the administrator to edit a form.
FR-03-04	2	The system shall allow the administrator to delete a form.
FR-03-05	1	The system shall facilitate access for administrators to visual representations of statistical data.
FR-03-06	1	The system shall allow administrators to download survey results in a readily usable format.
FR-03-07	1	The system shall allow the administrator to create users.
FR-03-08	1	The system shall allow the administrator to save teacher information.
FR-03-09	3	The system shall allow the administrator to update username.
FR-03-010	3	The system shall allow the administrator to update their password.

3.1.2 Non-functional Requirement

Non-functional Requirements do not affect overall working of system. Some Non-functional requirements of system are given below.

3.1.2.1 Performance

Performance requirements are referred by Table 3.4

TABLE 3.4: Performance (NFR-01-00)

Index	Priority	Requirements
NFR-01-01	1	The system must exhibit prompt responsiveness, delivering meaningful interactions within a five-second time-frame.

3.1.2.2 Scalability

Scalability requirements are referred by Table 3.5

TABLE 3.5: Scalability (NFR-02-00)

Index	Priority	Requirements
NFR-02-01	1	The system must accommodate up to 10,000 users on a proficiently configured server to safeguard operational efficiency during expansion.

3.1.2.3 Availability

Availability requirements are referred by Table 3.6

TABLE 3.6: Availability (NFR-03-00)

Index	Priority	Requirements
NFR-03-01	1	The system must maintain uninterrupted 24/7 operation, facilitating surveys at any time interval.

3.1.2.4 Capacity

Capacity requirements are referred by Table 3.7

TABLE 3.7: Capacity (NFR-04-00)

Index	Priority	Requirements
NFR-04-01	1	The system must manage 1,000 simultaneous users and store data for the same user count.

3.1.2.5 Reliability

Reliability requirements are referred by Table 3.8

TABLE 3.8: Reliability (NFR-05-00)

Index	Priority	Requirements
NFR-05-01	1	The system must deliver precise outcomes consistently, granting administrators the capacity to manage data.
NFR-05-02	1	The system must possess resilience to mitigate critical errors and ensure uninterrupted functionality.

3.1.2.6 Maintainability

Maintainability requirements are referred by Table 3.9

TABLE 3.9: Maintainability (NFR-06-00)

Index	Priority	Requirements
NFR-06-01	1	The system must provide proactive oversight by programmers to swiftly resolve issues.

3.1.2.7 Serviceability

Serviceability requirements are referred by Table 3.10

TABLE 3.10: Serviceability (NFR-07-00)

Index	Priority	Requirements
NFR-07-01	1	The system must address component-level concerns with agility and rapid responsiveness to user requests

3.1.2.8 Interoperability

Interoperability requirements are referred by Table 3.11

TABLE 3.11: Interoperability (NFR-08-00)

Index	Priority	Requirements
NFR-08-01	1	The system must facilitate seamless data exchange with other systems.

3.1.2.9 Security

Security requirements are referred by Table 3.12

TABLE 3.12: Security (NFR-09-00)

Index	Priority	Requirements
NFR-09-01	1	The system must ensure safeguards against unauthorized access and data breaches.

3.1.2.10 Regulatory

Regulatory requirements are referred by Table 3.13

TABLE 3.13: Regulatory (NFR-10-00)

Index	Priority	Requirements
NFR-10-01	1	The system must observe compliance with local and international laws.

3.1.2.11 Manageability

Manageability requirements are referred by Table 3.14

TABLE 3.14: Manageability (NFR-11-00)

Index	Priority	Requirements
NFR-11-01	1	The system must pivot on user-friendly nature, catering to diverse user profiles.
NFR-11-01	2	The system must protect the environment through conscious design choices.

3.1.2.12 Data Integrity

Data Integrity requirements are referred by Table 3.15

TABLE 3.15: Data Integrity (NFR-12-00)

Index	Priority	Requirements
NFR-12-01	1	The system must ensure user information remains confidential and untampered

3.1.2.13 Usability

Usability requirements are referred by Table 3.16

TABLE 3.16: Usability (NFR-13-00)

Index	Priority	Requirements
NFR-13-01	1	The system must grant an interactive user experience.

3.1.3 Requirement Traceability Matrix

A Requirement Traceability Matrix (RTM) is used to track and link requirements throughout the project lifecycle. It establishes a relationship between different parts of project such as requirements, design documents, test cases, and deliverables. The main purpose of an RTM is to ensure that all requirements are addressed and tested appropriately. For reference see Table 3.17

TABLE 3.17: Requirement Traceability Matrix

Req No.	Req Description	Test Case ID	Status
1	Sign-in to web-app	ID 001	Passed
2	Create a new User	ID 002	Passed
3	Create new Teacher	ID 003	Passed
4	Create new Admin	ID 004	Passed
5	Create Student Form	ID 005	Passed
6	Create Final year Student Form	ID 006	Passed
7	Create Alumina Form	ID 007	Passed
8	Create Organization Form	ID 008	Passed
9	Download Results	ID 009	Passed
10	Send Reminders	ID 010	Passed
11	Edit Profile	ID 011	Passed
12	Change Mode	ID 012	Passed

3.2 Use-case Description

The following are the description of all the use cases of the project. Use cases of the system include login, creating users, creating admins, store teacher information, adding new form, selecting forms, filling forms, submitting the filled forms, deleting forms, editing forms, manage admin profile, editing name, edit password, view stats, download result.

3.2.1 Login

Starting with the Login use case, the primary aim is to ensure that only individuals with proper authorization can gain access to the system, thereby enhancing security measures and safeguarding sensitive information. By restricting access to authorized personnel, the system mitigates the risk of unauthorized data breaches and maintains confidentiality, integrity, and availability of resources. To ensure safe login, a new React hook called ‘useRequireAuth’ was implemented, along with a context named ‘AdminContext’, which allows for a global scope state management, enabling efficient authentication checks across various components. Refer to table 3.18 for details of Login use case

TABLE 3.18: User Login

Name	Login
Participating Actors	Admin
Goal	Display a login form
Trigger	When the website is visited
Pre-Condition	<ul style="list-style-type: none"> • Stable internet connection.
Post-Condition	Redirect to home page
Basic Flow	<ul style="list-style-type: none"> • Admin shall open the website.
Alternate Flow	N/A.
Exceptions	<ul style="list-style-type: none"> • Server Error <ul style="list-style-type: none"> – Display Error "Server Error" • Connection Error <ul style="list-style-type: none"> – Display message "Check Internet Connection "
Qualities	<ul style="list-style-type: none"> • Must be available 24/7 • Must be responsive and quick

3.2.2 Create Users

There are two ways to create a user for which the forms are created, the details of which are written in table 3.19

TABLE 3.19: Add User

Name	Create Users
Participating Actors	Admin
Goal	Add New Users to Database
Trigger	When User Information is entered and submitted
Pre-Condition	<ul style="list-style-type: none"> • Stable internet connection.
Post-Condition	Add Users to database and send them a confirmation email
Basic Flow	<ul style="list-style-type: none"> • Admin shall open the website. • Admin shall select add user from navigation • Admin Shall enter the user data. • Admin shall press Confirm button
Alternate Flow	<ul style="list-style-type: none"> • Admin shall open the website. • Admin shall select add user from navigation • Admin Shall upload the user data from a .csv file. • Admin Shall press Confirm Button
Exceptions	<ul style="list-style-type: none"> • Server Error <ul style="list-style-type: none"> – Display Error "Server Error" • Connection Error <ul style="list-style-type: none"> – Display message "Check Internet Connection "
Qualities	<ul style="list-style-type: none"> • Must be available 24/7 • Must be responsive and quick

3.2.3 Create Admins

The details of Add Admin use case are described in table 3.20

TABLE 3.20: Add Admin

Name	Create Admins
Participating Actors	SuperAdmin
Goal	Add New Admins to Database
Trigger	When Admins Information is entered and submitted
Pre-Condition	Stable internet connection.
Post-Condition	Add Admins to database and send them a confirmation email
Basic Flow	<ul style="list-style-type: none"> • Admin shall open the website. • Admin shall select add admin from navigation • Admin Shall enter the admin data. • Admin shall review the info entered through the table displayed on the other side • Admin shall press Confirm button
Alternate Flow	<ul style="list-style-type: none"> • Admin shall open the website. • Admin shall select add user from navigation • Admin Shall upload the user data from a .csv file. • Admin Shall press Confirm Button
Exceptions	<ul style="list-style-type: none"> • Server Error <ul style="list-style-type: none"> – Display Error "Server Error" • Connection Error <ul style="list-style-type: none"> – Display message "Check Internet Connection "
Qualities	<ul style="list-style-type: none"> • Must be available 24/7 • Must be responsive and quick

3.2.4 Store Teachers Information

Refer to table 3.21 for details of this use case

TABLE 3.21: Store Teachers Information

Name	Store Teachers Info
Participating Actors	Admin
Goal	Add New Teachers to Database
Trigger	When teacher Information is entered and submitted
Pre-Condition	<ul style="list-style-type: none"> • Stable internet connection.
Post-Condition	Add Teacher to database
Basic Flow	<ul style="list-style-type: none"> • Admin shall open the website. • Admin shall select add teacher from navigation • Admin Shall enter the teacher data. • Admin shall press Confirm button
Alternate Flow	<ul style="list-style-type: none"> • Admin shall open the website. • Admin shall select add teacher from navigation • Admin Shall upload the teacher data from a .csv file. • Admin Shall press Confirm Button
Exceptions	<ul style="list-style-type: none"> • Server Error <ul style="list-style-type: none"> – Display Error "Server Error" • Connection Error <ul style="list-style-type: none"> – Display message "Check Internet Connection "
Qualities	<ul style="list-style-type: none"> • Must be available 24/7 • Must be responsive and quick

3.2.5 Select Form

User can select the forms generated for them using their emails, a link of their respective form will be present in the email. Users can click on the link to access the form which will open in a browser. The details of the select form use case are given in the table below, refer to table 3.22 for more details.

TABLE 3.22: Select Form

Name	Select Form
Participating Actors	User
Goal	Display a survey relevant to the user
Trigger	When Link clicked from email
Pre-Condition	<ul style="list-style-type: none"> • Stable internet connection. • Link Clicked
Post-Condition	Redirect to Form whose link was clicked
Basic Flow	<ul style="list-style-type: none"> • User shall open their email. • User shall open hyperlink attached in the email
Alternate Flow	NA
Exceptions	<ul style="list-style-type: none"> • Server Error <ul style="list-style-type: none"> – Display Error "Server Error" • Connection Error <ul style="list-style-type: none"> – Display message "Check Internet Connection "
Qualities	<ul style="list-style-type: none"> • Must be available 24/7 • Must be responsive and quick

3.2.6 Fill Form

After selecting the form user is supposed to fill the form and submit response. For the flow and details of the fill form details refer to table 3.23

TABLE 3.23: Fill Form

Name	Fill Form
Participating Actors	User
Goal	Display selected survey form
Trigger	When Form is selected
Pre-Condition	<ul style="list-style-type: none"> ● Stable internet connection. ● Form Selected
Post-Condition	Redirect to form page
Basic Flow	<ul style="list-style-type: none"> ● User shall open their email. ● User shall open hyperlink attached in the email ● user shall answer the questions asked
Alternate Flow	N/A
Exceptions	<ul style="list-style-type: none"> ● Server Error <ul style="list-style-type: none"> – Display Error "Server Error" ● Connection Error <ul style="list-style-type: none"> – Display message "Check Internet Connection "
Qualities	<ul style="list-style-type: none"> ● Must be available 24/7 ● Must be responsive and quick

3.2.7 Submit Form

Refer to table 3.24 for the details of this use case

TABLE 3.24: Submit Form

Name	Submit Form
Participating Actors	User
Goal	Submit survey response
Trigger	When Form is completely filled and the submit button is clicked
Pre-Condition	<ul style="list-style-type: none"> • Stable internet connection. • Form Selected • Form Filled and Submit Button Clicked
Post-Condition	Redirect to home page after showing a success message
Basic Flow	<ul style="list-style-type: none"> • User shall open their email. • User shall open hyperlink attached in the email • user shall answer the questions asked • User shall click the submit button
Alternate Flow	N/A
Exceptions	<ul style="list-style-type: none"> • Server Error <ul style="list-style-type: none"> – Display Error "Server Error" • Connection Error <ul style="list-style-type: none"> – Display message "Check Internet Connection "
Qualities	<ul style="list-style-type: none"> • Must be available 24/7 • Must be responsive and quick

3.2.8 Add New Form

Adding new forms require admin to enter certain details and the form questions, details of all are given in the table 3.25

TABLE 3.25: Add New Form

Name	Add New Form
Participating Actors	Admin
Goal	Display a dynamic form that take inputs for form data
Trigger	When the create form button is pressed from admin dashboard
Pre-Condition	<ul style="list-style-type: none"> • Stable internet connection. • Admin logged in successfully
Post-Condition	Redirect to Create Form page
Basic Flow	<ul style="list-style-type: none"> • Admin shall open the website. • Admin Shall login with correct admin credentials • Admin shall click on the create form button • Admin shall enter the title, topic and questions. • Admin shall select the category • Admin shall click the submit button
Alternate Flow	N/A.
Exceptions	<ul style="list-style-type: none"> • Server Error <ul style="list-style-type: none"> – Display Error "Server Error" • Connection Error <ul style="list-style-type: none"> – Display message "Check Internet Connection "
Qualities	<ul style="list-style-type: none"> • Must be available 24/7 • Must be responsive and quick

3.2.9 Delete Form

Admins have the right to delete any form, but there are certain constraints to it, the details of this use case are described in table 3.26

TABLE 3.26: Delete Form

Name	Delete Form
Participating Actors	Admin
Goal	Delete any form according to the will of admin
Trigger	When the delete form button is pressed from desired form selection
Pre-Condition	<ul style="list-style-type: none"> • Stable internet connection. • Admin logged in successfully
Post-Condition	Form successfully deleted
Basic Flow	<ul style="list-style-type: none"> • Admin shall open the website. • Admin shall login with correct admin credentials • Admin shall select the form to be deleted • Admin shall click on the delete form button
Alternate Flow	N/A.
Exceptions	<ul style="list-style-type: none"> • Server Error <ul style="list-style-type: none"> – Display Error "Server Error" • Connection Error <ul style="list-style-type: none"> – Display message "Check Internet Connection " • Response of the form already exists <ul style="list-style-type: none"> – Does not allow deletion of this form
Qualities	<ul style="list-style-type: none"> • Must be available 24/7 • Must be responsive and quick

3.2.10 Edit Form

Forms can be edited after creation, the flow and exceptions of this use case are written in [3.27](#)

TABLE 3.27: Edit Form

Name	Edit Form
Participating Actors	Admin
Goal	Display a dynamic form that lets admin edit form data
Trigger	When the edit form button is pressed from desired form
Pre-Condition	<ul style="list-style-type: none"> ● Stable internet connection. ● Admin logged in successfully
Post-Condition	Redirect to Edit Form page
Basic Flow	<ul style="list-style-type: none"> ● Admin shall open the website. ● Admin shall login with correct admin credentials ● Admin shall select the form to be edited ● Admin shall click on the edit form button
Alternate Flow	N/A.
Exceptions	<ul style="list-style-type: none"> ● Server Error <ul style="list-style-type: none"> – Display Error "Server Error" ● Connection Error <ul style="list-style-type: none"> – Display message "Check Internet Connection " ● Response of the form already exists <ul style="list-style-type: none"> – Does not allow deletion of this form
Qualities	<ul style="list-style-type: none"> ● Must be available 24/7 ● Must be responsive and quick

3.2.11 Manage Profile

Admins are allowed to edit their information according to their will, to do so, they have to navigate to the edit profile page, details of this use case are described in table 3.28

TABLE 3.28: Manage Profile

Name	Manage Profile
Participating Actors	Admin
Goal	To edit profile information
Trigger	When the edit profile button is pressed from Navigation
Pre-Condition	<ul style="list-style-type: none"> • Stable internet connection. • logged in successfully
Post-Condition	Redirect to Edit Profile page
Basic Flow	<ul style="list-style-type: none"> • Admin shall open the website. • Admin shall login with correct credentials • Admin shall click on the edit profile button
Alternate Flow	N/A.
Exceptions	<ul style="list-style-type: none"> • Server Error <ul style="list-style-type: none"> – Display Error "Server Error" • Connection Error <ul style="list-style-type: none"> – Display message "Check Internet Connection "
Qualities	<ul style="list-style-type: none"> • Must be available 24/7 • Must be responsive and quick

3.2.12 Edit Username

Admins are allowed to edit their name, for details of this use case refer to the table [3.29](#)

TABLE 3.29: Edit Username

Name	Edit Username
Participating Actors	Admin
Goal	To Edit Username
Trigger	When the edit username button is pressed from edit profile tab
Pre-Condition	<ul style="list-style-type: none"> • Stable internet connection. • logged in successfully • manage profile section opened
Post-Condition	username updated
Basic Flow	<ul style="list-style-type: none"> • Admin shall open the website. • Admin shall login with correct credentials • Admin shall click on the edit profile button • Admin shall enter new username and click update button
Alternate Flow	N/A.
Exceptions	<ul style="list-style-type: none"> • Server Error <ul style="list-style-type: none"> – Display Error "Server Error" • Connection Error <ul style="list-style-type: none"> – Display message "Check Internet Connection "
Qualities	<ul style="list-style-type: none"> • Must be available 24/7 • Must be responsive and quick

3.2.13 Edit Password

Admins are allowed to change password for their convenience or in case of suspicion of security breach. Refer to table 3.30 for more details.

TABLE 3.30: Edit Password

Name	Edit Password
Participating Actors	Admin
Goal	To Edit Password
Trigger	When the edit password button is pressed from edit profile tab
Pre-Condition	<ul style="list-style-type: none"> • Stable internet connection. • logged in successfully • manage profile section opened
Post-Condition	password updated
Basic Flow	<ul style="list-style-type: none"> • Admin shall open the website. • Admin shall login with correct credentials • Admin shall click on the edit profile button • Admin shall click enter new password and click update button
Alternate Flow	N/A.
Exceptions	<ul style="list-style-type: none"> • Server Error <ul style="list-style-type: none"> – Display Error "Server Error" • Connection Error <ul style="list-style-type: none"> – Display message "Check Internet Connection "
Qualities	<ul style="list-style-type: none"> • Must be available 24/7 • Must be responsive and quick

3.2.14 View Stats

Admins can view the stats of all the forms in visual representation in forms of graphs, the flow and details of this use case are described in the table 3.31

TABLE 3.31: View Stats

Name	View Stats
Participating Actors	Admin
Goal	Watch graphical representation of stats
Trigger	When the view stats button is pressed from Navigation bar
Pre-Condition	<ul style="list-style-type: none"> • Stable internet connection. • Admin logged in successfully
Post-Condition	Redirect to View Stats page
Basic Flow	<ul style="list-style-type: none"> • Admin shall open the website. • Admin Shall login with correct admin credentials
Alternate Flow	N/A.
Exceptions	<ul style="list-style-type: none"> • Server Error <ul style="list-style-type: none"> – Display Error "Server Error" • Connection Error <ul style="list-style-type: none"> – Display message "Check Internet Connection "
Qualities	<ul style="list-style-type: none"> • Must be available 24/7 • Must be responsive and quick

3.2.15 View Results

Admin can view the results of the surveys taken, or seats reserved in tabular format, for further details refer to table 3.32

TABLE 3.32: View Results

Name	View Results
Participating Actors	Admin
Goal	View results of a survey in tabular form
Trigger	When the results button is pressed from Navigation
Pre-Condition	<ul style="list-style-type: none"> • Stable internet connection. • Admin logged in successfully
Post-Condition	View results successfully in a tabular format
Basic Flow	<ul style="list-style-type: none"> • Admin shall open the website. • Admin Shall login with correct admin credentials • Admin shall click on the retrieve button for a specific survey
Alternate Flow	N/A.
Exceptions	<ul style="list-style-type: none"> • Server Error <ul style="list-style-type: none"> – Display Error "Server Error" • Connection Error <ul style="list-style-type: none"> – Display message "Check Internet Connection " • No results available <ul style="list-style-type: none"> – No data is shown
Qualities	<ul style="list-style-type: none"> • Must be available 24/7 • Must be responsive and quick

3.2.16 Download Results

Admin can download the results of the surveys taken, or seats reserved in .csv format, for further details refer to table 3.33

TABLE 3.33: Download Results

Name	Download Results
Participating Actors	Admin
Goal	Download results of a survey
Trigger	When the download results button is pressed from Navigation bar
Pre-Condition	<ul style="list-style-type: none"> • Stable internet connection. • Admin logged in successfully
Post-Condition	download results successfully in a usable format[8]
Basic Flow	<ul style="list-style-type: none"> • Admin shall open the website. • Admin shall login with correct admin credentials • Admin shall click on the download results button of a specific survey
Alternate Flow	N/A.
Exceptions	<ul style="list-style-type: none"> • Server Error <ul style="list-style-type: none"> – Display Error "Server Error" • Connection Error <ul style="list-style-type: none"> – Display message "Check Internet Connection "
Qualities	<ul style="list-style-type: none"> • Must be available 24/7 • Must be responsive and quick

3.2.17 Forgot Password

Admin can recover password in case they forget 3.34

TABLE 3.34: Forgot Password

Name	Forgot Password
Participating Actors	Admin
Goal	Recover Password through email
Trigger	When the forgot password text is clicked from login page
Pre-Condition	<ul style="list-style-type: none"> • Stable internet connection. • Correct Admin email entered
Post-Condition	A new password generated, updated in database and sent to admin through email.
Basic Flow	<ul style="list-style-type: none"> • Admin shall open the website. • Admin shall enter correct email address • Admin shall click on the forgot password text on login buttons
Alternate Flow	N/A.
Exceptions	<ul style="list-style-type: none"> • Server Error <ul style="list-style-type: none"> – Display Error "Server Error" • Connection Error <ul style="list-style-type: none"> – Display message "Check Internet Connection "
Qualities	<ul style="list-style-type: none"> • Must be available 24/7 • Must be responsive and quick

3.3 Use-Case Diagram

The working of system through use case diagram, in which user will only fill form and admin will be displayed results is referred by Figure 3.1.

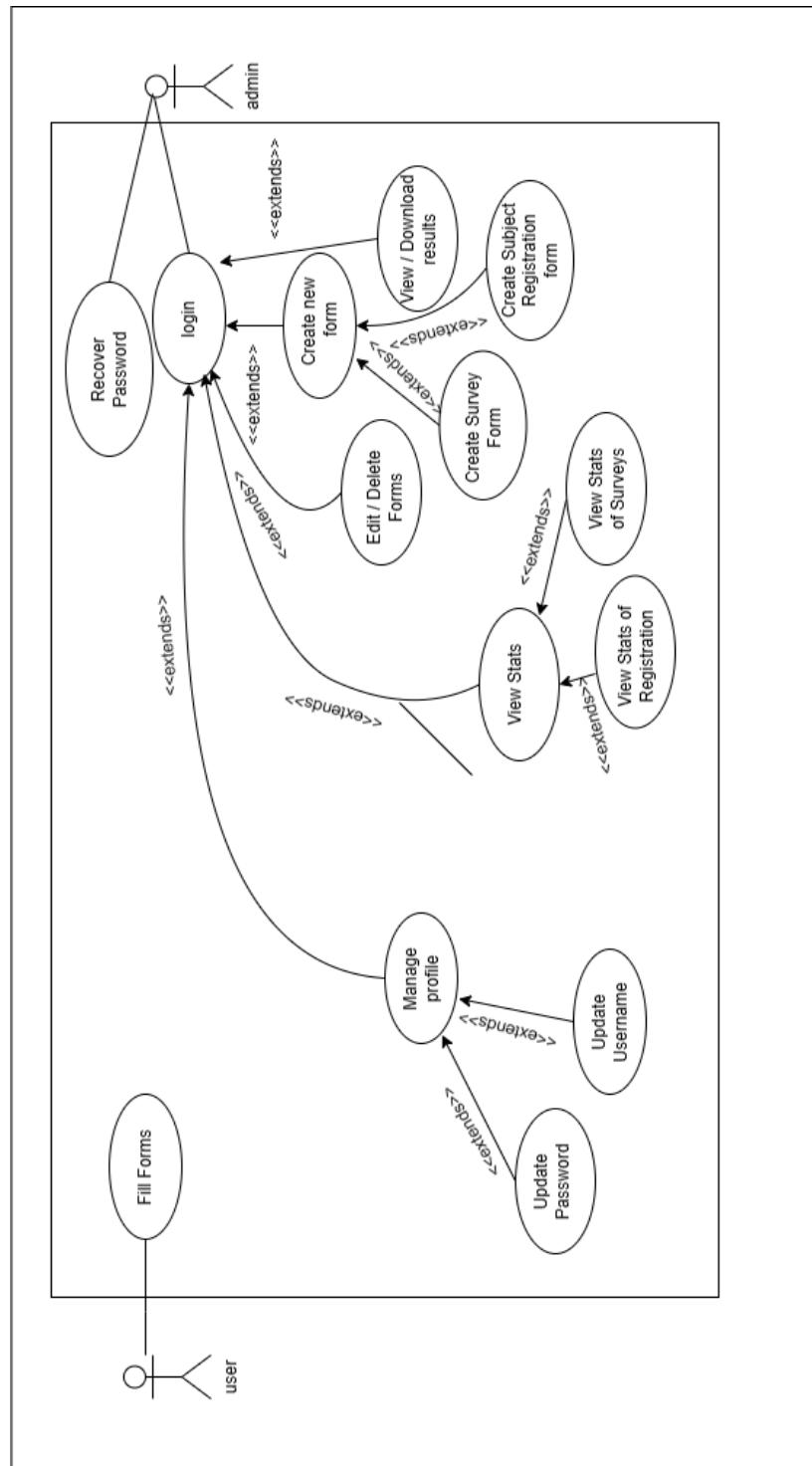


FIGURE 3.1: UseCase Diagram

Chapter 4

System Design

4.1 Work BreakDown Structure

The BreakDown of Working Structure into seven main parts which were further divided into more parts is referred by Figure 4.1.

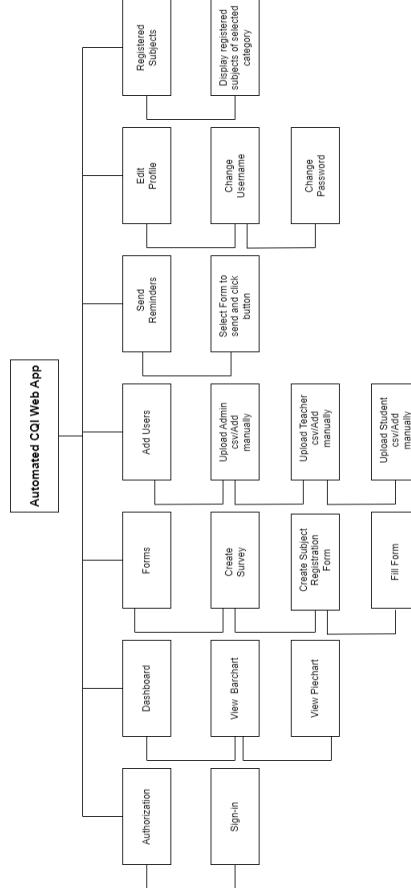


FIGURE 4.1: Work BreakDown Structure

4.2 Gantt chart

A Gantt chart is created to display the time allocation for project phases or iterations. This chart identifies major milestones and their achievement criteria, as well as estimates the duration of all necessary activities during project development, along with the responsible human resources for each task. [4.2](#) for further details.

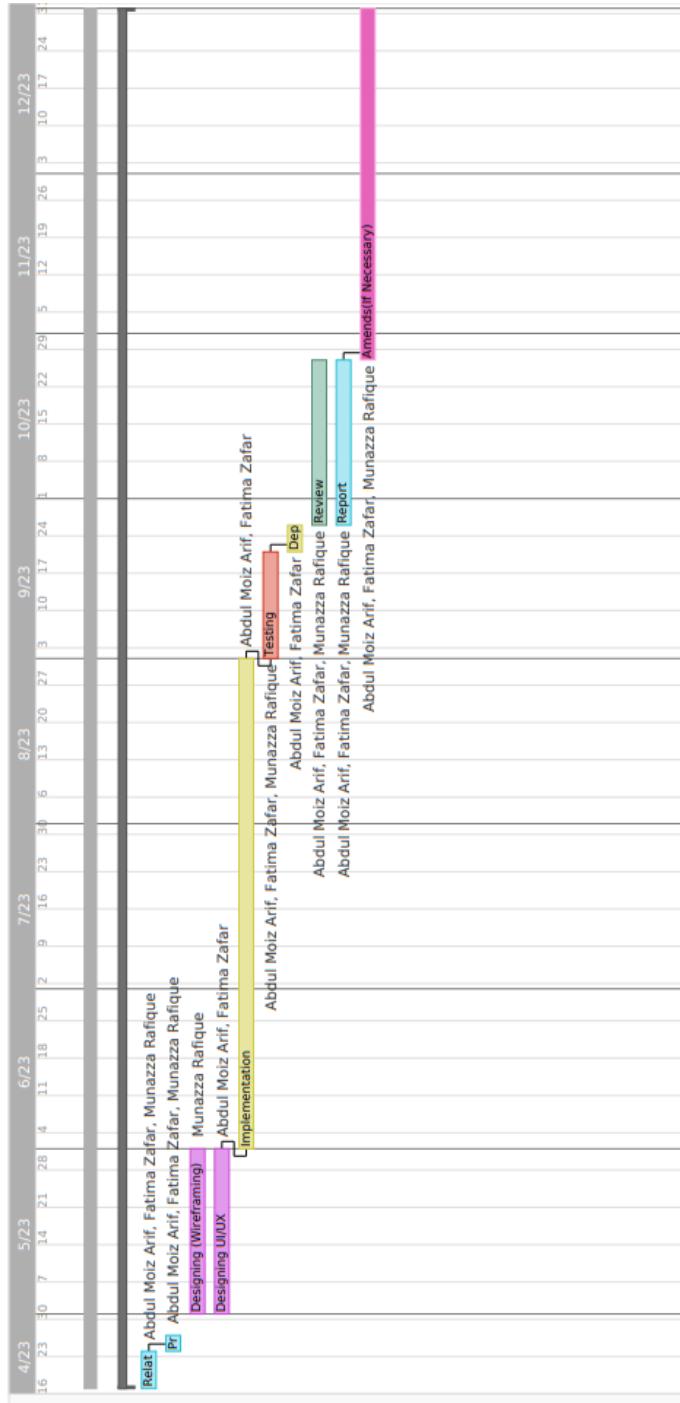


FIGURE 4.2: Gantt Chart

4.3 Wireframes

4.3.1 Log in

Login page design of system that contains logo, header and text inputs referred by Figure 4.3.

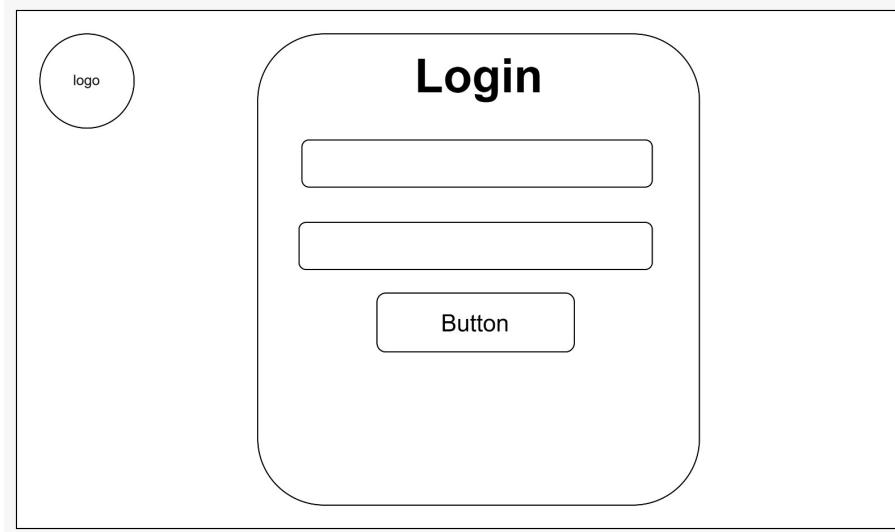


FIGURE 4.3: Login Wireframe

4.3.2 Fill Form

Fill Form page design of system that has questions and space to respond to the questions ,referred by Figure 4.4.

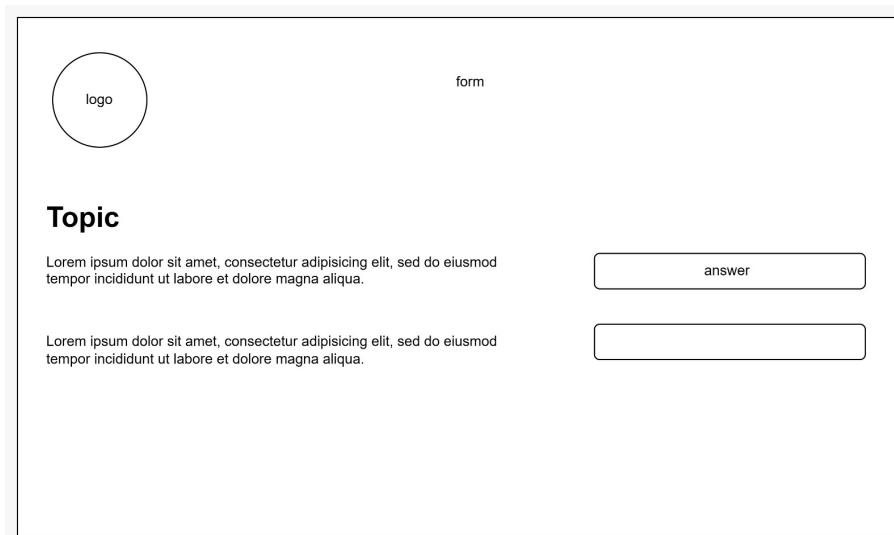


FIGURE 4.4: Fill Form Wireframe

4.3.3 Manage Profile

Profile page design of system that contains profile info of the admin,referred by Figure 4.5.

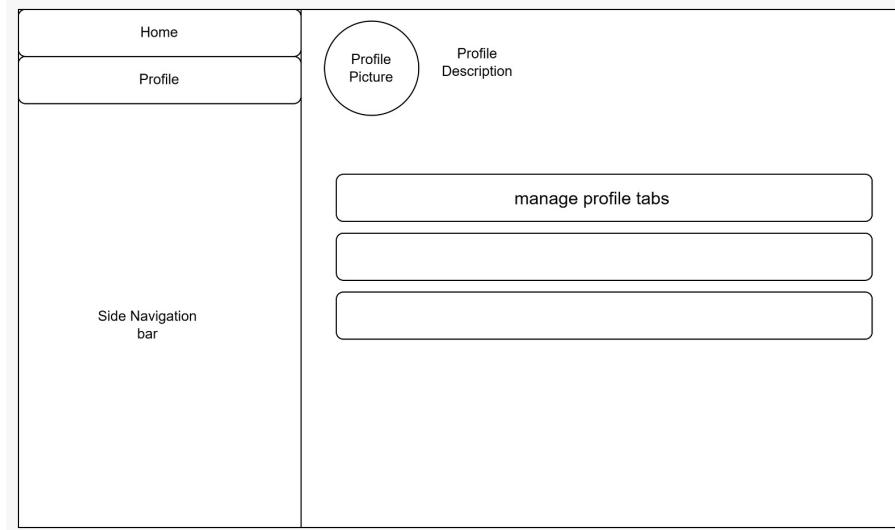


FIGURE 4.5: Manage Profile Wireframe

4.3.4 Manage Form

Form page design of system, that has all the options to edit or create form data,referred by Figure 4.6.

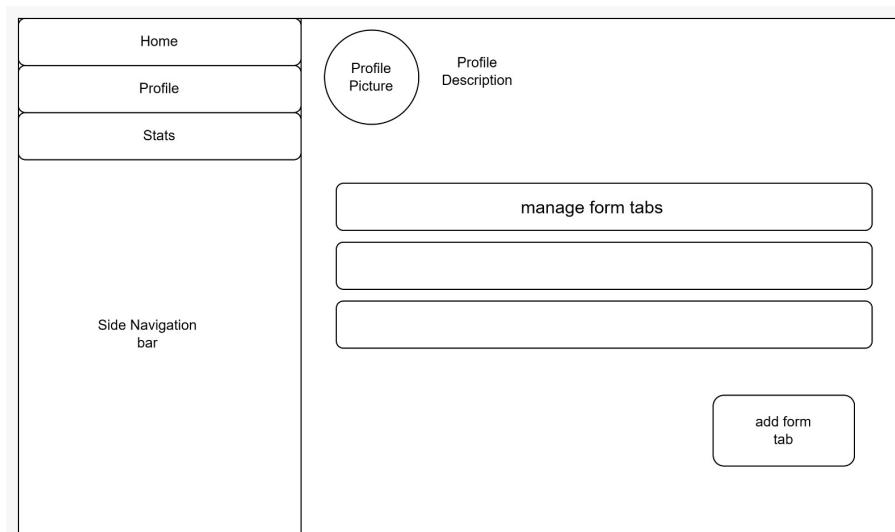


FIGURE 4.6: Manage Form Wireframe

4.3.5 Download Results

Download Result page design of system that contains a button to download the results of a form,referred by Figure 4.7.

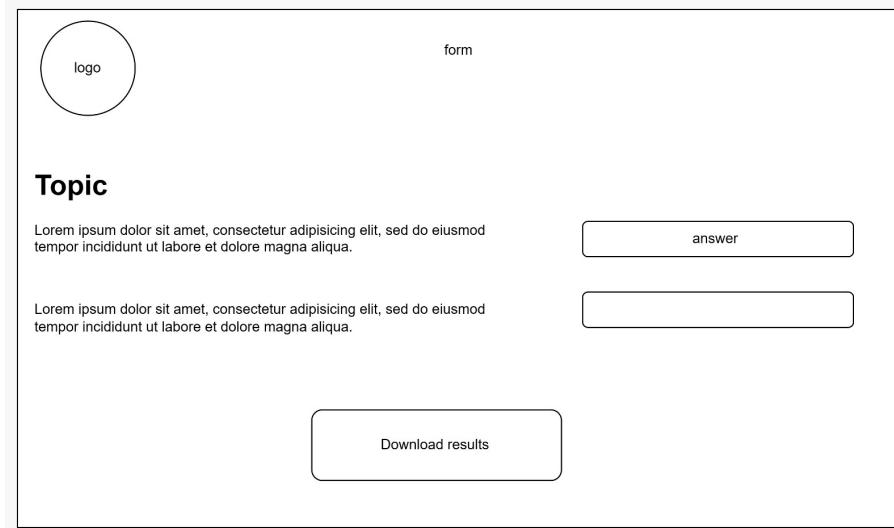


FIGURE 4.7: Download Results Wireframe

4.3.6 Create new Form

Page design for creating new survey in system that contains inputs for Form title, topic and questions,is referred by Figure 4.8.

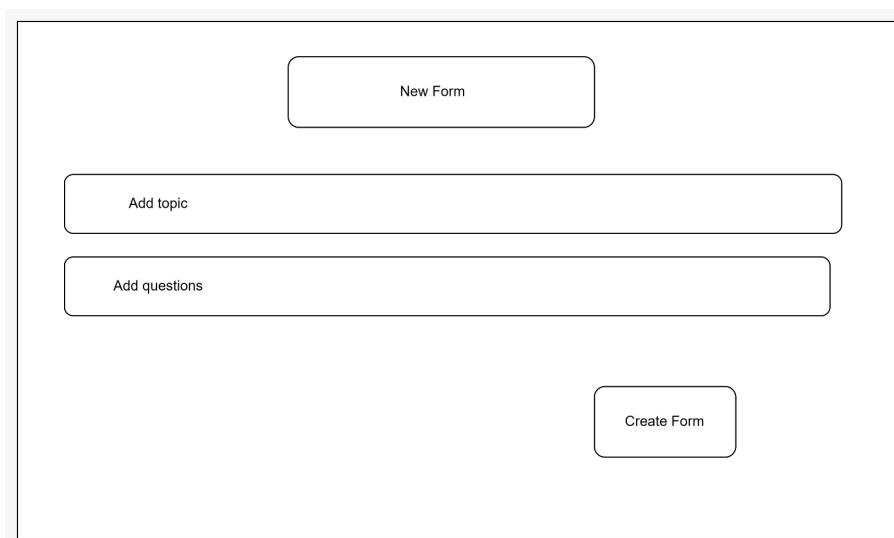


FIGURE 4.8: Create new Form Wireframe

4.3.7 Create new Subject Registration Form

Page design for creating new subject registration form that contains inputs to enter the number of subjects to be offered, name and available seats for that particular subject, is referred by Figure 4.9.

The wireframe shows a rectangular form with a title 'Subjects' at the top. Below the title are six input fields arranged in two columns of three. The first column contains two input fields stacked vertically, and the second column contains one input field. All input fields are represented by simple rectangles.

FIGURE 4.9: Create new Registration Form Wireframe

4.3.8 Add User/Admin/Teacher

Page design for adding Admin/Teacher/User to system. that has two sections. One has two inputs and a button to upload name and emails manually and another button that lets you upload a .csv file. Other section displays the names and emails of the users entered along with a delete button that lets you delete a certain user from the list, is referred by Figure 4.10.

The wireframe shows a page layout with three main sections. On the left, there are five input fields stacked vertically. In the center, there are two input fields, a 'Confirm' button, and a 'Delete' button. Below these are two buttons: 'Add from .csv' and 'Delete'. To the right, there is a table with several rows, each containing two columns. A 'Delete' button is located at the bottom right of the table area.

FIGURE 4.10: Add User/Admin/Teacher

4.3.9 View Stats

Stats page design of system that has an upper section to display responses of the survey forms in pie charts and bottom section displays results of subject registration forms, referred by Figure 4.11.

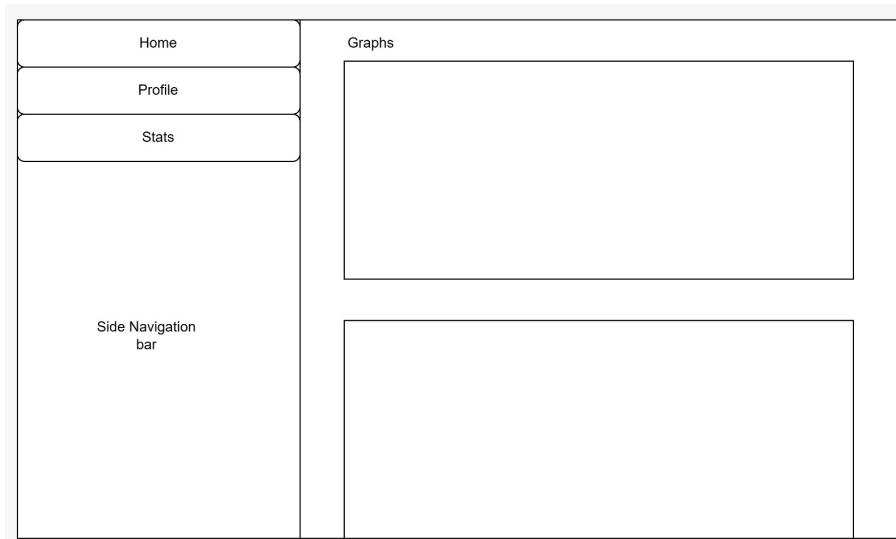


FIGURE 4.11: View Stats Wireframe

4.4 Activity Diagram

This section comprises all the activity diagrams related to this project

4.4.1 Admin-side Diagram

Figure 4.12 displays Activity Diagram of Admin side.

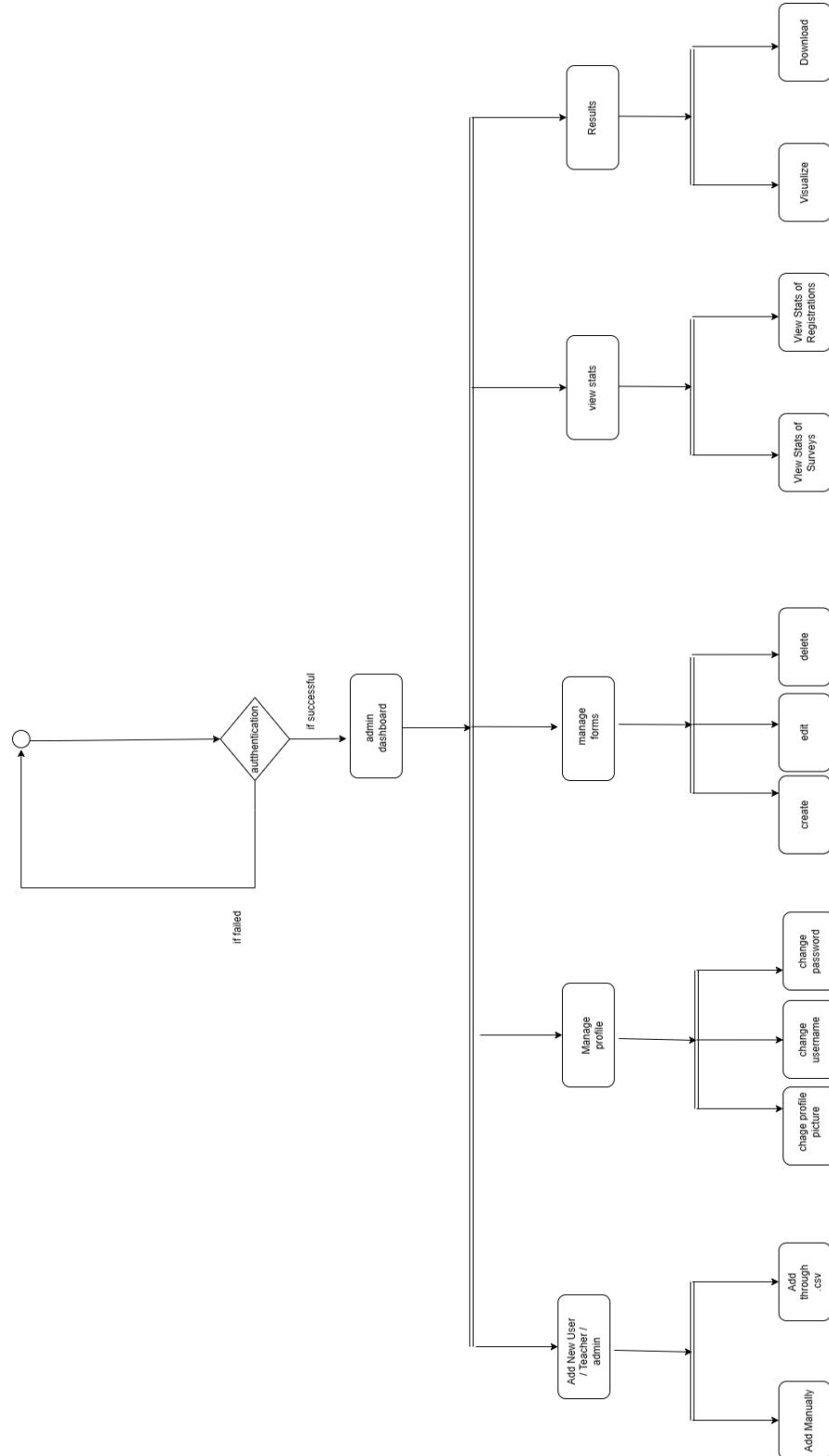


FIGURE 4.12: Admin-side Diagram

4.4.2 User-side Diagram

Figure 4.13 displays Activity Diagram of User side.

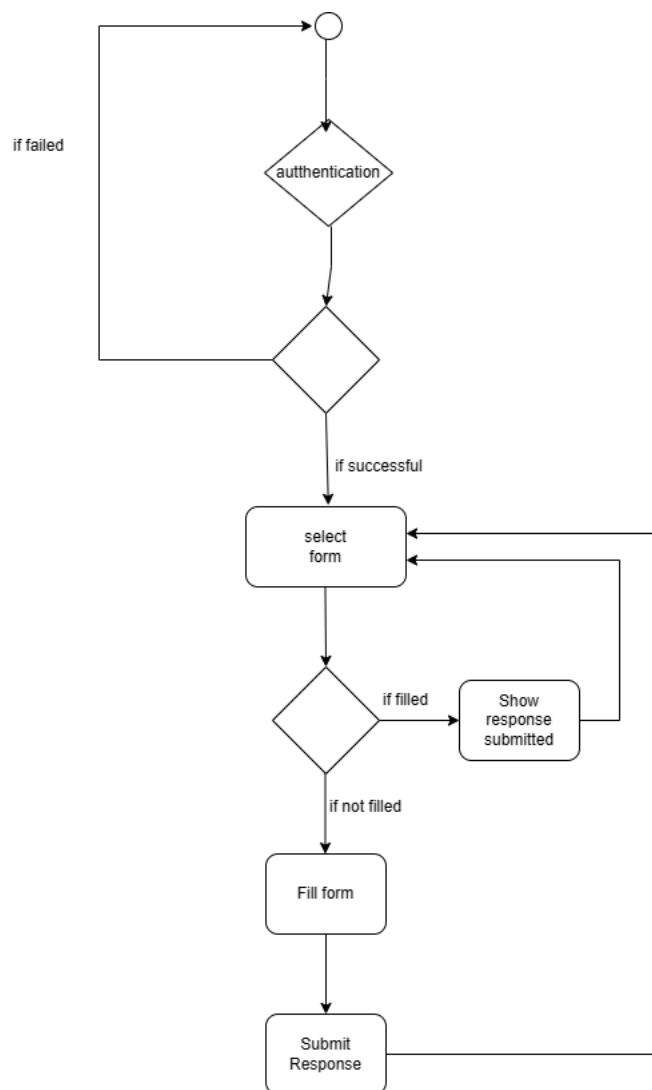


FIGURE 4.13: User-side Diagram

4.5 State Diagram

This section displays all the state diagrams related to this project

4.5.1 Admin-side Diagram

Figure 4.14 displays State Diagram of Admin side.

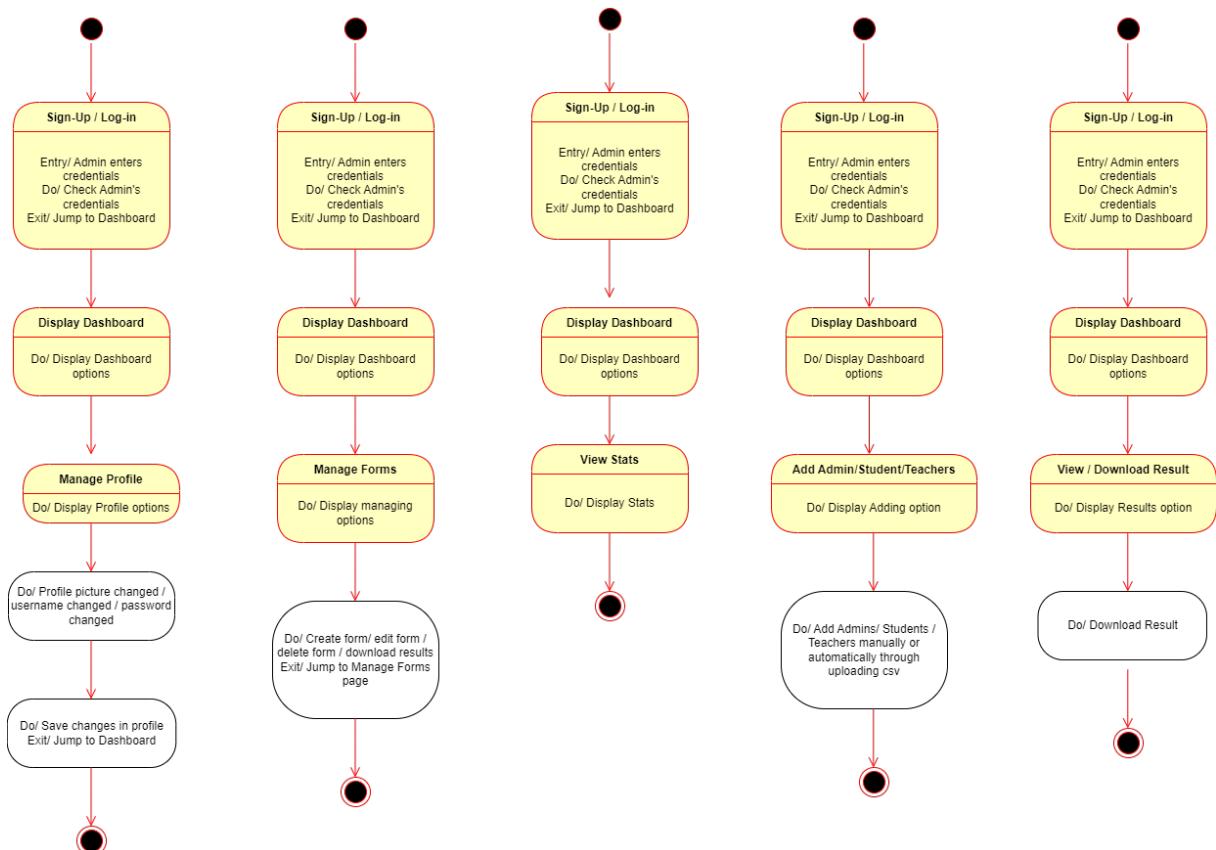


FIGURE 4.14: Admin-side State Diagram

4.5.2 User-side Diagram

Figure 4.15 displays State diagram of User side.

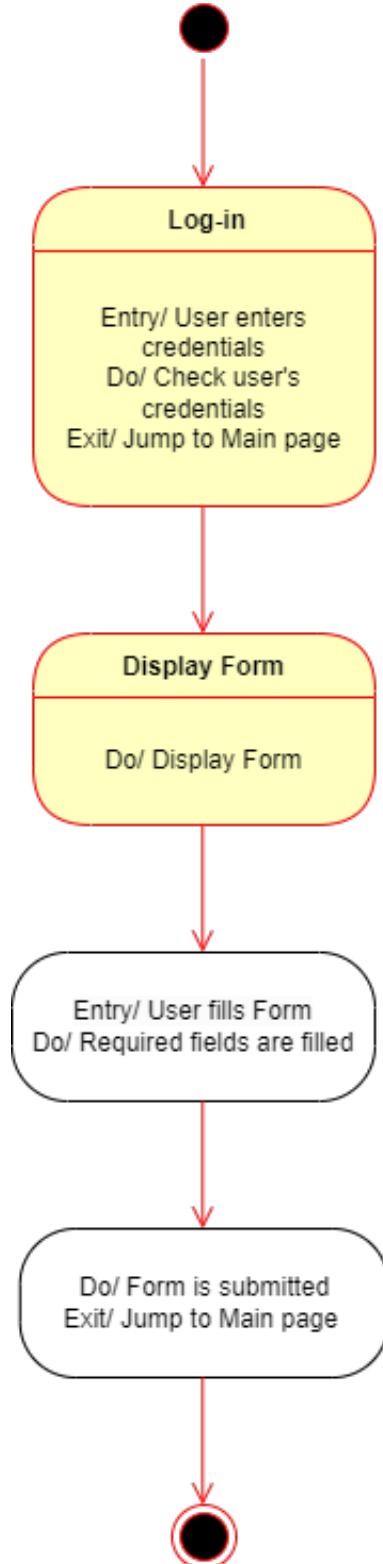


FIGURE 4.15: User-side State Diagram

4.6 Sequence Diagram

This section has all the sequence diagrams related to this project.

4.6.1 Login

The sequence followed by the admin in order to login is displayed in diagram 4.16

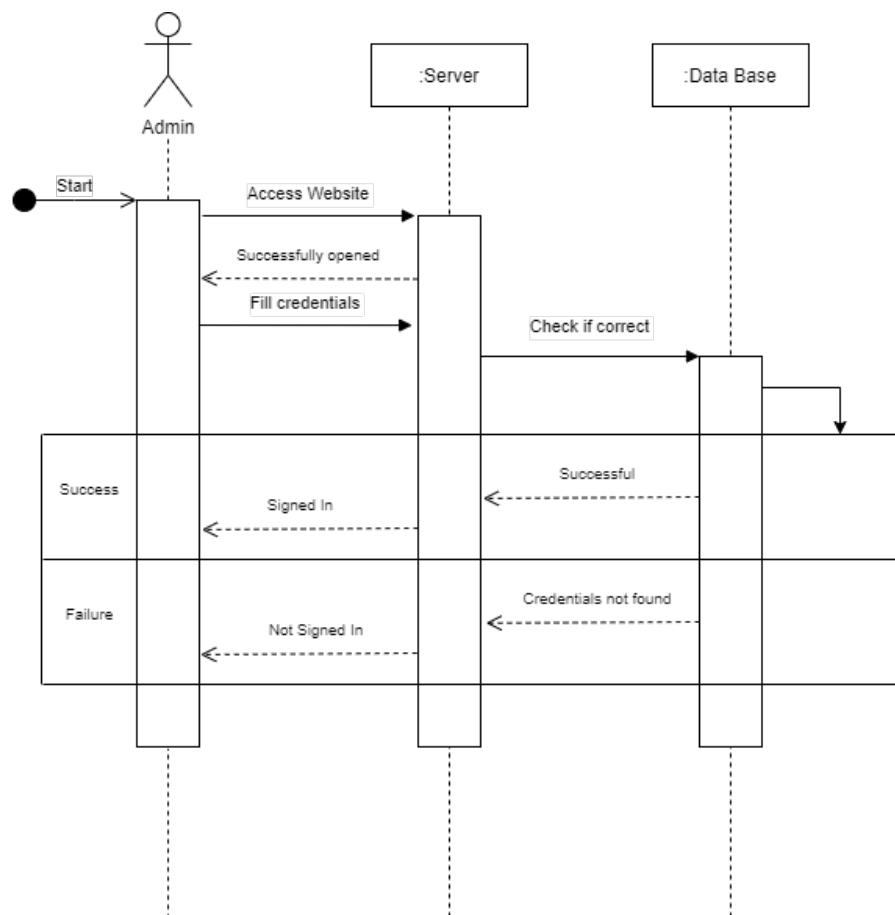


FIGURE 4.16: Login Sequence Diagram

4.6.2 Fill Form

The sequence followed by the user in order to fill a form is displayed in a diagram, refer to figure 4.17 for details.

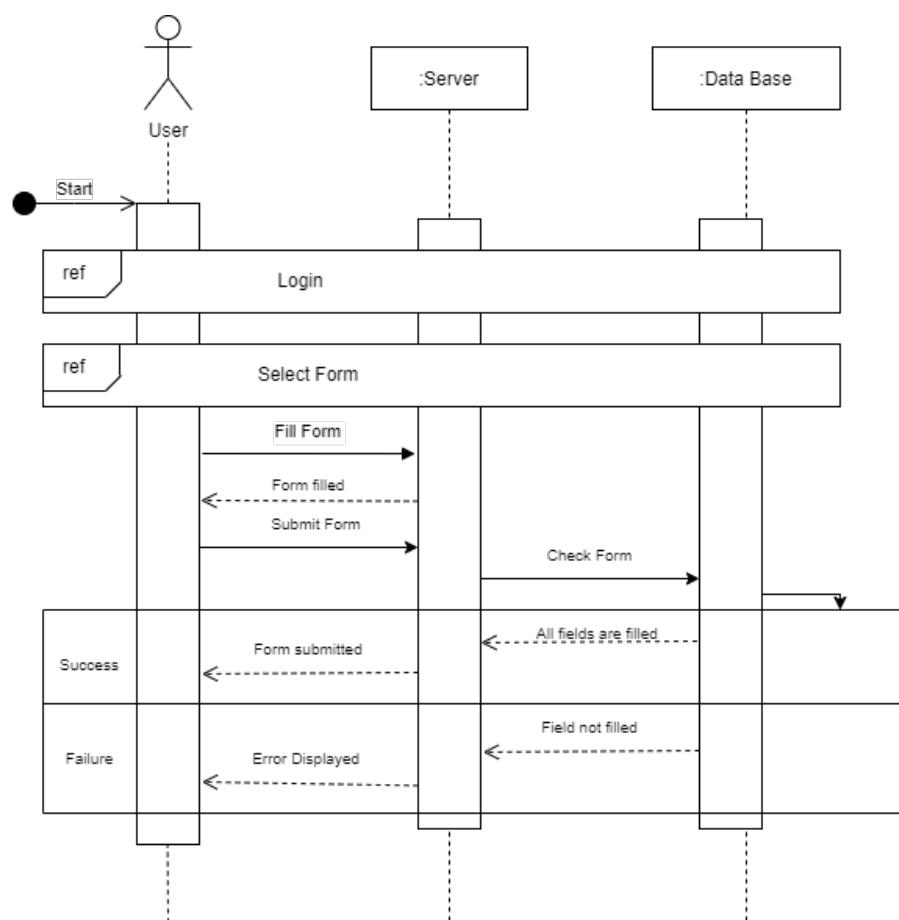


FIGURE 4.17: Fill Form Sequence Diagram

4.6.3 Add Admins

The sequence followed by the Admin in order to add more admins is displayed in a diagram, refer to figure 4.18 for details.

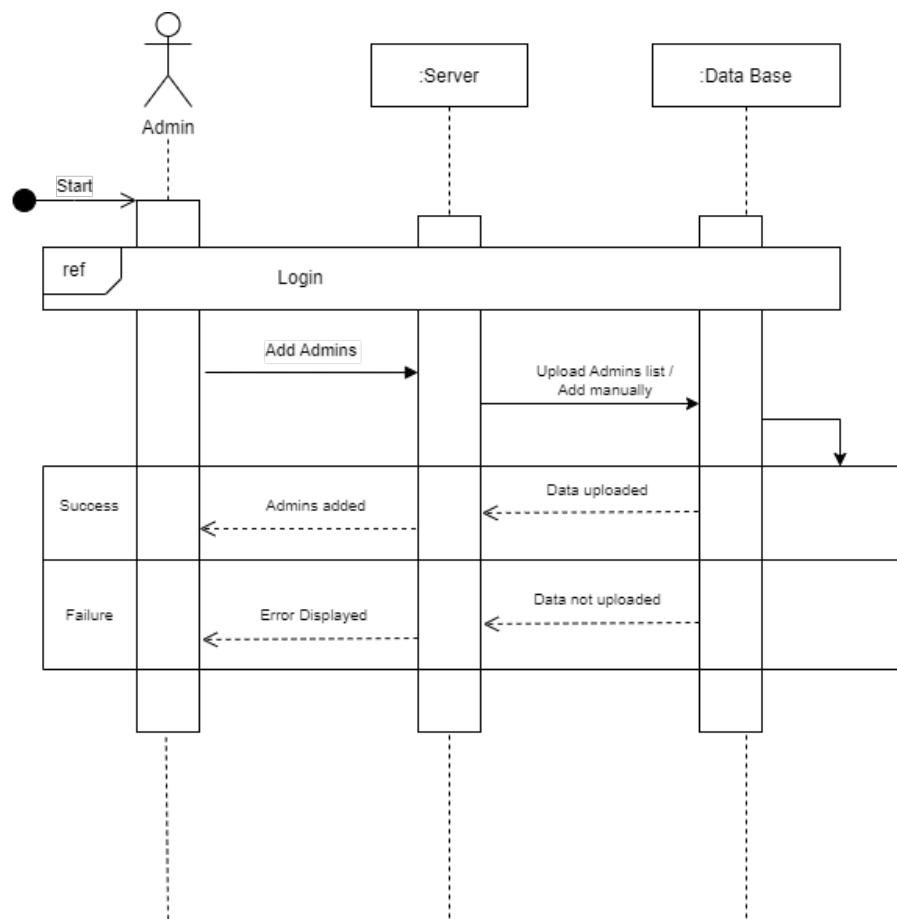


FIGURE 4.18: Add Admins Sequence Diagram

4.6.4 Add Teachers

To add new teachers to the system, admin must follow the sequence displayed in figure 4.19

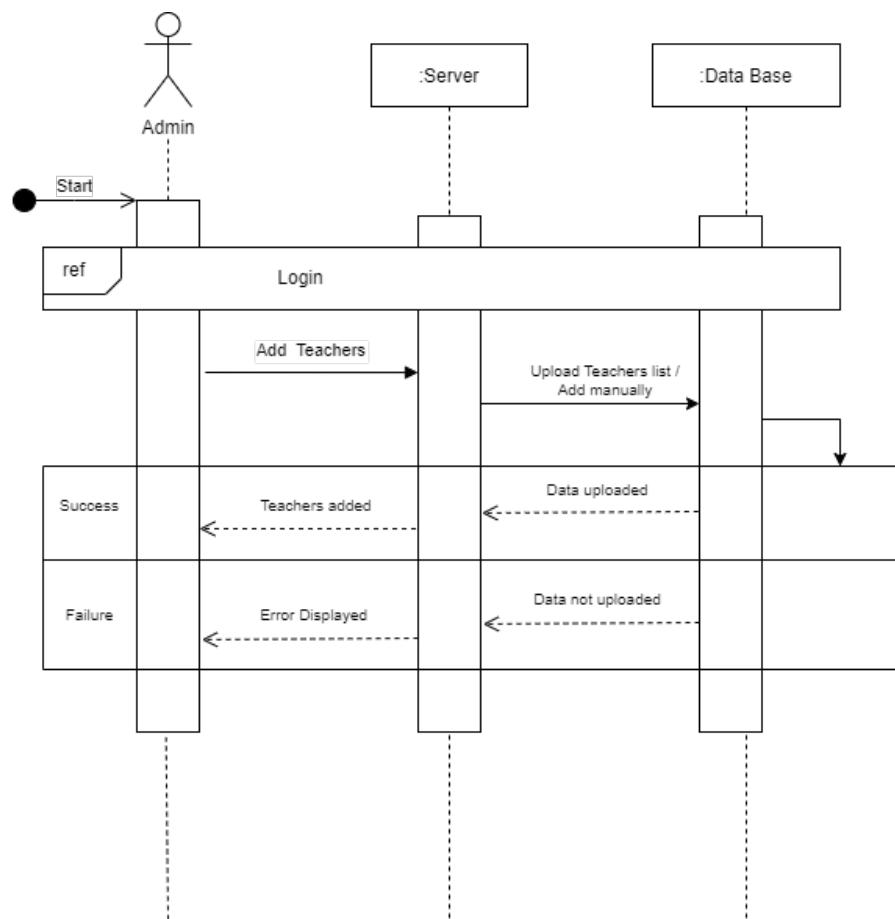


FIGURE 4.19: Add Teachers Sequence Diagram

4.6.5 Add Students

To add new students to the system, admin must follow the sequence displayed in figure 4.20

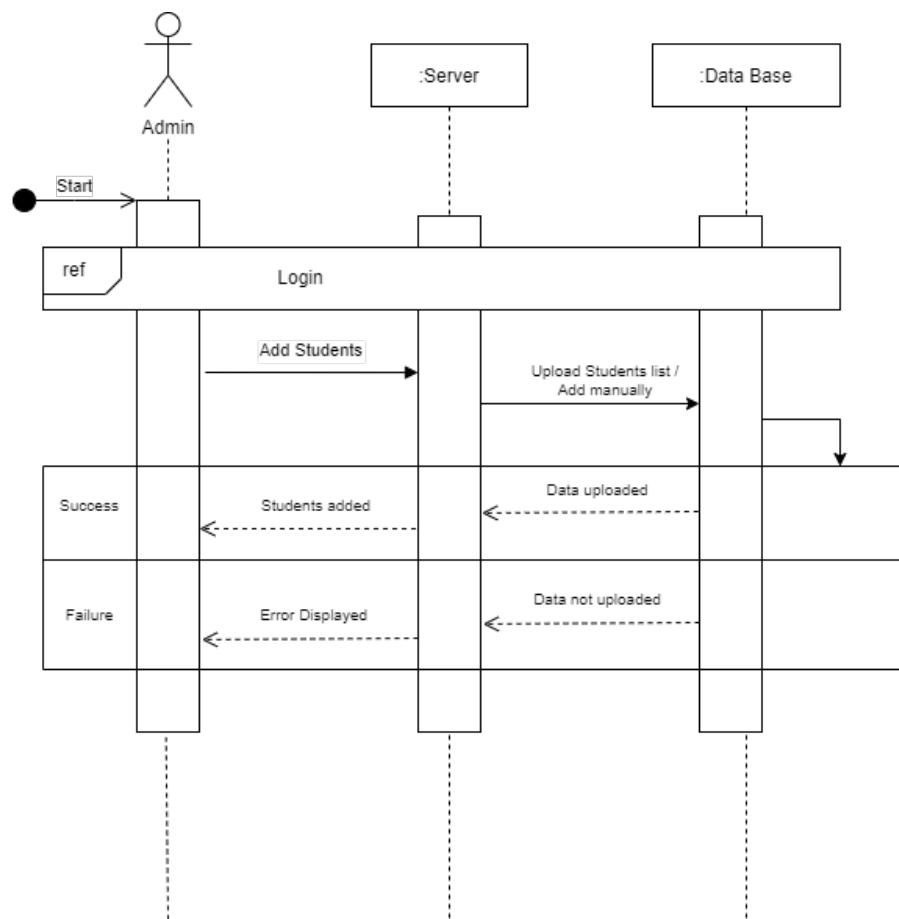


FIGURE 4.20: Add Students Sequence Diagram

4.6.6 Create Form

Admins can create a new form by following the sequence displayed in figure 4.21

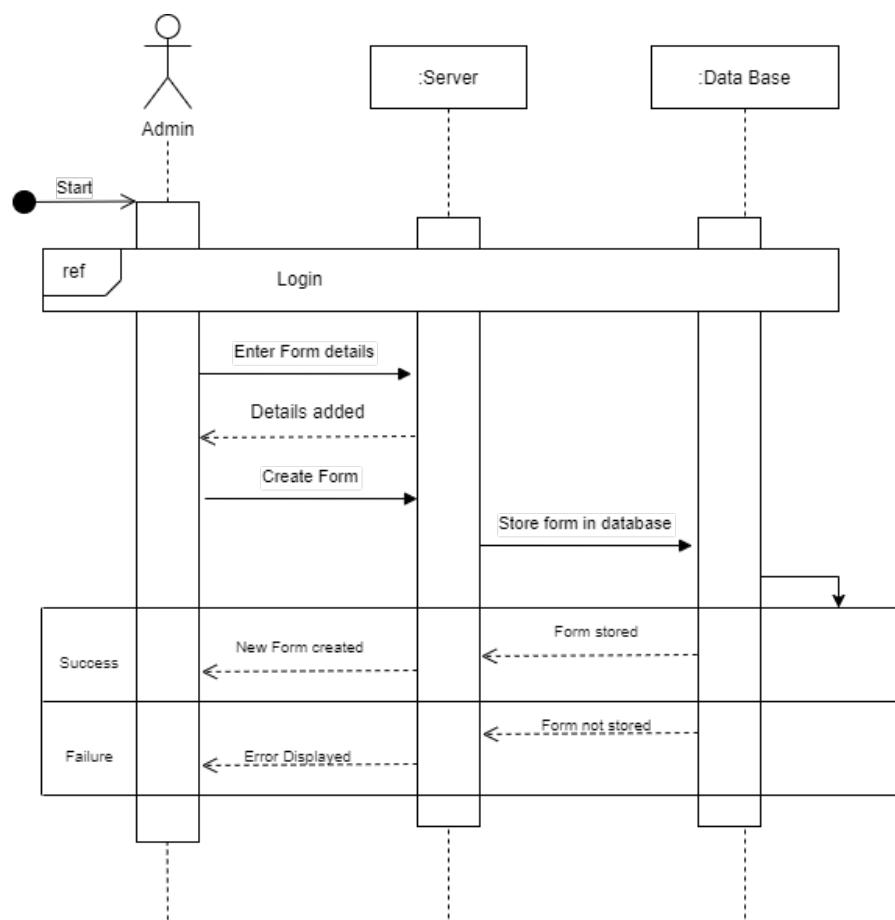


FIGURE 4.21: Create Form Sequence Diagram

4.6.7 Delete Form

Sequence Diagram for deleting a form is displayed in figure 4.22

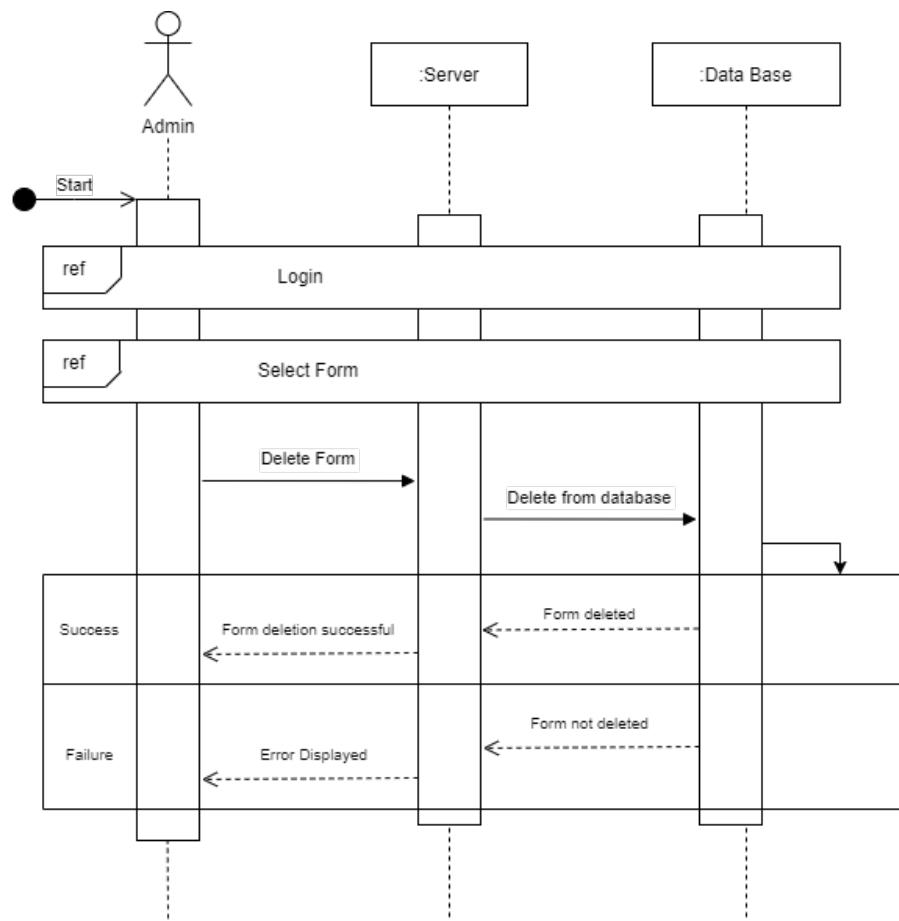


FIGURE 4.22: Delete Form Sequence Diagram

4.6.8 Edit Form

Sequence Diagram for deleting a form is displayed in figure 4.23

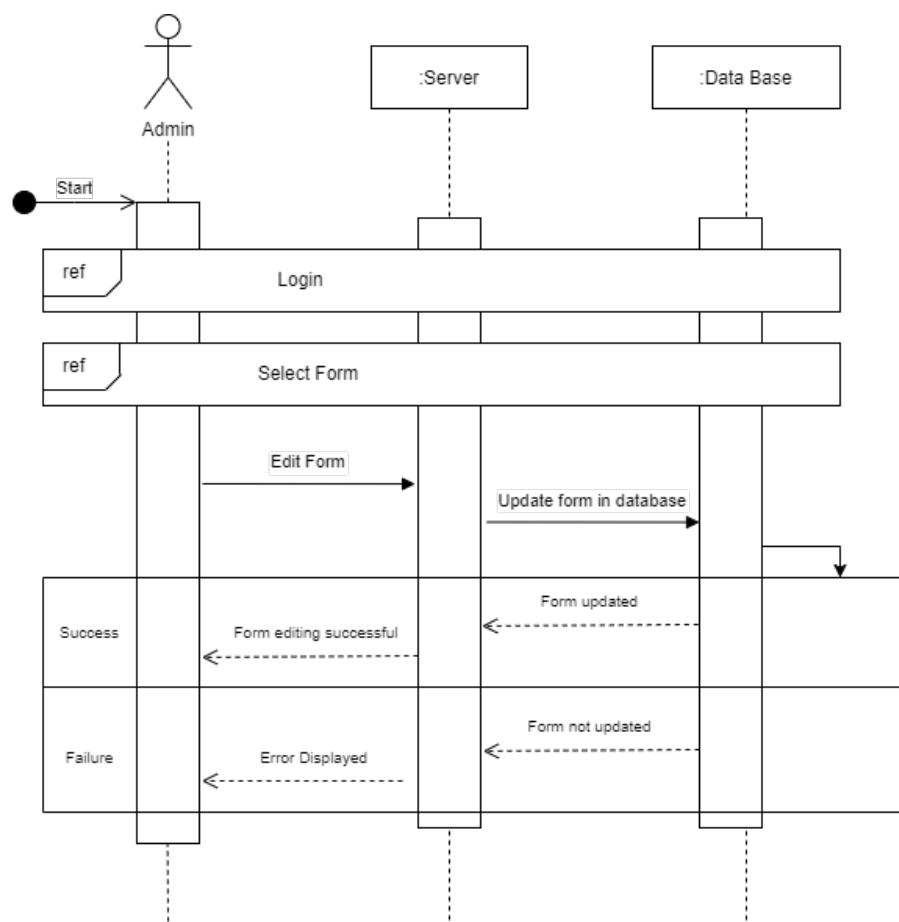


FIGURE 4.23: Edit Form Sequence Diagram

4.6.9 Manage Profile

Admins can manage their profile by following the sequence displayed in figure 4.24

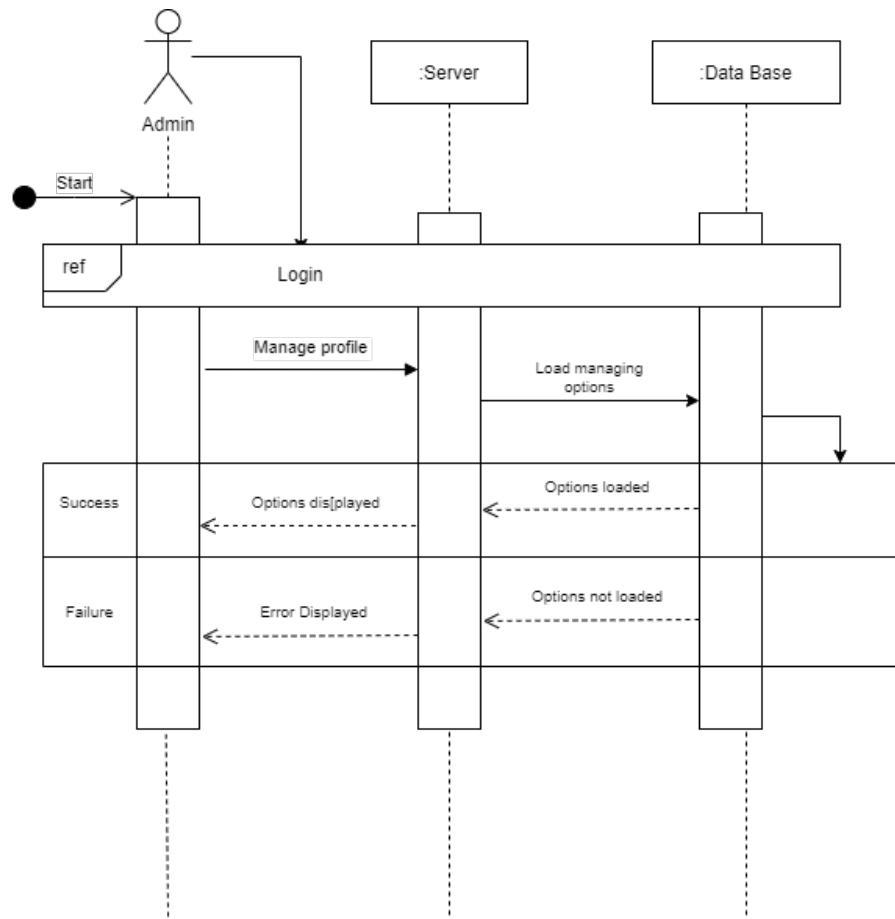


FIGURE 4.24: Manage Profile Sequence Diagram

4.6.10 Edit Username

Admins can update their usernames by following the sequence displayed in figure 4.25

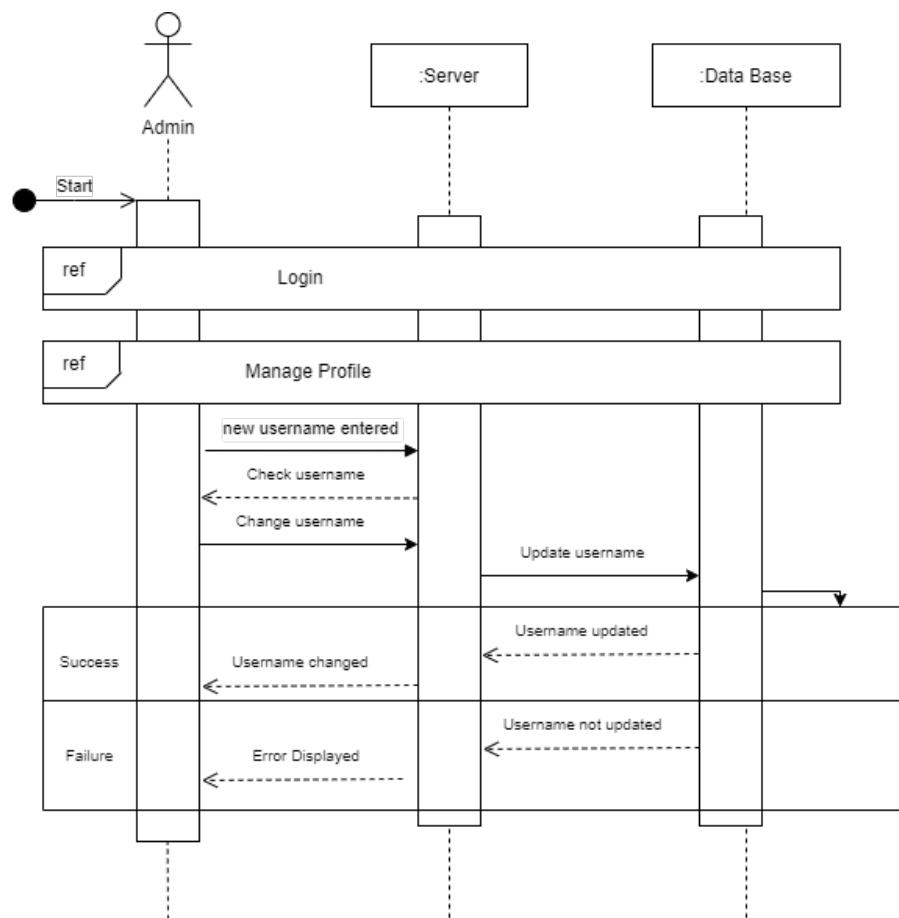


FIGURE 4.25: Edit Username Sequence Diagram

4.6.11 Edit Password

Admins can update their passwords by following the sequence displayed in figure 4.26

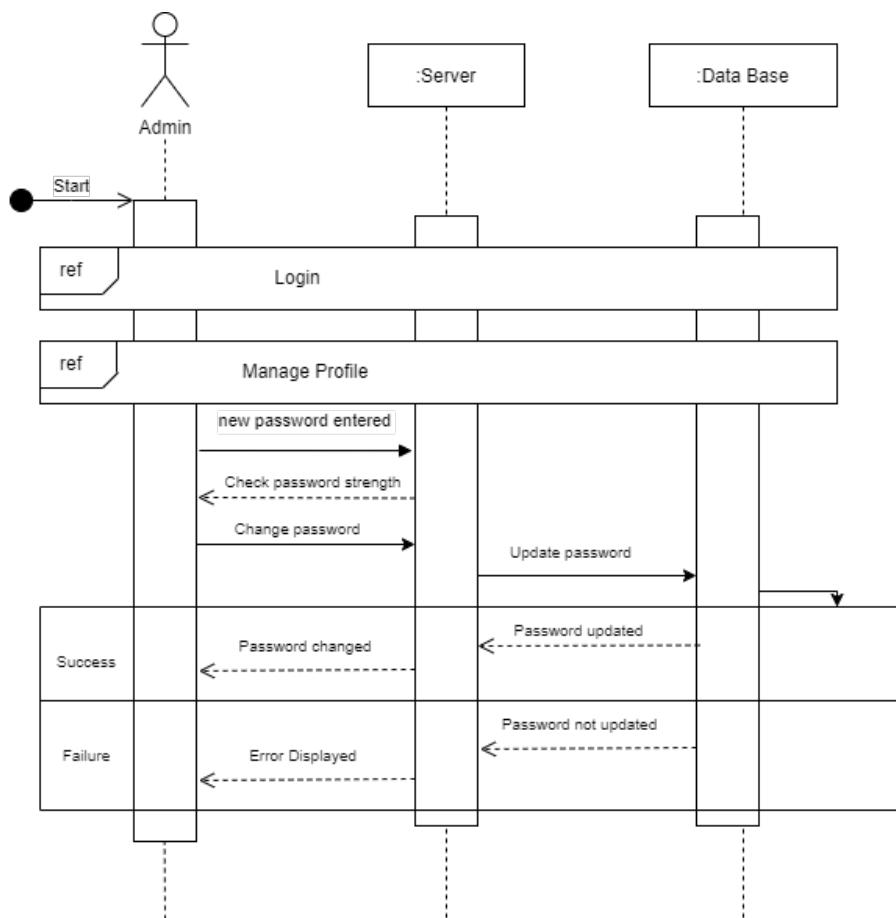


FIGURE 4.26: Edit Password Sequence Diagram

4.6.12 View Stats

To view the stats of forms admins follow the sequence displayed in figure 4.27

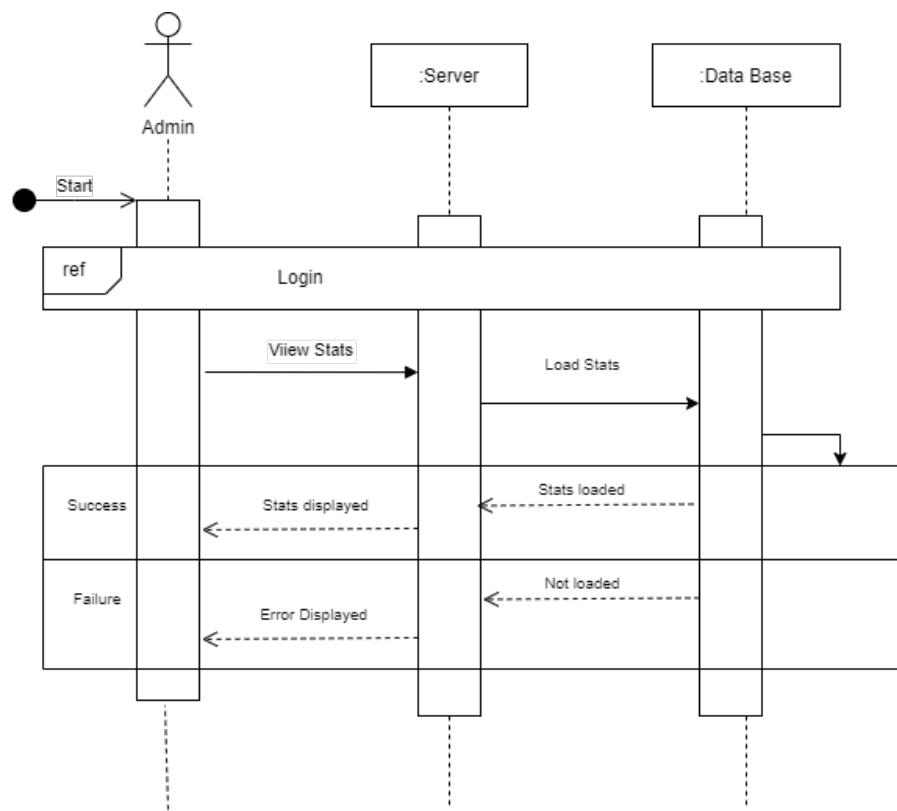


FIGURE 4.27: View Stats Sequence Diagram

4.6.13 Download result

To download the results of forms admins follow the sequence displayed in figure 4.28

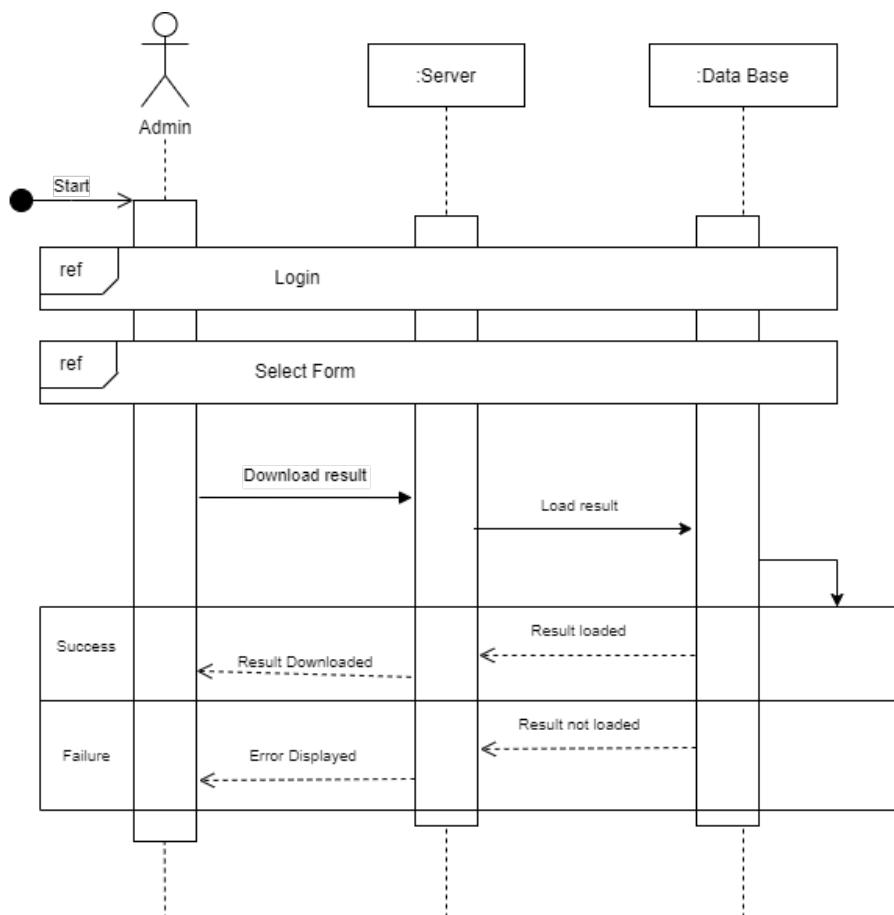


FIGURE 4.28: Download result Sequence Diagram

4.7 Collaboration Diagram

Collaboration Diagrams of system are displayed below.

4.7.1 User-side Diagram

Collaboration diagram of user side that shows collaboration of user in system i.e. User will receive mails for forms or surveys and fill them and submit them,referred by Figure 4.29.

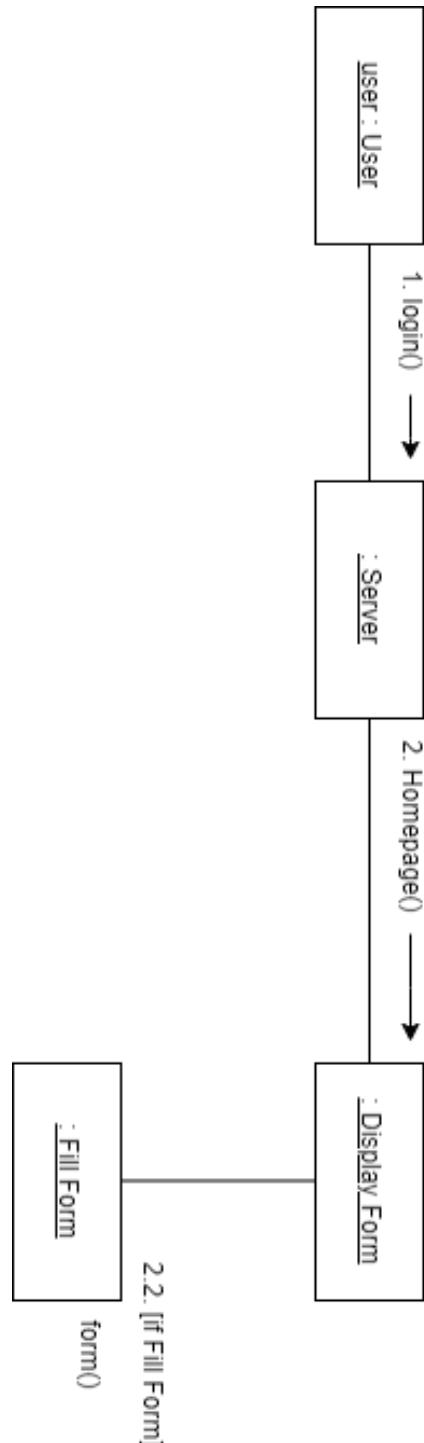


FIGURE 4.29: User-side Collaboration Diagram

4.7.2 Admin-side Diagram

Collaboration diagram of Admin side showing what is collaboration of Admin in project i.e. Admin will add other Admins,Teachers,Students to system and send them created forms or surveys which will be stored and retrieved through database.Moreover Admin can edit profile and see results of forms filled by user,referred by Figure 4.30.

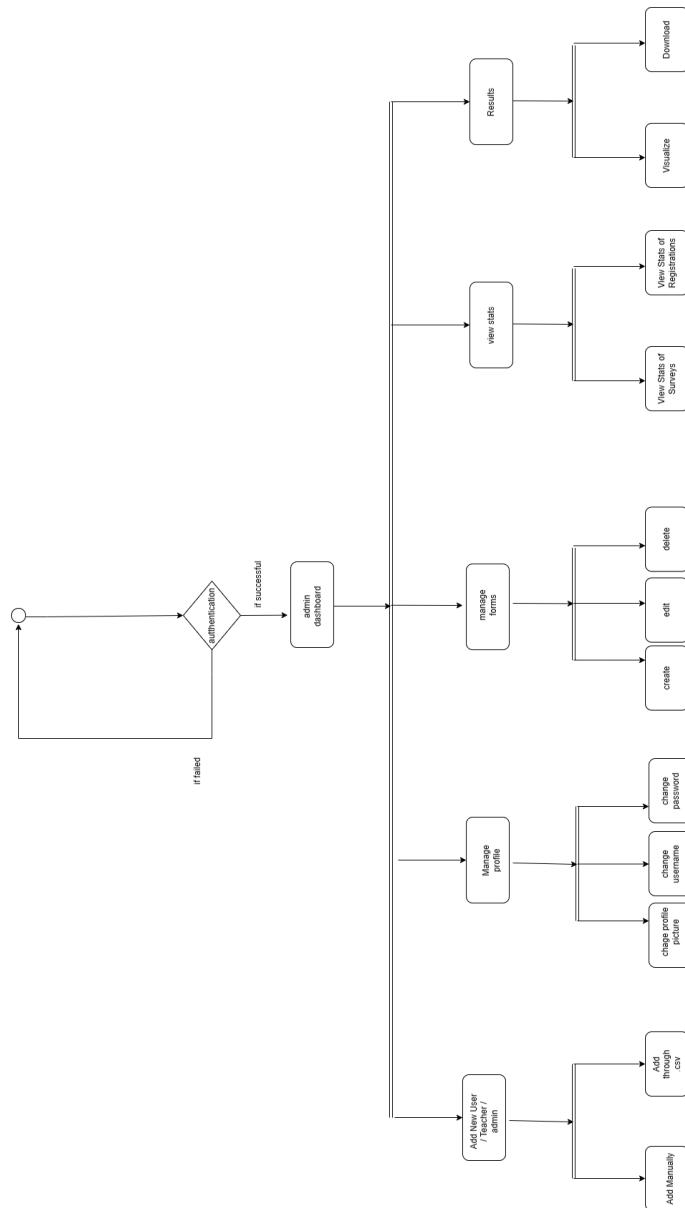


FIGURE 4.30: Admin-side Collaboration Diagram

4.8 Class Diagram

We are using react hooks[13] instead of conventional classes in our project. Refer Table 4.31 for Class Diagram.

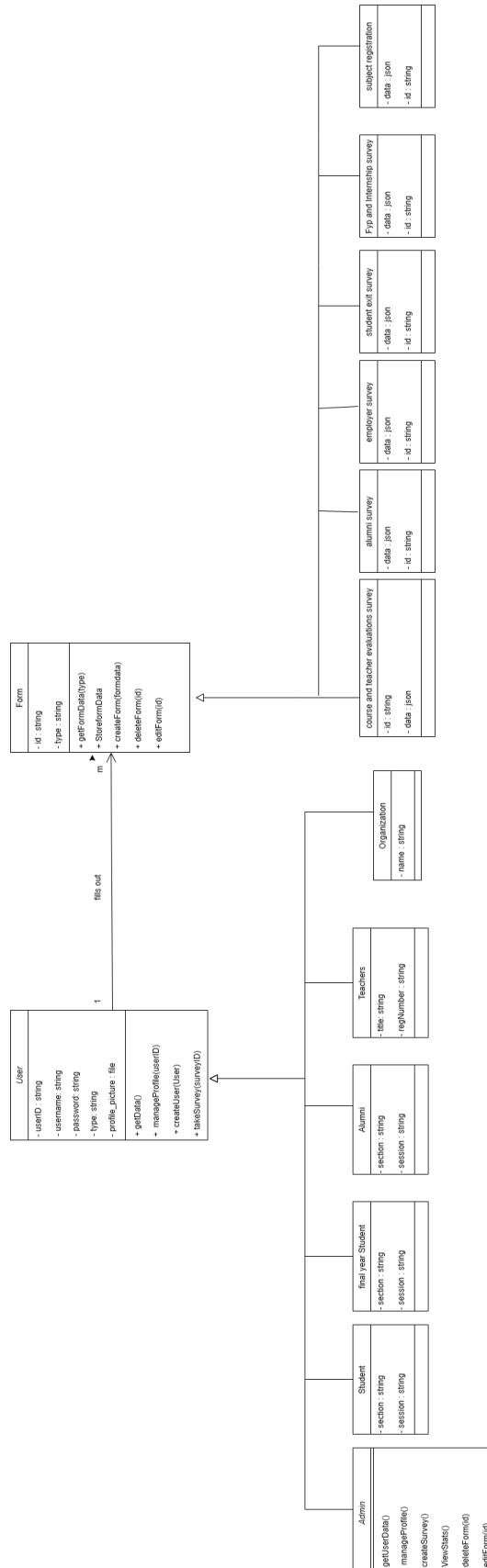


FIGURE 4.31: Class Diagram

4.9 Component Diagram

Components of system that include User,Profile,Form,Admin,Results and Stats.The component of Profile depends on Admin ,each Admin has its own profile. Admin provides Forms to Users that are filled by them.Admin requires Stats and Results for mapping purposes and there is a dependency of results on Forms as each form fill display Different results.User is providing results to Admin by filling form,referred by Figure 4.32.

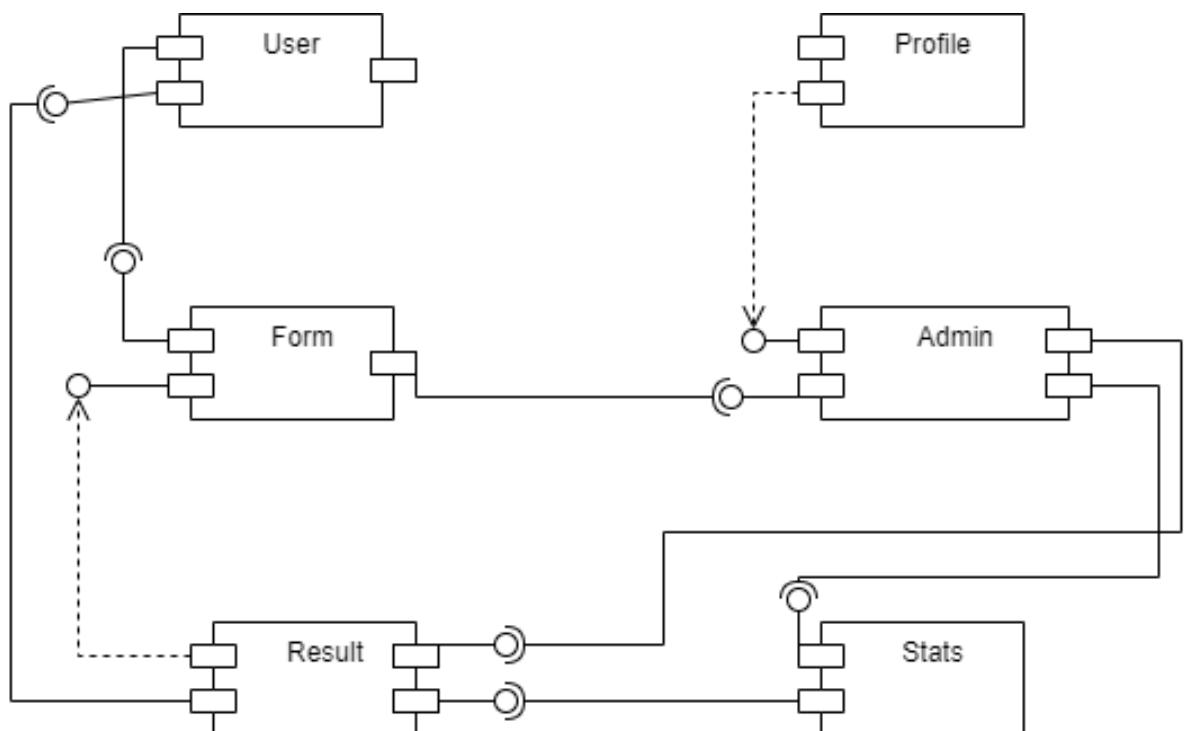


FIGURE 4.32: Component Diagram

4.10 Deployment Diagram

Deployment diagram in which multiple users can access website through internet. Front end and back end of the website is deployed on separate servers, and so is the database. Users can communicate with data through front end which send and receive information through REST APIs and Cookies while back end builds a connection with database to send and receive data, referred by Figure 4.33.

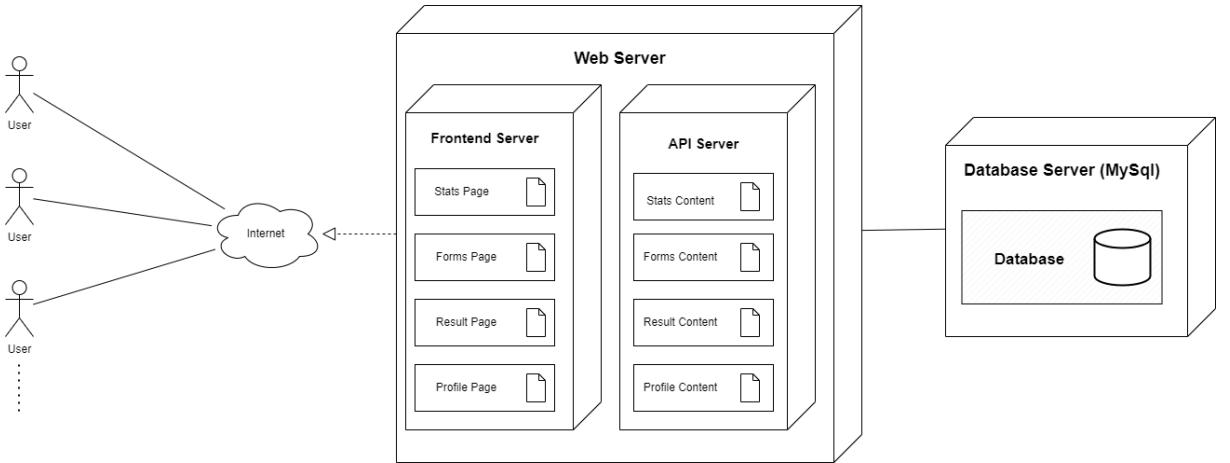


FIGURE 4.33: Deployment Diagram

4.11 ER Diagram

Entity relationship diagram of the website displays the relation between three main entities of the system User, Admin and Forms. Admin has all the right, admin can create users and forms, and also view and edit forms, delete forms, view both in tabular and visual representation, and download results. While the users have one sole purpose that is to fill the forms, referred by figure 4.34.

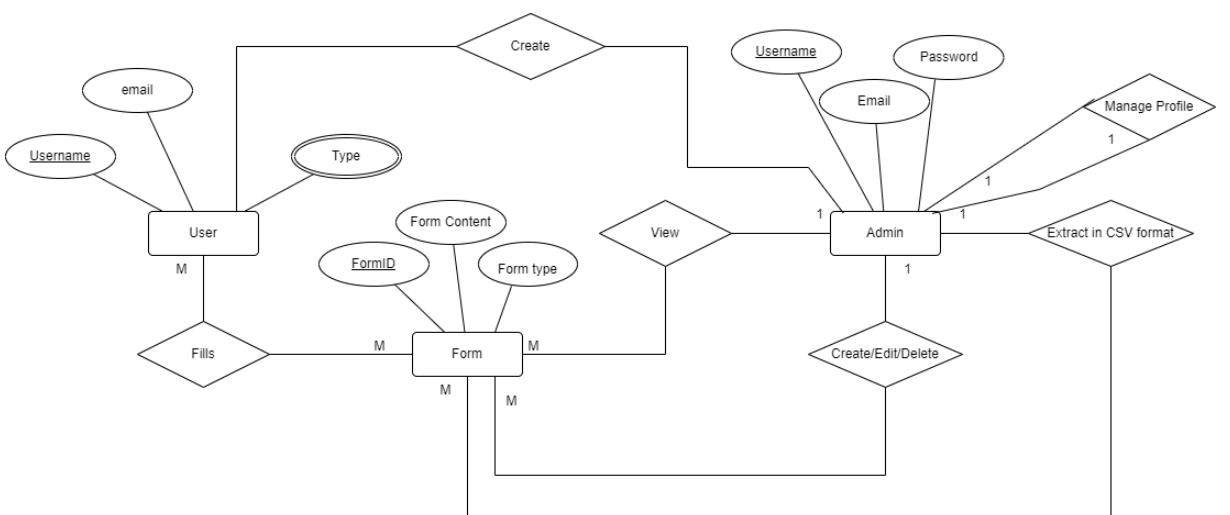


FIGURE 4.34: ER Diagram

4.12 Software Architecture

Architecture diagram displays that Admin will be logged in when he/she is authorized ,Admin can then store list of Admins,Students or Teachers in database through uploading a csv file or entering manually, that can be retrieved later on. Admin can set category of students and send them forms after creating forms and sending reminders to selected category.Users will receive a mail with links to forms and they can fill forms from those links.Data of filled form will be available to Admin and Admin can download csv of results from system.Admin can further manage profile and both Admin and User can change mode to dark mode ,working is demonstrated by Figure 4.35.

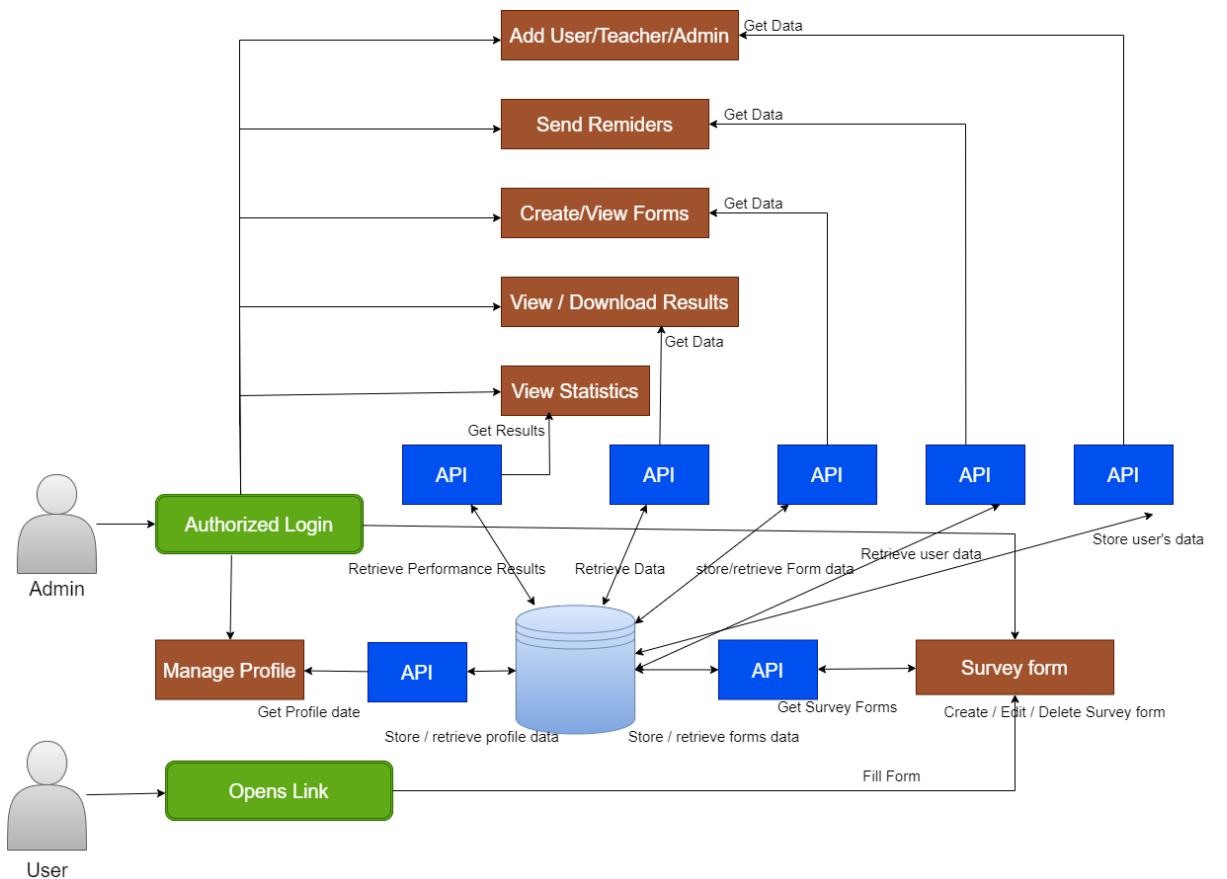


FIGURE 4.35: ER Diagram

Chapter 5

System Implementation

5.1 System Implementation

System Implementation and Requirements are given below.

5.1.1 Software Requirements

Software Requirements of system are referred by Table 5.1

TABLE 5.1: Software Requirements

Purpose	Technology
IDE	Visual Studio Code for development
Database	MySQL
REST API	Express
Front-end	React.js
Back-end	Node.js
Visual representations of data/Charts	Chart.js
Online Database	Clever Cloud
Documentation	Latex
Language	JavaScript, XML, CSS, HTML
Wire-frames/Diagrams	Draw.io
Front-end Hosting	Vercel
Back-end Hosting	Glitch
Version Control System	GitHub

5.1.2 Hardware Requirements

Hardware Requirements of system are referred by Table 5.2

TABLE 5.2: Hardware Requirements

Purpose	Technology
Operating System	64-bit Operating system
Windows	Windows 11
RAM	8GB RAM
Processor	Intel(R) Core(TM) i5-6300U CPU @ 2.40 GHz

5.1.3 Sign-in

Super Admin will be able to add more Admins. These added Admins will be able to sign in and perform all tasks that super Admin can perform but they can not remove other admins or Super Admin. They can Add/edit/remove user, Admin, Teacher manually or through uploading .csv, sending Reminders, viewing and downloading Results file etc., But to access the system with administrative rights, admin has to sign-in first.

Refer Figure 5.1 for Code Snippet of Token Generation.

```
const query = "SELECT * FROM admin WHERE LOWER(email) = LOWER(?);";
const [user] = await db.query(query, [email]);

if (!user || user[0].password === undefined || user[0].password === null) {
  return res.status(401).send({ error: "Invalid credentials" });
}

const isPasswordMatch = await bcrypt.compare(password, user[0].password);

if (!isPasswordMatch) {
  return res.status(401).send({ error: "Invalid password" });
}

const token = jwt.sign(
  [
    {
      user_id: user[0].id,
      user_name: user[0].name,
      user_email: user[0].email,
    },
  ],
  process.env.JWT_SECRET
);
```

FIGURE 5.1: Code Snippet for Token Generation

Refer Figure 5.2 for Code Snippet of Token Returning.

```
    return res
      .status(200)
      .cookie("token", token, {
        path: "/",
        httpOnly: true,
        sameSite: "None",
        secure: true,
      })
      .send({
        message: "Authentication successful",
        user_name: user[0].name,
        user_email: user[0].email,
      });
    } catch (error) {
      console.error("Error:", error);
      return res.status(500).send({ error: "Error during authentication" });
    }
  );
}
```

FIGURE 5.2: Code Snippet for Token Returning

Refer Figure 5.3 for Code Snippet of Token Checking.

```
router.get('/', async (req, res) => {
  try {
    if (!req.cookies.token) {
      return res.status(401).json({ error: 'Authentication failed' });
    }
    const token = req.cookies.token;
    const decoded = jwt.verify(token, process.env.JWT_SECRET);

    res.json({ user_name: decoded.user_name, user_email: decoded.user_email });
  } catch (error) {
    console.error('Error checking authentication:', error);
    res.status(500).json({ error: 'Error checking authentication' });
  }
});
```

FIGURE 5.3: Code Snippet for Token Checking

5.1.4 Create Form and Survey

Admin can Create Subject registration form with specific number of students and specific number of subjects to choose for each student.Teacher Evaluation Survey can also be created and each form and surveys will have a name . Admin will then be able to select session and section for each form and survey.Created Survey or form can be edited or deleted later on by Admin.

Refer Figure 5.4 for Code Snippet of Add Form.

```
try {
  const insertFormQuery = 'INSERT INTO forms (formTitle, category, teacherMapping) VALUES (?, ?, ?)';
  const formResult = await db.query(insertFormQuery, [formTitle, category, teacherMapping]);
  const formId = formResult[0].insertId;

  for (const title of titles) {
    const titleValue = title.title;
    const questions = title.questions;
    const insertTitleQuery = 'INSERT INTO titles (title, form_id) VALUES (?, ?)';
    const titleResult = await db.query(insertTitleQuery, [titleValue, formId]);
    const titleId = titleResult[0].insertId;

    for (const question of questions) {
      const questionText = question.text;
      const responseType = question.ResponseType;
      const responseTable = responseType === 'text' ? 'TextResponses' : 'RatingResponses';

      const insertQuestionQuery = 'INSERT INTO questions (question, title_id, responseType, form_id, response_table) VALUES (?, ?, ?, ?, ?)';
      await db.query(insertQuestionQuery, [questionText, titleId, responseType, formId, responseTable]);
    }
  }
}
```

FIGURE 5.4: Code Snippet for Add Form

5.1.5 Edit Form

Admin can edit form but admins are not allowed to manipulate the forms after receiving responses since that can alter results ,Refer Figure 5.5 for Code Snippet of Edit Form.

```
const ratingResponse = await db.query(
  "SELECT * FROM ratingresponses WHERE form_id = ?",
  [form_id]
);
const textResponse = await db.query(
  "SELECT * FROM textresponses WHERE form_id = ?",
  [form_id]
);

if (ratingResponse[0].length > 0 || textResponse[0].length > 0) {
  // If responses exist for the form, send 401 Unauthorized
  return res.status(401).send({ error: "Responses exist for this form. Deletion not allowed." });
}

// If no responses exist, proceed with editing the form and related entries
const formdata = req.body; // Assuming the form data is sent in the request body

// Iterate through the formdata titles and questions to update the database
for (const title of formdata.titles) {
  for (const question of title.questions) {
    // Update the question in the database using the questions_id
    await db.query(
      "UPDATE questions SET question = ? WHERE id = ?",
      [question.question, question.questions_id]
    );
  }
}
```

FIGURE 5.5: Code Snippet for Edit Form

5.1.6 Delete Form

Admin can edit form but admins are not allowed to manipulate the forms after receiving responses since that will cause problems in result formation Refer Figure 5.6 for Code Snippet of Delete Form.

```

const ratingResponse = await db.query(
  "SELECT * FROM ratingresponses WHERE form_id = ?",
  [form_id]
);
const textResponse = await db.query(
  "SELECT * FROM textresponses WHERE form_id = ?",
  [form_id]
);

if (ratingResponse[0].length > 0 || textResponse[0].length > 0) {
  // If responses exist for the form, send 401 Unauthorized
  return res.status(401).send({ error: "Responses exist for this form. Deletion not allowed." });
}

// If no responses exist, proceed with deleting the form and related entries
await db.query("DELETE FROM forms WHERE id = ?", [form_id]);
await db.query("DELETE FROM titles WHERE form_id = ?", [form_id]);
await db.query("DELETE FROM questions WHERE form_id = ?", [form_id]);

// Optionally, you can send a success message if the deletion is successful
res.status(200).send({ message: "Form deleted successfully." });

```

FIGURE 5.6: Code Snippet for Delete Form

5.1.7 Add User (Backend Implementation)

Admins have to add users that are supposed to receive form links through emails, according to their specific details. The back-end implementation of this particular module is given in figure 5.7.

```

try {
  await Promise.all(
    userDetails.map(async (user) => {
      const query =
        "INSERT INTO users (Name, Email, Category, Semester, Section, endOfSemester) VALUES (?, ?, ?, ?, ?, ?)";
      await db.query(query, [
        user.Name,
        user.Email,
        user.Category,
        user.Semester,
        user.Section,
        null
      ]);

      const mailOption = {
        from: "Quality Enhancement Cell, Department of Computer Science, UET NewCampus <moizarif100@gmail.com>",
        to: user.Email,
        subject: "Registration in CQI System",
        html: `<h1>Verification</h1>
          <p>Dear ${user.Name}, You have been registered successfully in the system, anytime your feedback is required by the university</p>`;
      };
      transporter.sendMail(mailOption, (error, info) => {
        if (error) throw error;
        else {
          console.log("Email Sent Successfully");
        }
      });
    });
}

```

FIGURE 5.7: Code Snippet for Add User

5.1.8 Add Admin (Backend Implementation)

Admins can add other admins that can help in making the work faster and divide the workload, refer to figure 5.8 implementation details of this module.

```
const generateRandomPassword = () => {
  const length = 8;
  const charset = "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789";
  let password = "";
  for (let i = 0; i < length; i++) {
    const randomIndex = Math.floor(Math.random() * charset.length);
    password += charset[randomIndex];
  }
  return password;
};

try {
  const dataWithHashedPasswords = jsonData.map((user) => {
    const password = generateRandomPassword();
    const hashedPassword = bcrypt.hashSync(password, 10);

    return {
      ...user,
      Password: hashedPassword,
      PlainPassword: password,
    };
  });
}
```

FIGURE 5.8: Code Snippet for Add Admin

5.1.9 Add Teacher (Backend Implementation)

Admins have to add teachers for which, students are supposed to fill surveys, the back-end implementation of this particular module is given in figure 5.9.

```
try {
  const jsonData = req.body;
  console.log(jsonData);

  if (!Array.isArray(jsonData) || jsonData.length === 0) {
    return res.status(400).json({ error: "Invalid data format" });
  }

  // Assuming your teachers table has columns id, name, email, section, and semester
  const insertQuery = "INSERT INTO teachers (name, email, section, semester, endOfSemester) VALUES (?, ?, ?, ?, ?)";

  for (const teacher of jsonData) {
    const { Name, Email, Section, Semester } = teacher;

    // Execute the SQL query to insert data into the database
    await db.query(insertQuery, [Name, Email, Section, Semester, null]);
  }

  // You can send a success response if needed
  res.status(200).json({ success: true, message: "Teachers added successfully" });
}
```

FIGURE 5.9: Code Snippet for Add Teacher

5.1.10 Add User, Admin, Teacher manually Front-end

Admin can enter details of Admin and Teacher manually by entering name,email addresses .Admin can add students manually by entering their session ,semester and section.Refer Figure 5.10 for Code Snippet of Add User,Teacher,Admin manually.

```
<div className="addAdminPageContainer">
  <div className="manualcard">
    <div className="marksemesterend">
      <button onClick={handleMarkSemesterEnd}>Mark Semester End</button>
    </div>
    <form>
      <input
        type="text"
        placeholder="Enter New User Name"
        name="name"
        value={newuser.name}
        onChange={handleInputChange}
        required
      />
      <input
        type="email"
        placeholder="Enter New User Email"
        name="email"
        value={newuser.email}
        onChange={handleInputChange}
        required
      />
      <button className="add-btn" onClick={handleAdduser}>
        Add
      </button>
    </form>
  </div>
</div>
```

FIGURE 5.10: Code Snippet for Add User,Teacher,Admin manually

5.1.11 Add User, Admin, Teacher through .csv Front-end

Admin can upload csv file containg Admin ,Teacher or User List with their details in it,refer figure 5.11 for Code Snippet of Add User,Teacher,Admin csv.

```
const parseCSV = (csvContent) => {
  Papa.parse(csvContent, {
    header: true,
    skipEmptyLines: true,
    complete: (parsedData) => {
      const { data } = parsedData;

      if (data && Array.isArray(data) && data.length > 0) {
        const headers = Object.keys(data[0]);

        const newData = data.map((row) => {
          const rowData = {};
          headers.forEach((header) => {
            rowData[header] = row[header];
          });
          return rowData;
        });

        setJsonData((prevData) => [...prevData, ...newData]);

        const userDetailsFromCSV = newData.map((user) => ({
          Name: user.Name,
          Email: user.Email,
          Category: selectedCategory,
          Semester: selectedSemester,
          Section: selectedSection,
        }));
      }
    }
  });
}
```

FIGURE 5.11: Code Snippet for Add User,Teacher,Admin csv

5.1.12 View and download results

Admin will be displayed results of forms filled by User.A pie chart displaying average positive and negative response of a specific form, and a graph displaying the position of Teacher according to results collected from users. Refer Figure 5.12 for Code Snippet of View and Download Results.

```

const downloadCSV = () => {
  const csvContent = "data:text/csv;charset=utf-8," + formatDataToCSV();
  const encodedURI = encodeURI(csvContent);
  const link = document.createElement('a');
  link.setAttribute("href", encodedURI);
  link.setAttribute(
    "download", `Result_${selectedCategory}_${selectedSemester}_${selectedSection}.csv`
  );
  document.body.appendChild(link);
  link.click();
};

const formatDataToCSV = () => {
  let csv = "Questions,";
  uniqueTeachers.forEach((teacher, index) => {
    csv += `${teacher} Disagree,${teacher} Agree${index === uniqueTeachers.length - 1 ? "\n" : ","};`;
  });
  uniqueQuestions.forEach((question, qIndex) => {
    csv += `${question},`;
    uniqueTeachers.forEach((teacher, tIndex) => {
      const correspondingData = resultData.find(item =>
        item.teacher_name === teacher && item.question_text === question
      );
      if (correspondingData) {
        csv += `${correspondingData.averages[0]},${correspondingData.averages[1]}`;
      }
      csv += tIndex === uniqueTeachers.length - 1 ? "\n" : ",";
    });
  });
  return csv;
};

```

FIGURE 5.12: Code Snippet for View and Download Results

5.1.13 Send Reminders

After creating a new form Admin can specify for which category the form is created and then a mail with names of Teachers and Links in front of them will be sent to specific category of selected user. Refer Figure 5.13 for Code Snippet of Send Reminders.

```

async function generateFormLinks(formIds, user) {
  const formLinks = [];
  for (const formIdObj of formIds) {
    const { id: formId, teacherMapping } = formIdObj;
    let formLink = '';
    if (teacherMapping === 1) {
      const teacherDetails = await getTeacherDetails(user.Semester, user.Section);
      for (const teacher of teacherDetails) {
        formLink = `${teacher.teacherName}: ${process.env.FRONT_END_URL}/form/${formId}/${user.ID}/${teacher.teacherId}`;
        formLinks.push(formLink);
      }
    } else {
      formLink = `${process.env.FRONT_END_URL}/form/${formId}/${user.ID}`;
      formLinks.push(formLink);
    }
  }
  return formLinks;
}
async function getTeacherDetails(semester, section) {
  try {
    const query = 'SELECT id as teacherId, name as teacherName FROM teachers WHERE semester = ? AND section = ?';
    const [teacherDetails] = await db.query(query, [semester, section]);
    return teacherDetails;
  } catch (error) {
    console.error('Error getting teacher details:', error);
    throw error;
  }
}

```

FIGURE 5.13: Code Snippet for Send Reminders

5.1.14 Edit Profile

Admin can Edit Username and password. After admin is created he can change his default password or username sent by Super Admin , Refer Figure 5.14 for Code Snippet of Edit Profile.

```

const emailRegex = /^[^@\s]+@[^\s]+\.\w+$/;
if (!emailRegex.test>Email)) {
  return res.status(400).send({ error: "Invalid email format" });
}

// Check if the password is at least 8 characters long
if (Password.length < 8) {
  return res.status(400).send({ error: "Password must be at least 8 characters long" });
}

// Encrypt the password before storing it in the database
const hashedPassword = bcrypt.hashSync>Password, 10);

// Send an email to the provided email
const mailOptions = {
  from: process.env.EMAIL_FROM,
  to: Email,
  subject: "Profile Updated",
  text: `Your profile has been updated successfully.
  Name : ${Name}
  Email: ${Email}
  Password:${Password}`,
};
await transporter.sendMail(mailOptions);

// If email sending is successful, update the database with name, email, and encrypted password values
await db.query("UPDATE admin SET Name = ?, Email = ?, Password = ? WHERE email = ?", [Name, Email, hashedPassword, adminEmail]);

```

FIGURE 5.14: Code Snippet for Edit Profile

5.1.15 Change Mode

Admin can change mode of whole system to Dark mode whereas User is displayed only the fill form page therefore, user can change fill form page mode only, Refer Figure 5.15 for Code Snippet of Change Mode.

```
const emailRegex = /^[^@\s]+@[^\s]+\.\[^@\s]+$/;
if (!emailRegex.test(Email)) {
    return res.status(400).send({ error: "Invalid email format" });
}

// Check if the password is at least 8 characters long
if (Password.length < 8) {
    return res.status(400).send({ error: "Password must be at least 8 characters long" });
}

// Encrypt the password before storing it in the database
const hashedPassword = bcrypt.hashSync(Password, 10);

// Send an email to the provided email
const mailOptions = {
    from: process.env.EMAIL_FROM,
    to: Email,
    subject: "Profile Updated",
    text: `Your profile has been updated successfully.
Name : ${Name}
Email: ${Email}
Password:${Password}`
};
await transporter.sendMail(mailOptions);

// If email sending is successful, update the database with name, email, and encrypted password values
await db.query("UPDATE admin SET Name = ?, Email = ?, Password = ? WHERE email = ?", [Name, Email, hashedPassword, adminEmail]);
```

FIGURE 5.15: Code Snippet for Change Mode

Chapter 6

System Testing and Evaluation

The objective of employing various testing methodologies in your development process is to ensure that your software can function in various environments and across different platforms. These methodologies are typically categorized into functional and non-functional testing.

Functional testing involves testing the application against the specified business requirements. It encompasses all test types aimed at ensuring that each component of the software behaves as expected, using the use cases provided by the design team or business analyst. These testing methods are usually conducted sequentially and include:

- Unit testing
- Integration testing
- System testing
- Acceptance testing

Non-functional testing methods focus on the operational aspects of the software. These include:

- Performance testing
- Security testing
- Usability testing
- Compatibility testing

Apply all these testing techniques to your final year project under the guidance of your supervisor and an explanation of all these testing techniques is given below. Write down your test cases in tabular form.

6.1 Unit Testing

Unit testing involves the thorough examination of each unit or individual component of the software application. Serving as the initial stage of functional testing, its primary goal is to validate the performance of these unit components.

6.1.1 Log-in page Testing

This page is used to sign-in ,If Email and Password are correct then Admin is signed in.

- Email Text Field.
- Password Text field.
- Log-in button.

6.1.1.1 Email Text Field

Testing of Email is displayed by Table 6.1.

TABLE 6.1: Email Text Field Testing

Sr No	Test Data	Expected Result	Test Result
1	example@gmail.com	Email Pass	Pass
2	example@mail.com	Email Pass	Pass
3	example@0123gmail.com	Email Fail	Pass
4	example@mail.pk	Email Pass	Pass
5	example@mail.	Email Fail	Pass
6	@mail.com	Email Fail	Pass
7	email does not exist	Email Fail	Fail

6.1.1.2 Password Text field

Password Testing is displayed by Table 6.2.

TABLE 6.2: Password Text Field Testing

Sr No	Test Data	Expected Result	Test Result
1	ABCabc123@	Password must be more than 8 characters	Pass
2	ABCD	Password must have small letters	Fail
3	abcd	Password must have numbers	Fail
4	abc123	Password must have capital letters	Fail
5	ABC123.	Password must have special characters	Fail
6	12@/?	Password must have small and capital letters.	Fail
7	a@A1	Password must be more than 8 characters	Fail

6.1.1.3 Log-in Button

Log-in button testing is displayed by Table 6.3.

TABLE 6.3: Log-in Button Testing

Sr No	Test Data	Expected Result	Test Result
1	Email is Invalid	Enter valid Email	Fail
2	Password is Invalid	Enter valid Password	Fail
3	Email is Empty	Fill Email Field	Fail
4	Password is Empty	Fill Password Field	Fail
5	Email and Password are valid	User must be logged in	Pass
6	Email and Password are empty	Fill Email and Password Fields	Fail
7	Email and Password do not exist	Do not Log in	Fail

6.1.2 Top Nav-bar Testing

This page is used to Log-out, display current admin's Profile including Admin's name and Profile picture and displays UET logo button that acts as go back Button.

- Back Button.
- Profile Display.
- Log-out Button.
- Dark Mode Toggle

6.1.3 Side Nav-bar Testing

Side-Nav displays different options for user that allows Admin to do different tasks.

6.1.2.1 Back Button

Back Button Testing is displayed by Table 6.4.

TABLE 6.4: Back Button Testing

Sr No	Test Data	Expected Result	Test Result
1	Button Clicked	Go back to page	Pass
2	Button not clicked	Stay on same page	Pass

6.1.2.2 Profile Display

Profile Display Testing is shown by Table 6.5.

TABLE 6.5: Profile Display Testing

Sr No	Test Data	Expected Result	Test Result
1	Profile Clicked	Go to Edit Profile Page	Pass
2	Button not clicked	Display Profile	Pass

6.1.2.3 Log-out Button

Log-out Button Testing is displayed by Table 6.6.

TABLE 6.6: Log-out Button Testing

Sr No	Test Data	Expected Result	Test Result
1	Button Clicked	All changings are saved then Log-out	Pass
2	Button clicked	Changings made by Admin are not saved.	Fail
3	Button not clicked	Do not Log-out.	Pass

6.1.2.4 Dark Mode Toggle

Dark Mode Toggle Button Testing is displayed by Table 6.7.

TABLE 6.7: Dark Mode Toggle Button Testing

Sr No	Test Data	Expected Result	Test Result
1	Button Clicked	Mode Changed to Dark Mode	Pass
2	Button Clicked Twice	Mode reverts back to light mode	Pass

- Dashboard Option.
- Surveys Option.
- Users Option.
- Teachers Option.
- Admins Option.
- Reminders Option.

- Results Option.
- Profile Option.

6.1.3.1 Dashboard Option

Refer to Table 6.8 to view Dashboard Display Testing.

TABLE 6.8: Dashboard Button Testing

Sr No	Test Data	Expected Result	Test Result
1	Button Clicked	Display Stats page	Pass
2	Button not clicked	Display Stats page as it is default page	Pass

6.1.3.2 Surveys Option

Survey Options Testing is shown in Table 6.9.

TABLE 6.9: Survey Button Testing

Sr No	Test Data	Expected Result	Test Result
1	Button Clicked	Display Survey page	Pass
2	Button not clicked	Display Stats page as it is default page	Pass

6.1.3.3 Users Option

Refer to Table 6.10 for Student Button Testing.

TABLE 6.10: Students Button Testing

Sr No	Test Data	Expected Result	Test Result
1	Button Clicked	Display Student page	Pass
2	Button not clicked	Display Stats page as it is default page	Pass

6.1.3.4 Teachers Option

For Teacher Button Testing Refer to Table 6.11.

TABLE 6.11: Teachers Button Testing

Sr No	Test Data	Expected Result	Test Result
1	Button Clicked	Display Teachers page	Pass
2	Button not clicked	Display Stats page as it is default page	Pass

6.1.3.5 Admins Option

For Admin Button Testing Refer to Table 6.12.

TABLE 6.12: Admin Button Testing

Sr No	Test Data	Expected Result	Test Result
1	Button Clicked	Display Admins page	Pass
2	Button not clicked	Display Stats page as it is default page	Pass

6.1.3.6 Reminders Option

Refer to Table 6.13 for viewing Reminder Sending Button Testing .

TABLE 6.13: Reminder Button Testing

Sr No	Test Data	Expected Result	Test Result
1	Button Clicked	Display Reminder page	Pass
2	Button not clicked	Display Stats page as it is default page	Pass

6.1.3.7 Results Option

For Results options Testing refer to Table 6.14.

TABLE 6.14: Results Button Testing

Sr No	Test Data	Expected Result	Test Result
1	Button Clicked	Display Results page	Pass
2	Button not clicked	Display Stats page as it is default page	Pass

6.1.3.8 Profile Option

For Profile Button Testing refer Table 6.15.

TABLE 6.15: Profile Button Testing

Sr No	Test Data	Expected Result	Test Result
1	Button Clicked	Display Profile page	Pass
2	Button not clicked	Display Stats page as it is default page	Pass

6.1.4 Admin Dashboard Testing

This page is used to display graph for surveys and subject registration progress.

- Display Survey graphs.
- Display Subject Registration graphs.

6.1.4.1 Display Survey graphs

Refer to Table 6.16 for viewing Testing of Display Survey Graph Component .

TABLE 6.16: Survey Graph Testing

Sr No	Test Data	Expected Result	Test Result
1	Hover over graph	Display positive and negative response	Pass
2	Survey Filled	Display Graphs	Pass

6.1.4.2 Display Subject Registration graphs

Refer to Table 6.17 for viewing Testing of Display Registration graph Component.

TABLE 6.17: Subject Registration Graph Testing

Sr No	Test Data	Expected Result	Test Result
1	Hover over graph	Display positive and negative response	Pass
2	Registration Form Filled	Display Graphs	Pass

6.1.5 Admin page Testing

This page is used to add or remove Admins or their details.

- Add Admin List through .csv
- Add Admins manually
- Remove Admins from list

6.1.5.1 Add Admin List through .csv

Refer Table 6.18 to view Testing of Display Admin List Component.

TABLE 6.18: Add Admin List through .csv Testing

Sr No	Test Data	Expected Result	Test Result
1	Upload .csv file	Display Admins List	Pass
2	No .csv uploaded	Admin list not displayed	Pass

6.1.5.2 Add Admins manually

For Testing of Add Admins manually view Table 6.19.

TABLE 6.19: Add Admins manually Testing

Sr No	Test Data	Expected Result	Test Result
1	Enter Admin email	Display details if Admin	Pass
2	Enter Admin email	Invalid email if not found	Pass

6.1.5.3 Remove Admins from list

To view testing of Remove Admin from list view Table 6.20.

TABLE 6.20: Remove Admin Testing

Sr No	Test Data	Expected Result	Test Result
1	Remove Admin button Clicked	Admin Removed from list	Pass
2	Remove Admin after Confirm clicked	Admin not removed	Fail

6.1.6 Student page Testing

This page is used to add or remove Students or their details.

- Add Student List through .csv
- Add Students manually
- Remove Students from list

6.1.6.1 Add Student List through .csv

Refer Table 6.21 to view Testing of Display Student List Component.

TABLE 6.21: Add Student List through .csv Testing

Sr No	Test Data	Expected Result	Test Result
1	Upload .csv file	Display Students List	Pass
2	No .csv uploaded	Student list not displayed	Pass

6.1.6.2 Add Students manually

For Testing of Add Students manually view Table 6.22.

TABLE 6.22: Add Admins manually Testing

Sr No	Test Data	Expected Result	Test Result
1	Enter Student email	Display details if Student	Pass
2	Enter Student email	Invalid email if not found	Pass

6.1.6.3 Remove Students from list

To view testing of Remove Student from list view Table 6.23.

TABLE 6.23: Remove Student Testing

Sr No	Test Data	Expected Result	Test Result
1	Remove Student button Clicked	Student Removed from list	Pass
2	Remove Student after Confirm clicked	Student not removed	Fail

6.1.7 Teacher page Testing

This page is used to add or remove Teachers or their details.

- Add Teacher List through .csv
- Add Teachers manually
- Remove Teachers from list

6.1.7.1 Add Teacher List through .csv

Refer Table 6.24 to view Testing of Display Teacher List Component.

TABLE 6.24: Add Teacher List through .csv Testing

Sr No	Test Data	Expected Result	Test Result
1	Upload .csv file	Display Teachers List	Pass
2	No .csv uploaded	Teacher list not displayed	Pass

6.1.7.2 Add Teachers manually

For Testing of Add Teachers manually view Table 6.25.

TABLE 6.25: Add Teachers manually Testing

Sr No	Test Data	Expected Result	Test Result
1	Enter Teacher email	Display details if Teacher	Pass
2	Enter Teacher email	Invalid email if not found	Pass

6.1.7.3 Remove Teachers from list

To view testing of Remove Teacher from list view Table 6.26.

TABLE 6.26: Remove Student Testing

Sr No	Test Data	Expected Result	Test Result
1	Remove Teacher button Clicked	Teacher Removed from list	Pass
2	Remove Teacher after Confirm clicked	Teacher not removed	Fail

6.1.8 Reminder Testing

This page is used to send reminders to selected Students,Admins or Teachers.

- Selected Category.
- Send reminder button.

Refer to Table 6.27 to view testing of Selected Category Component.

TABLE 6.27: Selected Category Testing

Sr No	Test Data	Expected Result	Test Result
1	Students	Display Student details	Pass
2	Teachers	Display Teacher details	Pass
3	Admins	Display Admin details	Pass

Refer Table 6.28 to view testing of Send reminder button.

TABLE 6.28: Reminder button Testing

Sr No	Test Data	Expected Result	Test Result
1	Category selected	Reminder sent	Pass
2	Category not selected	Reminder not sent	Pass

6.1.9 Results Testing

This page is used to View and download results of surveys and subject registration forms.

- View Results.
- Download Results button.

Admin can view results of filled forms ,its testing is shown in Table 6.29.

TABLE 6.29: View result Testing

Sr No	Test Data	Expected Result	Test Result
1	Extract data from database	Display results	Pass
2	Data not extracted from database	Results not Displayed	Pass

Admin can download results for each category ,its testing is shown in Table 6.30.

TABLE 6.30: Download result button Testing

Sr No	Test Data	Expected Result	Test Result
1	Button clicked	Results downloaded if data is available	Pass
2	Button clicked	Results not downloaded if data is not available	Pass
3	Button not clicked	Results not downloaded data is viewed only	Pass

6.1.10 Profile Testing

This page is used to View and edit user's profile picture email and password.

- View /edit Profile picture.
- View /edit Email address field.
- View /edit Password field.
- Save Changes button.

Admin can View and Edit Profile picture,Testing is shown in Table 6.31.

TABLE 6.31: View/edit Profile picture Testing

Sr No	Test Data	Expected Result	Test Result
1	Upload .png or .jpg	Picture uploaded	Pass
2	Picture format not correct	picture not uploaded	Pass

Admin can View and Edit Email address field ,its testing is displayed in Table 6.32.

TABLE 6.32: View/edit Email Testing

Sr No	Test Data	Expected Result	Test Result
1	example@gmail.com	Email Pass	Pass
2	example@mail.com	Email Pass	Pass
3	example@0123gmail.com	Email Fail	Pass
4	example@mail.pk	Email Pass	Pass
5	example@mail.	Email Fail	Pass
6	@mail.com	Email Fail	Pass
7	email does not exist	Email Fail	Fail

Refer to Table 6.33 to view testing of View and Edit Password field.

TABLE 6.33: View/edit Password Testing

Sr No	Test Data	Expected Result	Test Result
1	ABCabc123@	Password must be more than 8 characters	Pass
2	ABCD	Password must have small letters	Fail
3	abcd	Password must have numbers	Fail
4	abc123	Password must have capital letters	Fail
5	ABC123.	Password must have special characters	Fail
6	12@/?	Password must have small and capital letters.	Fail
7	a@A1	Password must be more than 8 characters	Fail

Save Changes Testing is displayed by Table 6.34.

TABLE 6.34: Save changes button Testing

Sr No	Test Data	Expected Result	Test Result
1	Button clicked	Profile edited if valid fields	Pass
2	Button clicked	Profile not edited if invalid fields	Pass
3	Button not clicked	Profile not edited	Pass

6.1.11 Fill Form Testing

This page is used to Fill Form.

- Progress bar.
- Text field.
- Send Response button.

User can fill form by filling progress bar which is tested and testing is shown in Table 6.35.

TABLE 6.35: Progress bar Testing

Sr No	Test Data	Expected Result	Test Result
1	Progress bar filled 20%	Progress Submitted	Pass
2	Progress bar filled 40%	Progress Submitted	Pass
3	Progress bar filled 60%	Progress Submitted	Pass
4	Progress bar filled 80%	Progress Submitted	Pass
5	Progress bar filled 100%	Progress Submitted	Pass
6	Progress bar filled 0%	Progress not Submitted	Fail

User can fill Text box whose testing is displayed in Table 6.36.

TABLE 6.36: Text Field Testing

Sr No	Test Data	Expected Result	Test Result
1	No comments.	Text Pass	Pass
2		Empty text field not Pass.	Fail

User will Submit response by clicking Send Response Button whose testing is displayed by Table 6.37.

TABLE 6.37: Send Response Button Testing

Sr No	Test Data	Expected Result	Test Result
1	Progress bar or text field not filled.	Response not submitted	Fail
2	Progress bar and text field filled.	Response submitted	Pass

6.2 Integration Testing

Once each unit undergoes comprehensive testing, it is combined with other units to form modules or components that are tasked with specific activities.

Integration testing is then performed to ensure that entire sections of an application operate as intended, ensuring smooth interactions between units.

These tests are often based on user scenarios, like logging into an application or opening files. Integration tests can be conducted by developers or independent testers and typically include a mix of automated functional and manual tests.

6.2.1 Sign-in

User must be able to Sign in and View profile.Sign in Module is integrated with Authentication module which allows only authenticated user to log-in ,its testing is displayed in Table 6.38.

TABLE 6.38: Sign in Integration Testing

Test case	Description	Expected Result	Actual Result	Pass/Fail
1	User signs in in to system	User must be able to sign in to system	Able to sign in	Pass
2	User does not exist	User must not be able to sign in to system	Cannot sign in	Pass

6.2.2 View Stats

User must be able to View Stats in a graphical format for all the forms that are currently up, on the stats page .Admin can View Stats after Forms are filled by user ,Testing is shown in Table 6.39.

TABLE 6.39: View Stats Integration Testing

Test case	Description	Expected Result	Actual Result	Pass/Fail
1	User opens stats page	User must be able to see graphical results of forms	Able to see graphs	Pass
2	Forms does not exist	User must be able to see a beatloader	beatloader is visible	Pass
3	Forms are loading	User must be able to see a beatloader	beatloader is visible	Pass

6.2.3 Create Student Forms

Admin must be able to create form for students.Admin can create Student Forms ,Student Form Creation Testing is displayed by Table 6.40.

TABLE 6.40: Create Student Forms Integration Testing

Test case	Description	Expected Result	Actual Result	Pass/Fail
1	Category Student selected	User able to create form	Student form created	Pass
2	Category not selected	User not able to create form	Student form not created	Pass

6.2.4 Create Final year Student Forms

Admin must be able to create form for Final year students. Admin can create Final year Student Forms , Final year Student Form Creation Testing is displayed by Table 6.41

TABLE 6.41: Create Final year Student Forms Integration Testing

Test case	Description	Expected Result	Actual Result	Pass/Fail
1	Category Final year Student selected	User able to create form	Final year Student form created	Pass
2	Category not selected	User not able to create form	Final year Student form not created	Pass

6.2.5 Create Alumina Forms

Admin must be able to create form for Alumina. Admin can create Alumina Forms , Alumina Form Creation Testing is displayed by Table 6.42.

TABLE 6.42: Create Alumina Forms Integration Testing

Test case	Description	Expected Result	Actual Result	Pass/Fail
1	Category Alumina selected	User able to create form	Alumina form created	Pass
2	Category not selected	User not able to create form	Alumina form not created	Pass

6.2.6 Create Organization Forms

Admin must be able to create form for students. Admin can create Organization Forms , Organization Form Creation Testing is displayed by Table 6.43.

TABLE 6.43: Create Organization Forms Integration Testing

Test case	Description	Expected Result	Actual Result	Pass/Fail
1	Category Organization selected	User able to create form	Organization form created	Pass
2	Organization not selected	User not able to create form	Organization form not created	Pass

6.2.7 Create Registration Form

User must be able to create Registration forms. Admin can create Registration Forms , Registration Form Creation Testing is displayed by Table 6.44.

TABLE 6.44: Create Registration Forms Integration Testing

Test case	Description	Expected Result	Actual Result	Pass/Fail
1	Enter number of seats, Subjects.	User able to create form	Registration form created	Pass
2	Number of seats, Subjects not entered.	User not able to create form	Registration form not created	Pass

6.2.8 View and Download Results

User must be able to View and Download Results. Admin must be able to Add Student with specific category. Admin can View and Download Results, Module Testing is displayed in Table 6.45.

TABLE 6.45: Results Integration Testing

Test case	Description	Expected Result	Actual Result	Pass/Fail
1	Category is selected	User must be able to view/download results	As expected	Pass
2	Category is not selected	User must not be able to view/download results	As expected	Pass

6.2.9 Add user

Admin can add Students by selecting specific category for each ,for viewing Testing of module refer to Table 6.46.

TABLE 6.46: Add User Integration Testing

Test case	Description	Expected Result	Actual Result	Pass/Fail
1	csv uploaded or data entered manually	User must be added or edited	As expected	Pass
2	csv not uploaded/data not entered manually	User must not be added/edited	As expected	Pass
3	error in data entered manually/error in csv	User must not be added/edited	As expected	Pass

6.2.10 Add Teacher

Admin must be able to Add Teachers. Admin can add Teachers by entering their details ,for viewing Testing of module refer to Table 6.47.

TABLE 6.47: Add Teacher Integration Testing

Test case	Description	Expected Result	Actual Result	Pass/Fail
1	csv uploaded or data entered manually	Teacher must be added or edited	As expected	Pass
2	csv not uploaded/data not entered manually	Teacher must not be added/edited	As expected	Pass
3	error in data entered manually/error in csv	Teacher must not be added/edited	As expected	Pass

6.2.11 Add Admin

Super Admin must be able to add Admins. Admin can add other Admins by entering their details, for viewing Testing of module refer to Table 6.48.

TABLE 6.48: Add Admin Integration Testing

Test case	Description	Expected Result	Actual Result	Pass/Fail
1	csv uploaded or data entered manually	Admin must be added or edited	As expected	Pass
2	csv not uploaded/data not entered manually	Admin must not be added/edited	As expected	Pass
3	error in data entered manually/error in csv	Admin must not be added/edited	As expected	Pass

6.2.12 Fill Form

User must be able to fill forms. User will Fill form sent by Admin and submit it after filling it, the testing of this module is displayed by Table 6.49.

TABLE 6.49: Fill form Integration Testing

Test case	Description	Expected Result	Actual Result	Pass/Fail
1	form is sent to user through link.	User must be able to fill form	As expected	Pass
2	form is not sent to user through mail.	User will not be able to fill form	As expected	Pass

6.2.13 Send Reminders

Admin must be able to Send reminders. Admin can Send Reminders to selected category of Users , as shown in Table 6.50.

TABLE 6.50: Send reminders Integration Testing

Test case	Description	Expected Result	Actual Result	Pass/Fail
1	category and form name is selected.	reminder will be sent for that form.	As expected	Pass
2	category and form name is not selected.	reminder will not be sent for that form.	As expected	Pass

6.2.14 Edit Profile

Admin must be able to edit profile, by editing his Username and Password ,Testing is shown in Table 6.51.

TABLE 6.51: Edit Profile Integration Testing

Test case	Description	Expected Result	Actual Result	Pass/Fail
1	User Signed in	User must be able to edit profile	As expected	Pass
2	User is not Signed in.	User must not be able to edit profile	As expected	Pass

6.2.15 Change Mode

Both Admin and User can change mode to Dark mode ,to view testing of this module refer to Table 6.52.

TABLE 6.52: Change mode Integration Testing

Test case	Description	Expected Result	Actual Result	Pass/Fail
1	Toggle button clicked.	User must be able to change mode.	As expected	Pass
2	Toggle button not clicked.	User must not be able to change mode.	As expected	Pass

6.3 System Testing

System testing is a crucial black box testing method employed to assess the completed and integrated system as a whole, ensuring it fulfills specified requirements. This phase involves evaluating the functionality of the software from end-to-end, covering all aspects of its operation. By examining the system as a unified entity, system testing ensures that all components interact seamlessly and that the software performs as expected in real-world scenarios.

6.3.1 Sign-in Testing

Refer to Table 6.53 to view Sign in Testing of System that allows authorized Admin to login.

TABLE 6.53: Sign in System Testing

Test case ID	ID_001	Test case Description	Sign in to Account	
Created by	Moiz	Reviewed by	Fatima	Version
QA Testers log	Review comments from Munazza incorporated in version 1.0.1			
Tester's name	Fatima	Date Tested	3/3/2024	Pass/Fail/Not executed
S#	Prerequisites	S#	Test data	
1	Have internet connection	1	Email=fz1@gmail.com	
2	User must exist	2	Password=Aa123@	
Test scenario	Verified user able to sign in into the system			
Step#	Step details	Expected results	Actual results	Pass/Fail
1	Enter Email and password	Credentials entered	As expected	Pass
2	Click sign in button	Signed in into system	As expected	Pass
Post Condition	User Signed in into system and able to use its functionalities.			

6.3.2 Create and Add User Testing

Admin can add user through csv or create user manually by selecting its specifications, Refer to Table 6.54 for details.

TABLE 6.54: Create user System Testing

Test case ID	ID_002	Test case Description		Create new user	
Created by	Moiz	Reviewed by	Munazza	Version	1.0.1
QA Testers log	Review comments from Fatima incorporated in version 1.0.1				
Tester's name	Fatima	Date Tested	1/3/2024	Pass/Fail/Not executed	Pass
S#	Prerequisites		S#	Test data	
1	Must have an internet connection		1	Upload csv=Uploaded	
2	User is signed in		2	Name=Fatima	
3	User has uploaded csv or entered data manually		3	Email=fz1@gmail.com	
4			4	Semester=8	
5			5	Section=B	
Test scenario	Verified user able to Add/edit or remove Student				
Step#	Step details	Expected results		Actual results	Pass/Fail
1	Fill all fields to add/edit Users	Data entered in to system		As expected	Pass
2	Upload csv file	Data entered in to system		As expected	Pass
3	Click Add button	Users will be added		As expected	Pass
Post Condition	User must be able to Add/edit or remove Users.				

6.3.3 Create and Add Teacher Testing

Admin can add Teacher through csv or create Teacher manually by entering its details, Refer to Table 6.55 for details.

TABLE 6.55: Create Teacher System Testing

Test case ID	ID_003	Test case Description		Create new Teacher	
Created by	Fatima	Reviewed by	Moiz	Version	1.0.1
QA Testers log	Review comments from Munazza incorporated in version 1.0.1				
Tester's name	Munazza	Date Tested	1/3/2024	Pass/Fail/Not executed	Pass
S#	Prerequisites		S#	Test data	
1	Must have an internet connection		1	Upload csv=Uploaded	
2	User is signed in		2	Name=Fatima	
3	User has uploaded csv or entered data manually		3	Email=fz1@gmail.com	
4			4	Semester=8	
5			5	Section=B	
Test scenario	Verified user able to Add/edit or remove Teacher				
Step#	Step details	Expected results	Actual results	Pass/Fail	
1	Fill all fields to add/edit Teacher	Data entered in to system	As expected	Pass	
2	Upload csv file	Data entered in to system	As expected	Pass	
3	Click Add button	Teachers will be added	As expected	Pass	
Post Condition	User must be able to Add/edit or remove Teachers.				

6.3.4 Create and Add Admin Testing

Admin can add other Admins through csv or create Admins manually by entering its details, Refer to Table 6.56 for details.

TABLE 6.56: Create Admin System Testing

Test case ID	ID_004	Test case Description		Create new Admin	
Created by	Munazza	Reviewed by	Moiz	Version	1.0.1
QA Testers log	Review comments from Fatima incorporated in version 1.0.1				
Tester's name	Fatima	Date Tested	1/3/2024	Pass/Fail/Not executed	Pass
S#	Prerequisites				
1	Must have an internet connection				
2	User is signed in				
3	User has uploaded csv or entered data manually				
S#	Test data				
1	Upload csv=Uploaded				
2	Name=Fatima				
3	Email=fz1@gmail.com				
Test scenario	Verified user able to Add/edit or remove Admin				
Step#	Step details	Expected results	Actual results	Pass/Fail	
1	Fill all fields to add/edit Admin	Data entered in to system	As expected	Pass	
2	Upload csv file	Data entered in to system	As expected	Pass	
3	Click Add button	Admins will be added	As expected	Pass	
Post Condition	User must be able to Add/edit or remove Admins.				

6.3.5 Create Form Testing

Admin can create Subject Registration Form or Teacher Evaluation Survey, Refer to Table 6.57 for testing results.

TABLE 6.57: Create Forms System Testing

Test case ID	ID_005	Test case Description		Create new Form																															
Created by	Moiz	Reviewed by	Fatima	Version	1.0.1																														
QA Testers log	Review comments from Munazza incorporated in version 1.0.1																																		
Tester's name	Munazza	Date Tested	6/3/2024	Pass/Fail/Not executed	Pass																														
S#	<table border="1"> <thead> <tr> <th>S#</th><th>Prerequisites</th><th>S#</th><th>Test data</th><th></th></tr> </thead> <tbody> <tr> <td>1</td><td>Must have an internet connection</td><td>1</td><td>Category=Final Year Student</td><td></td></tr> <tr> <td>2</td><td>User is signed in</td><td>2</td><td>Header=Final Year Student evaluation Form</td><td></td></tr> <tr> <td>3</td><td>User has clicked Create form button.</td><td>3</td><td>Question= Test Question ?</td><td></td></tr> <tr> <td></td><td></td><td>4</td><td>Progress bar=100</td><td></td></tr> <tr> <td></td><td></td><td>5</td><td>Additional Comment= Test Comment</td><td></td></tr> </tbody> </table>					S#	Prerequisites	S#	Test data		1	Must have an internet connection	1	Category=Final Year Student		2	User is signed in	2	Header=Final Year Student evaluation Form		3	User has clicked Create form button.	3	Question= Test Question ?				4	Progress bar=100				5	Additional Comment= Test Comment	
S#	Prerequisites	S#	Test data																																
1	Must have an internet connection	1	Category=Final Year Student																																
2	User is signed in	2	Header=Final Year Student evaluation Form																																
3	User has clicked Create form button.	3	Question= Test Question ?																																
		4	Progress bar=100																																
		5	Additional Comment= Test Comment																																
Test scenario	Verified user able to create Form.																																		
Step#	Step details	Expected results		Actual results	Pass/Fail																														
1	Fill all fields to create form	Data entered in to system		Form created	Pass																														
2	All fields are not filled	Data not entered in to system		Form not created	Pass																														
Post Condition	User must be able to create form.																																		

6.3.6 View and Download Results Testing

Admin can View and download results of filled forms ,refer to Table 6.58 to view results of testing.

TABLE 6.58: Download result System Testing

Test case ID	ID_006	Test case Description		Download results in csv format	
Created by	Moiz	Reviewed by	Munazza	Version	1.0.1
QA Testers log	Review comments from Fatima incorporated in version 1.0.1				
Tester's name	Fatima	Date Tested	29/2/2024	Pass/Fail/Not executed	Pass
S#	Prerequisites				
1	Must have an internet connection				
2	User is signed in				
3	User has entered category for results to download				
S#	Test data				
1	Category=Students				
2	Form Name=Course and Teacher evaluation Form				
Test scenario	Verified user able to View and download Results in csv format				
Step#	Step details	Expected results		Actual results	Pass/Fail
1	Enter Category	Category will be selected		As expected	Pass
2	Enter Form name	Form will be selected		As expected	Pass
3	Click download button	csv of Results will be downloaded		As expected	Pass
Post Condition	User must be able to Add/edit or remove Admins.				

6.3.7 Select Category and Send Reminders Testing

Admin can select specific category and send them forms ,refer to Table 6.59 to view testing results.

TABLE 6.59: Send Reminders System Testing

Test case ID	ID_007		Test case Description	Send reminders to selected category	
Created by	Fatima	Reviewed by	Moiz	Version	1.0.1
QA Testers log	Review comments from Munazza incorporated in version 1.0.1				
Tester's name	Munazza	Date Tested	29/2/2024	Pass/Fail/Not executed	Pass
S#	Prerequisites				
1	Must have an internet connection				
2	User is signed in				
3	User has entered category to send reminders or enter user manually				
S#	Test data				
1	Category=Teachers				
2	Import data from database				
Test scenario	Verified user able to send reminder to selected category				
Step#	Step details	Expected results		Actual results	Pass/Fail
1	Enter Category	Category will be selected		As expected	Pass
2	Import data from database	Category will be selected		As expected	Pass
3	Click Send button	Reminders will be sent.		As expected	Pass
Post Condition	User must be able to send reminders to selected category.				

6.3.8 View and Edit Profile Testing

Admin can view and edit Profile ,refer to Table 6.60 for testing results.

TABLE 6.60: Edit Profile System Testing

Test case ID	ID_008	Test case Description	Edit Profile		
Created by	Munazza	Reviewed by	Moiz	Version	1.0.1
QA Testers log	Review comments from Fatima incorporated in version 1.0.1				
Tester's name	Fatima	Date Tested	5/3/2024	Pass/Fail/Not executed	Pass
S#	Prerequisites				
1	Must have an internet connection				
2	User exists				
3	User is signed in				
4	User has entered Picture , email , password or name to edit				
Test scenario	Verified user able to View and edit profile				
Step#	Step details	Expected results	Actual results	Pass/Fail	
1	Enter Field to edit	Data entered into system	As expected	Pass	
2	Click save changes button	Profile will be edited	As expected	Pass	
Post Condition	User must be able to send reminders to selected category.				

6.3.9 Fill Form Testing

Users can Fill forms sent by Admin ,refer to Table 6.61 for testing results.

TABLE 6.61: Fill Form System Testing

Test case ID	ID_009	Test case Description	Fill Form		
Created by	Moiz	Reviewed by	Fatima	Version	1.0.1
QA Testers log	Review comments from Munazza incorporated in version 1.0.1				
Tester's name	Fatima	Date Tested	6/3/2024	Pass/Fail/Not executed	Pass
S#	Prerequisites				
1	Have internet connection				
2	User must get link to form through mail.				
S#	Test data				
1	Progress bar=FILLED				
2	Test field=FILLED				
3	Submit Response button =Clicked				
Test scenario	Verified user able to fill form.				
Step#	Step details	Expected results	Actual results	Pass/Fail	
1	Fill Progress bar and Text fields.	Submit response.	As expected	Pass	
2	Text field not filled.	Response not submitted.	Response not submitted	Pass	
Post Condition	User Signed in into system and able to use its functionalities.				

6.4 Acceptance Testing

Acceptance testing marks the final phase of functional testing and serves to determine if the software is ready for delivery. Its primary aim is to verify that the final product aligns with the original business criteria and satisfies the end user's requirements. This entails conducting tests both internally and externally, necessitating the involvement of end users for beta testing. Beta testing is crucial for gathering authentic feedback from potential customers and addressing any remaining usability concerns before the product launch.

6.4.1 Sign-in Testing

Sign-in Testing ,System accepts only Authorized user and Super Admin can add other admins who will then receive a mail with username and password credentials,which can be used to sign in and can be edited by admins afterwards,referred by Table 6.62.

TABLE 6.62: Sign in Acceptance Testing

Test case ID	ID_001	Test case Description	Sign in to Account	
Created by	Moiz	Reviewed by	Fatima	Version
QA Testers log	Review comments from Munazza incorporated in version 1.0.1			
Tester's name	Fatima	Date Tested	3/3/2024	Pass/Fail/Not executed
S#	Prerequisites			
1	Have internet connection			
2	User must exist			
Test scenario		Verified user able to sign in into the system		
Step#	Step details	Expected results	Actual results	Pass/Fail
1	Enter Email and password	Credentials entered	As expected	Pass
2	Click sign in button	Signed in into system	As expected	Pass
Post Condition		User Signed in into system and able to use its functionalities.		

6.4.2 Create User Testing

Testing of Create User module, which allows Admin to Add Users csv in system ,Users can be Organization,Students,Alumina or Final Year Students.They will receive mail telling that they have been registered and can receive surveys to fill for teachers,referred by Table 6.63.

TABLE 6.63: Create user Acceptance Testing

Test case ID	ID_002	Test case Description	Create new user		
Created by	Moiz	Reviewed by	Munazza	Version	1.0.1
QA Testers log	Review comments from Fatima incorporated in version 1.0.1				
Tester's name	Fatima	Date Tested	1/3/2024	Pass/Fail/Not executed	Pass
S#	Prerequisites		S#	Test data	
1	Must have an internet connection		1	Upload csv=Uploaded	
2	User is signed in		2	Name=Fatima	
3	User has uploaded csv or entered data manually		3	Email=fz1@gmail.com	
Test scenario		Verified user able to Add/edit or remove Student			
Step#	Step details	Expected results		Actual results	Pass/Fail
1	Fill all fields to add/edit Users	Data entered in to system		As expected	Pass
2	Upload csv file	Data entered in to system		As expected	Pass
3	Click Add button	Users will be added		As expected	Pass
Post Condition		User must be able to Add/edit or remove Users.			

6.4.3 Create Teacher Testing

Testing of Create Teachers module that allows Admin to add Teachers of which evaluation is to be conducted, Admin will have to upload csv or add Teachers manually by entering details of Teachers, which are used to evaluate their performance, referred by Table 6.64.

TABLE 6.64: Create Teacher Acceptance Testing

Test case ID	ID_003	Test case Description	Create new Teacher		
Created by	Fatima	Reviewed by	Moiz	Version	1.0.1
QA Testers log	Review comments from Munazza incorporated in version 1.0.1				
Tester's name	Munazza	Date Tested	1/3/2024	Pass/Fail/Not executed	Pass
S#	Prerequisites		S#	Test data	
1	Must have an internet connection		1	Upload csv=Uploaded	
2	User is signed in		2	Name=Fatima	
3	User has uploaded csv or entered data manually		3	Email=fz1@gmail.com	
4			4	Semester=8	
5			5	Section=B	
Test scenario	Verified user able to Add/edit or remove Teacher				
Step#	Step details	Expected results	Actual results	Pass/Fail	
1	Fill all fields to add/edit Teacher	Data entered in to system	As expected	Pass	
2	Upload csv file	Data entered in to system	As expected	Pass	
3	Click Add button	Teachers will be added	As expected	Pass	
Post Condition	User must be able to Add/edit or remove Teachers.				

6.4.4 Create Admin Testing

Super Admin can add other Admins through uploading their csv or adding them manually ,which will have all the accesses to system that Super Admin has,Testing is shown in Table 6.65.

TABLE 6.65: Create Admin Acceptance Testing

Test case ID	ID_004	Test case Description	Create new Admin		
Created by	Munazza	Reviewed by	Fatima	Version	1.0.1
QA Testers log	Review comments from Moiz incorporated in version 1.0.1				
Tester's name	Moiz	Date Tested	1/3/2024	Pass/Fail/Not executed	Pass
S#	Prerequisites				
1	Must have an internet connection				
2	User is signed in				
3	User has uploaded csv or entered data manually				
S#	Test data				
1	Upload csv=Uploaded				
2	Name=Fatima				
3	Email=fz1@gmail.com				
Test scenario	Verified user able to Add/edit or remove Admin				
Step#	Step details	Expected results	Actual results	Pass/Fail	
1	Fill all fields to add/edit Admin	Data entered in to system	As expected	Pass	
2	Upload csv file	Data entered in to system	As expected	Pass	
3	Click Add button	Admins will be added	As expected	Pass	
Post Condition	User must be able to Add/edit or remove Admins.				

6.4.5 Create Student Form Testing

Admin can create Subject Registration Form or Survey for Users ,Table 6.66 displays testing of this module.

TABLE 6.66: Create Forms Acceptance Testing

Test case ID	ID_005	Test case Description		Create new Form									
Created by	Fatima	Reviewed by	Munazza	Version	1.0.1								
QA Testers log	Review comments from Moiz incorporated in version 1.0.1												
Tester's name	Moiz	Date Tested	1/3/2024	Pass/Fail/Not executed	Pass								
<table border="1"> <thead> <tr> <th>S#</th> <th>Prerequisites</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Must have an internet connection</td> </tr> <tr> <td>2</td> <td>User is signed in</td> </tr> <tr> <td>3</td> <td>User has clicked Create form button.</td> </tr> </tbody> </table>		S#	Prerequisites	1	Must have an internet connection	2	User is signed in	3	User has clicked Create form button.	S#	Test data		
S#	Prerequisites												
1	Must have an internet connection												
2	User is signed in												
3	User has clicked Create form button.												
		1	Category=Student										
		2	Header=Teachers and course evaluation Form										
		3	Question= Do you think syllabus is covered fully as per course outline?										
		4	Progress bar=100										
		5	Additional Comment= Test Comment										
Test scenario	Verified user able to create Student Form.												
Step#	Step details	Expected results		Actual results	Pass/Fail								
1	Fill all fields to create Student form	Data entered in to system		Student Form created	Pass								
2	All fields are not filled	Data not entered in to system		Student Form not created	Pass								
Post Condition	User must be able to create Student form.												

6.4.6 Create Final Year Student Form Testing

Admin can create Survey for Final Year students ,Table 6.67 displays testing of this module.

TABLE 6.67: Create Final Year Student Forms Acceptance Testing

Test case ID	ID_006	Test case Description	Create new Form														
Created by	Moiz	Reviewed by	Fatima	Version	1.0.1												
QA Testers log	Review comments from Munazza incorporated in version 1.0.1																
Tester's name	Munazza	Date Tested	6/3/2024	Pass/Fail/Not executed	Pass												
		<table border="1"> <thead> <tr> <th>S#</th> <th>Test data</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Category=Final Year Student</td> </tr> <tr> <td>2</td> <td>Header=Final Year Student evaluation Form</td> </tr> <tr> <td>3</td> <td>Question= Test Question ?</td> </tr> <tr> <td>4</td> <td>Progress bar=100</td> </tr> <tr> <td>5</td> <td>Additional Comment= Test Comment</td> </tr> </tbody> </table>				S#	Test data	1	Category=Final Year Student	2	Header=Final Year Student evaluation Form	3	Question= Test Question ?	4	Progress bar=100	5	Additional Comment= Test Comment
S#	Test data																
1	Category=Final Year Student																
2	Header=Final Year Student evaluation Form																
3	Question= Test Question ?																
4	Progress bar=100																
5	Additional Comment= Test Comment																
Test scenario	Verified user able to create Final Year Student Form.																
Step#	Step details	Expected results	Actual results	Pass/Fail													
1	Fill all fields to create form	Data entered in to system	Final Year Student Form created	Pass													
2	All fields are not filled	Data not entered in to system	Final Year Student Form not created	Pass													
Post Condition	User must be able to create Final Year Student form.																

6.4.7 Create Alumina Form Testing

Admin can create Survey for Alumina ,Table 6.68 displays testing of this module.

TABLE 6.68: Create Alumina Forms Acceptance Testing

Test case ID	ID_007	Test case Description	Create new Form		
Created by	Munazza	Reviewed by	Moiz	Version	1.0.1
QA Testers log	Review comments from Fatima incorporated in version 1.0.1				
Tester's name	Fatima	Date Tested	1/3/2024	Pass/Fail/Not executed	Pass
S#	Test data				
1	Category=Alumina				
2	Header=Alumina evaluation Form				
3	Question= Test question?				
4	Progress bar=100				
5	Additional Comment= Test Comment				
Test scenario	Verified user able to Alumina create Form.				
Step#	Step details	Expected results	Actual results	Pass/Fail	
1	Fill all fields to create form	Data entered in to system	Alumina Form created	Pass	
2	All fields are not filled	Data not entered in to system	Alumina Form not created	Pass	
Post Condition	User must be able to create Alumina form.				

6.4.8 Create Organization Form Testing

Admin can create Survey for Organization ,Table 6.69 displays testing of this module.

TABLE 6.69: Create Organization Forms Acceptance Testing

Test case ID	ID_008	Test case Description		Create new Form																					
Created by	Munazza	Reviewed by	Moiz	Version	1.0.1																				
QA Testers log		Review comments from Fatima incorporated in version 1.0.1																							
Tester's name	Fatima	Date Tested	1/3/2024	Pass/Fail/Not executed	Pass																				
<table border="1"> <thead> <tr> <th>S#</th> <th>Prerequisites</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Must have an internet connection</td> </tr> <tr> <td>2</td> <td>User is signed in</td> </tr> <tr> <td>3</td> <td>User has clicked Create form button.</td> </tr> </tbody> </table>		S#	Prerequisites	1	Must have an internet connection	2	User is signed in	3	User has clicked Create form button.	<table border="1"> <thead> <tr> <th>S#</th> <th>Test data</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Category=Organization</td> </tr> <tr> <td>2</td> <td>Header=Organization Form</td> </tr> <tr> <td>3</td> <td>Question=Test Question for Organization?</td> </tr> <tr> <td>4</td> <td>Progress bar=100</td> </tr> <tr> <td>5</td> <td>Additional Comment= Test Comment</td> </tr> </tbody> </table>				S#	Test data	1	Category=Organization	2	Header=Organization Form	3	Question=Test Question for Organization?	4	Progress bar=100	5	Additional Comment= Test Comment
S#	Prerequisites																								
1	Must have an internet connection																								
2	User is signed in																								
3	User has clicked Create form button.																								
S#	Test data																								
1	Category=Organization																								
2	Header=Organization Form																								
3	Question=Test Question for Organization?																								
4	Progress bar=100																								
5	Additional Comment= Test Comment																								
Test scenario		Verified user able to create Organization Form.																							
Step#	Step details	Expected results		Actual results	Pass/Fail																				
1	Fill all fields to create Organization form	Data entered in to system		Organization Form created	Pass																				
2	All fields are not filled	Data not entered in to system		Form not created	Pass																				
Post Condition		User must be able to create Organization form.																							

6.4.9 Download Results Testing

Table 6.70 displays testing of Download Results module which allows Admin to View and Download Results of forms filled.

TABLE 6.70: Download result Acceptance Testing

Test case ID	ID_009	Test case Description		Download results in csv format	
Created by	Moiz	Reviewed by	Munazza	Version	1.0.1
QA Testers log	Review comments from Fatima incorporated in version 1.0.1				
Tester's name	Fatima	Date Tested	29/2/2024	Pass/Fail/Not executed	Pass
S#	Prerequisites				
1	Must have an internet connection				
2	User is signed in				
3	User has entered category for results to download				
S#	Test data				
1	Category=Students				
2	Form Name=Course and Teacher evaluation Form				
Test scenario	Verified user able to View and download Results in csv format				
Step#	Step details	Expected results		Actual results	Pass/Fail
1	Enter Category	Category will be selected		As expected	Pass
2	Enter Form name	Form will be selected		As expected	Pass
3	Click download button	csv of Results will be downloaded		As expected	Pass
Post Condition	User must be able to Add/edit or remove Admins.				

6.4.10 Send Reminders Testing

Admin can send reminders of forms or surveys to selected category, testing is displayed in Table 6.71.

TABLE 6.71: Send Reminders Acceptance Testing

Test case ID	ID_010	Test case Description		Send reminders to selected category	
Created by	Moiz	Reviewed by	Fatima	Version	1.0.1
QA Testers log	Review comments from Munazza incorporated in version 1.0.1				
Tester's name	Munazza	Date Tested	29/2/2024	Pass/Fail/Not executed	Pass
S#	Prerequisites				
1	Must have an internet connection				
2	User is signed in				
3	User has entered category to send reminders or enter user manually				
Test scenario	Verified user able to send reminder to selected category				
Step#	Step details	Expected results		Actual results	Pass/Fail
1	Enter Category	Category will be selected		As expected	Pass
2	Import data from database	Category will be selected		As expected	Pass
3	Click Send button	Reminders will be sent.		As expected	Pass
Post Condition	User must be able to send reminders to selected category.				

6.4.11 Edit Profile Testing

Admin can edit his profile as well ,Refer to table 6.72 to view testing results of this module.

TABLE 6.72: Edit Profile Acceptance Testing

Test case ID	ID_011	Test case Description		Edit Profile	
Created by	Fatima	Reviewed by	Moiz	Version	1.0.1
QA Testers log	Review comments from Munazza incorporated in version 1.0.1				
Tester's name	Munazza	Date Tested	5/3/2024	Pass/Fail/Not executed	Pass
S#	Prerequisites				
1	Must have an internet connection				
2	User exists				
3	User is signed in				
4	User has entered Picture , email , password or name to edit				
S#	Test data				
1	Profile picture=uploaded				
2	Name=Munazza				
3	Email=mz231@gmail.com				
4	Password=Aa123@				
Test scenario	Verified user able to View and edit profile				
Step#	Step details	Expected results		Actual results	Pass/Fail
1	Enter Field to edit	Data entered into system		As expected	Pass
2	Click save changes button	Profile will be edited		As expected	Pass
Post Condition	User must be able to send reminders to selected category.				

6.4.12 Change Mode

Admin and user can change mode of system to dark mode ,results of tests are shown in Table 6.73.

TABLE 6.73: Change mode Acceptance Testing

Test case ID	ID_012	Test case Description		Change to dark mode	
Created by	Fatima	Reviewed by	Munazza	Version	1.0.1
QA Testers log		Review comments from Munazza incorporated in version 1.0.1			
Tester's name	Munazza	Date Tested	7/3/2024	Pass/Fail/Not executed	Pass
S#	Prerequisites				
1	Must have an internet connection				
2	User exists				
3	User is signed in				
Test scenario		Verified user able to change mode of app.			
Step#	Step details	Expected results		Actual results	Pass/Fail
1	Turn on Dark mode button.	Mode will be changed to Dark mode.		As expected	Pass
Post Condition		User must be able to change mode.			

6.5 Performance Testing

Performance testing is a non-functional testing methodology aimed at evaluating how an application responds under diverse conditions. Its objective is to assess the responsiveness and stability of the application in real-world scenarios. This type of testing encompasses four distinct categories:

- Load testing
- Stress testing
- Endurance testing
- Spike testing

6.5.1 Load testing

Load testing involves subjecting your software, application, or website to escalating simulated demands to determine its ability to handle the intended workload. This process verifies whether the system can effectively manage the anticipated levels of usage and traffic..For reference View Table [6.74](#) for reference.

TABLE 6.74: Load testing

Sr. No	Functionality	Estimated Load Time (seconds)
1	Sign in	2 to 3
2	Home	1 to 2
3	Load Profile	3 to 4
4	Load Results	3 to 4
5	Download Results	6 to 8
6	Add User	1 to 2
7	Add Teacher	1 to 2
8	Add Admin	1 to 2
9	Upload csv	3 to 4
10	Edit Profile	4 to 5
11	Send Reminders	2 to 3
12	Change Mode	1 to 2

6.5.2 Stress testing

Stress testing goes beyond load testing and assesses how your software performs at or above its peak load capacity. The aim is to intentionally push the application to its limits by applying both realistic and unrealistic load scenarios. Through stress testing, you can identify the point at which your software fails.For reference View Table [6.75](#) for reference.

TABLE 6.75: Stress testing

Sr. No	Functionality	Estimated Load Time (seconds)
1	Sign in	3 to 5
2	Home	2 to 3
3	Load Profile	4 to 5
4	Load Results	4 to 5
5	Download Results	9 to 10
6	Add User	2 to 3
7	Add Teacher	2 to 3
8	Add Admin	2 to 3
9	Upload csv	4 to 5
10	Edit Profile	5 to 6
11	Send Reminders	4 to 5
12	Change Mode	3 to 5

6.5.3 Endurance testing

We done this testing under the heavy load of the data and putting that load Over a Longer amount of time to understand how your system will behave under sustained use, making it a longer process than load or stress testing. For reference View Table 6.76 for reference.

TABLE 6.76: Endurance testing

Sr. No	Endurance Testing
1	Add so many Users so that system is not able to handle them.
2	Add so many Teachers so that system is not able to handle them.
3	Add so many Admins so that system is not able to handle them.
4	Download Result so many times that system is not able to handle downloading.
5	Send reminders so many times that system is not able to handle sending more reminders.

6.5.4 Spike testing

We done this testing under the heavy load of the data and putting that load of same amount of data over a longer amount of time to understand how your

system will behave when the load is suddenly and drastically increase. For reference View Table 6.77 for reference.

TABLE 6.77: Spike testing

Sr. No	Spike Testing
1	Enter many wrong email and passwords again and again.
2	Enter many invalid Users.
3	Enter many invalid Teachers.
4	Enter many invalid Admins.
5	Enter invalid file format.

6.6 Security Testing

The emergence of cloud-based testing platforms and the increasing prevalence of cyberattacks have sparked a rising concern regarding the security of data utilized and stored in software. Security testing, a non-functional software testing approach, is employed to ascertain the protection of information and data within a system. Its purpose is to systematically identify vulnerabilities and security risks that could lead to unauthorized access to or loss of information by probing the application for weaknesses. This testing method encompasses various types, each targeting the verification of five fundamental principles of security:

- Integrity
- Confidentiality
- Authentication
- Authorization
- Availability

6.6.1 Integrity

We have made a custom react hook to ensure the integrity of the system, and also we are using session management to ensure that the admin who logged in stays logged in as long as the browser is open, so the system can be used in multiple tabs. In addition to that the system is using a custom "check authentication" function on the server side to ensure that every client side page that is accessed must have the permission to be accessed by certain type of user. The pages that

are allowed to users are open and do not require log in, but the pages that are only allowed to admins are only allowed after admin is Passly logged in.

6.6.2 Confidentiality

To ensure the confidentiality, first there is a default admin called super admin whose credentials will be known only to the owner of the system, admins can be added only by previous admins, and even the previous admins cannot know the password of other admins. The password is generated by a special algorithm that generates a new password at random with predefined specifications, and even stored in the database in encrypted form. All the efforts are made to make sure that no one except for the admin can login to the system.

6.6.3 Authentication

When the system is accessed it is made sure to make the user stay on the login page if the session is yet to be created. When the correct credentials are added to the login form, they are sent to the server where the emails are matched from the database, and if a match is found then encrypted password is fetched and deciphered on the server to match the password entered. If there is a match only then a session is created that has admin information in it, but for safety purposes, the session created also have the admin information in encrypted form.

6.6.4 Authorization

Whenever access is requested to any page even when the same page is reloaded, it is made sure that the URL user is looking for has public access or admins only access, if it has public access it opens up, but if it has admins only access the encrypted information is requested from the session and matched the database to make sure that the admin is really an authorized admin and not a fake session created by a hacker or anything resonating with the above mentioned scenario.

6.6.5 Availability

System must be available 24/7 and provide services to the users and admins. User can send request to the server for their required form and admins can log in to the system at any time and if any kind of authentication error occurs on the admin side, the system ensures stability and does not crash, instead it redirects the admin to the admin page. Similarly, if the admin tries to go to the login page without logging out, he/she will not be allowed to do so and will be redirected to the admin dashboard page.

6.7 Usability Testing

Usability testing is a crucial method used to assess the ease-of-use of an application from the perspective of end-users. Typically conducted during the system or acceptance testing stages, its objective is to evaluate whether the visible design and aesthetics of an application align with the intended workflow for various processes, such as logging in. This testing focuses on gathering feedback from users to ensure that the application is user-friendly and intuitive in its operation. View Table 6.78 for reference.

TABLE 6.78: Usability Testing

Sr. No	System Flow
1	Super Admin sets up the system and grants access to authorized Admin
2	Only Admins added by the Super Admin are granted permission to sign in.
3	Admins can view and manage their profiles.
4	Admins can Add Teachers manually or by uploading csv file containing Teacher's data.
5	Admins can Add Users manually or by uploading csv file containing User's data.
6	Admins can create new Surveys and Forms for a specific category.
7	After form creation, Admins can send reminders to categorized user as needed.
8	Users receive emails containing encrypted links to access the form.
9	Users click on the link, fill out the forms, and submit their responses.
10	User's submitted data is securely stored and available for Admins to review at any time.
11	Admins can view result of a specific category and form.
12	For comprehensive analysis, Admins can easily download the entire dataset in CSV format.

6.8 Compatibility Testing

Compatibility testing is essential for assessing how well an application or software functions across various environments. This process involves checking whether your product is compatible with multiple operating systems, platforms, browsers, or resolution configurations. The objective is to guarantee that your software's functionality remains consistent across any environment your end users might utilize. To ensure compatibility, we conducted testing on our project by running it on different systems and configurations.

Chapter 7

Results and Outputs

Results and Outputs of system are shown in this section.

7.0.1 Sign-in

Allows authenticated user to sign-in. View Figure 7.1 for reference.

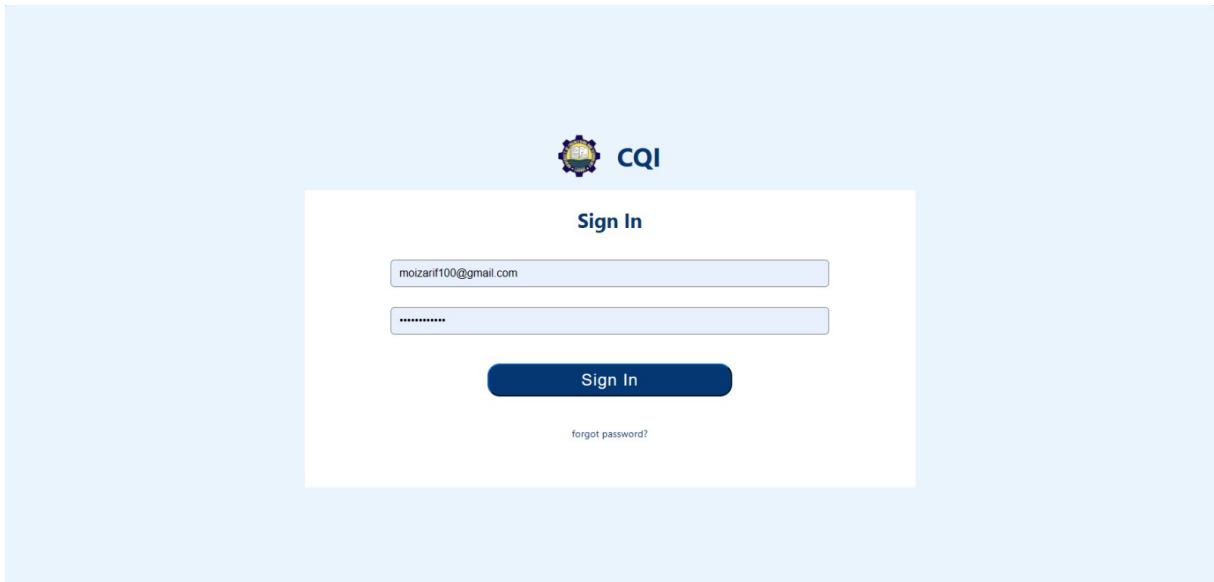


FIGURE 7.1: Sign-in

7.0.2 Admin Dashboard

Allows admin to view results of forms. View Figure 7.2 for reference.

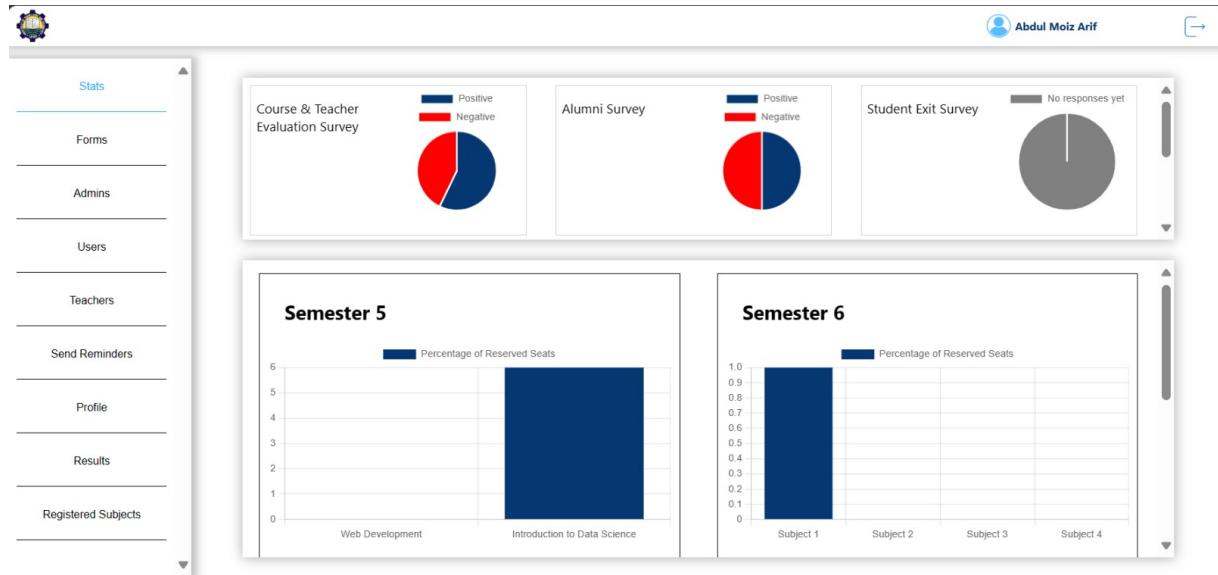


FIGURE 7.2: Admin Dashboard

7.0.3 Created Forms

Allows Admin to create surveys and forms. View Figure 7.3 for reference.

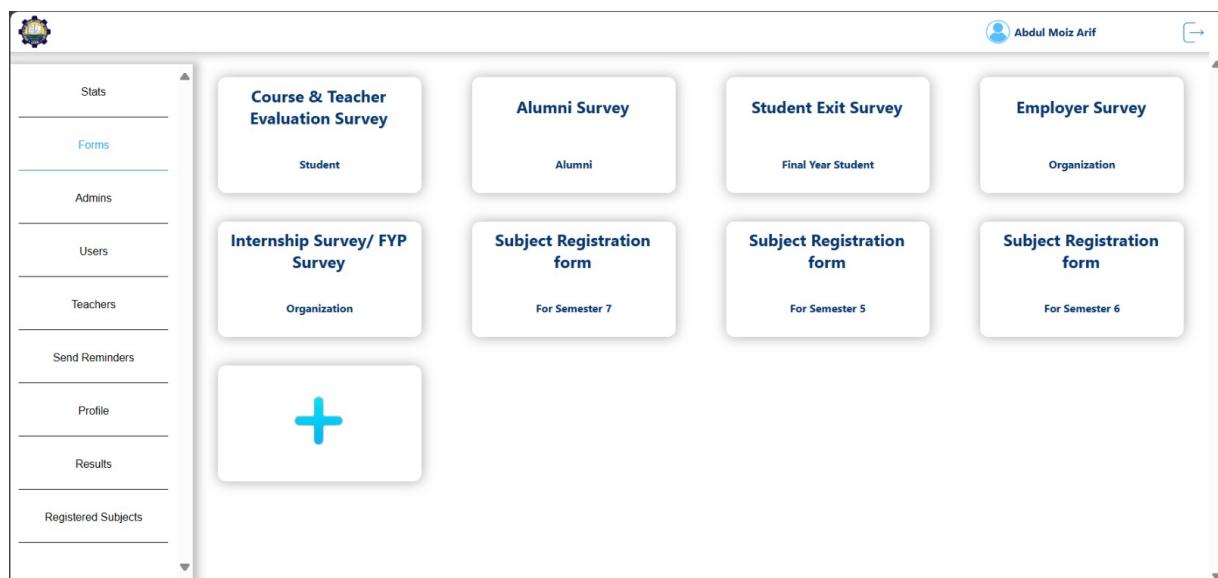


FIGURE 7.3: Created Forms

7.0.4 Create new Survey

Allows Admin to select category while creating survey. View Figure 7.4 for reference.

FIGURE 7.4: Create new Survey

7.0.5 Create Subject Registration Form

Allows Admin to choose number of students and subjects for subject registration. View Figure 7.5 for reference.

FIGURE 7.5: Create Subject Registration Form

7.0.6 Filling Survey Form

Allows User to fill survey. View Figure 7.6 for reference.

The screenshot shows a survey titled "Student Exit Survey". It contains two horizontal rating scales. The first scale is labeled "How independent you are in your thinking? Please rate?" with a dark blue segment on the left and a light purple segment on the right, ending at 20%. The second scale is labeled "Are you willing to continue to enhance your education? If yes, please rate" with a dark blue segment on the left and a light purple segment on the right, ending at 100%. Below these scales is a section titled "Lifelong Learning" with the question "Are you willing to participate in professional trainings? If yes, please rate". This section has a dark blue segment on the left and a light purple segment on the right, ending at 60%. There is also a "Additional Comments?" text input field containing "Example text" and a "SUBMIT" button at the bottom.

FIGURE 7.6: Filling Survey Form

7.0.7 Filling Subject Registration Form

Allows User to fill form and makes sure that user selects specific number of subjects. View Figure 7.7 for reference.

The screenshot shows a form titled "Select 2 Subjects out of 4". It lists four subject options with checkboxes: "Enterprise Application Development", "Game Development", "Big Data Analysis", and "Cloud Computing". Below the list is a message in red text: "Please select 2 more checkboxes." A scroll bar is visible on the right side of the form.

FIGURE 7.7: Filling Subject Registration Form

7.0.8 Filling Subject Registration Form

Allows User to fill form. View Figure 7.8 for reference.

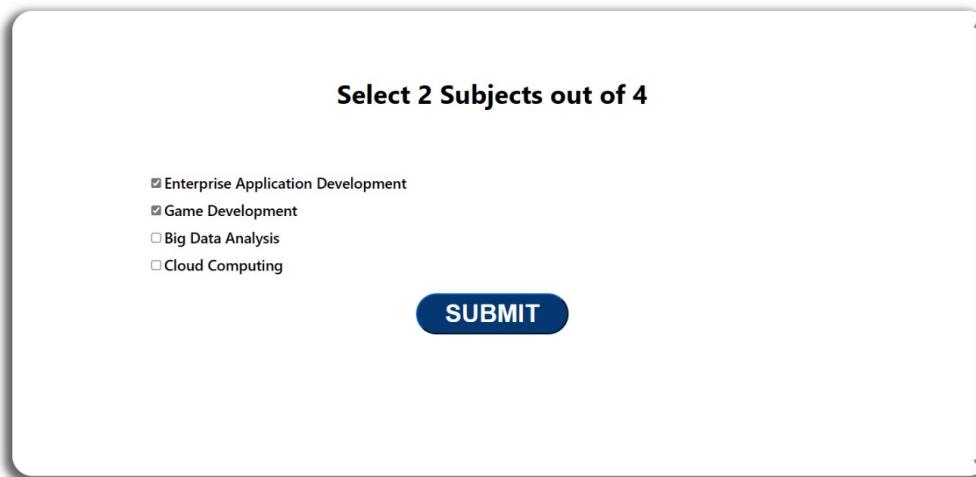


FIGURE 7.8: Filling Subject Registration Form

7.0.9 Add User through csv

Allows Admin to add user csv. View Figure 7.9 for reference.

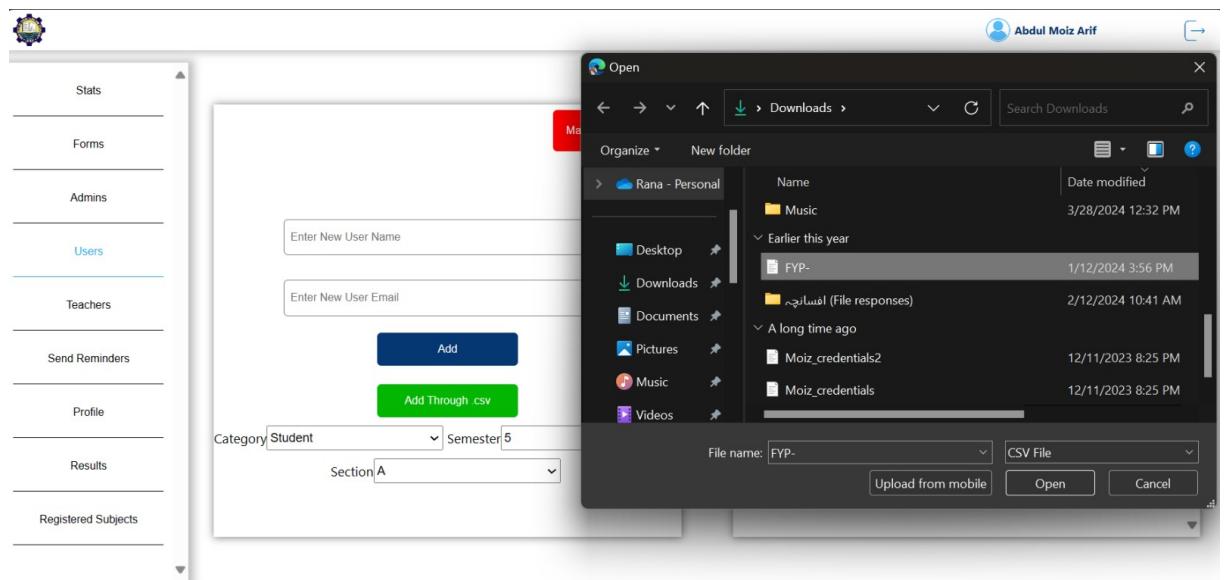


FIGURE 7.9: Add User through csv

7.0.10 Add User manually

Allows Admin to add user manually. View Figure 7.10 for reference.

The screenshot shows a user interface for adding a new user. On the left is a vertical sidebar with navigation links: Stats, Forms, Admins, **Users**, Teachers, Send Reminders, Profile, Results, and Registered Subjects. The 'Users' link is highlighted. The main area has a title 'Add Users'. It contains two input fields: 'Enter New User Name' and 'Enter New User Email'. Below these are two buttons: a blue 'Add' button and a green 'Add Through csv' button. At the bottom are three dropdown menus: 'Category' set to 'Student', 'Semester' set to 'Select Semester', and 'Section' set to 'Select Section'. A red button at the top right says 'Mark Semester End'.

FIGURE 7.10: Add User manually

7.0.11 Add Admin manually/through csv

Allows Admin to add Admin csv/manually. View Figure 7.11 for reference.

The screenshot shows a user interface for adding a new admin. The sidebar on the left is identical to Figure 7.10, with 'Admins' now highlighted. The main area has a title 'Add Admin'. It contains two input fields: 'Enter New Admin Name' and 'Enter New Admin Email'. Below these are two buttons: a blue 'Add' button and a green 'Add Through csv' button. The bottom section is identical to Figure 7.10, with 'Category' set to 'Student', 'Semester' set to 'Select Semester', and 'Section' set to 'Select Section'.

FIGURE 7.11: Add Admin manually/through csv

7.0.12 Add Teacher manually/through csv

Allows Admin to add Teacher csv/manually.View Figure 7.12 for reference.

Sr#	Name	Email	Action
1	Jane Doe	JaneDoe@example.com	
2	Michelle Jones	abc@gmail.com	

FIGURE 7.12: Add Teacher manually/through csv

7.0.13 Send Reminders

Allows Admin to send reminders to selected category.View Figure 7.13 for reference.

FIGURE 7.13: Send Reminders

7.0.14 User Added Email

Sends Email to user when he is added. View Figure 7.14 for reference.

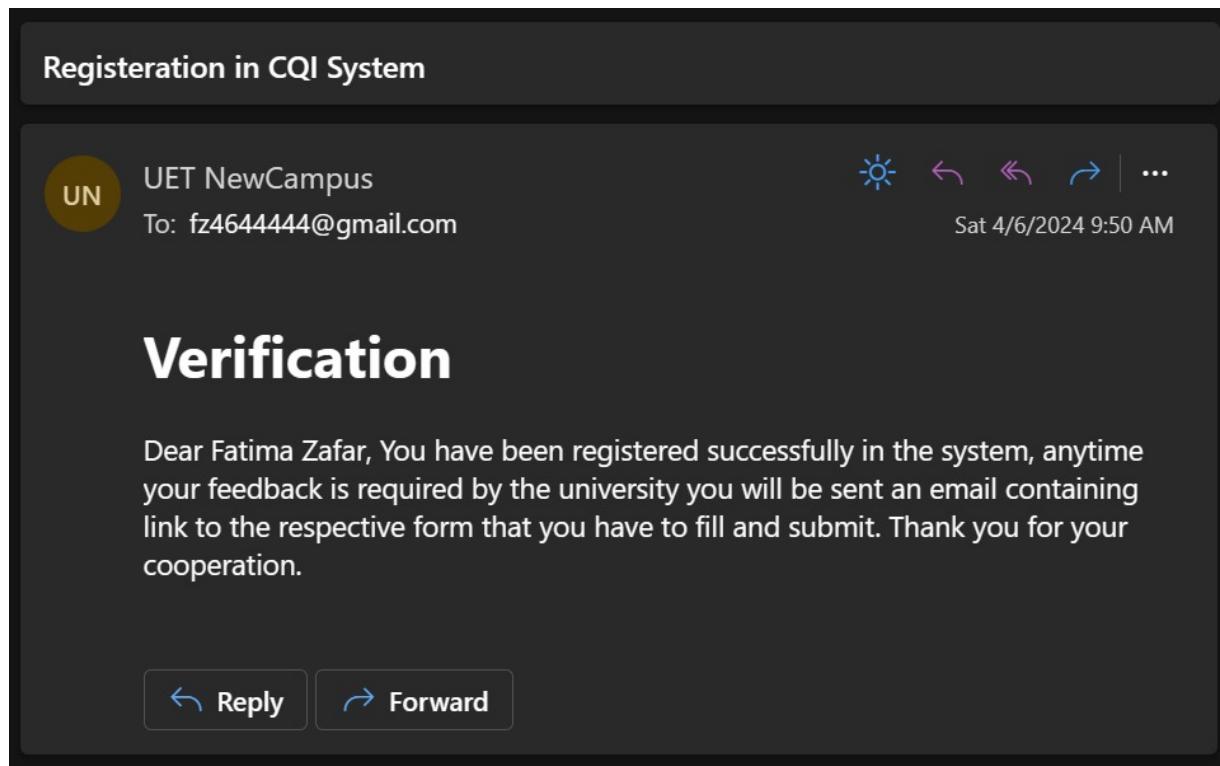


FIGURE 7.14: User Added Email

7.0.15 Fill Form Email

Sends Email to user when form link is to be sent. View Figure 7.15 for reference.

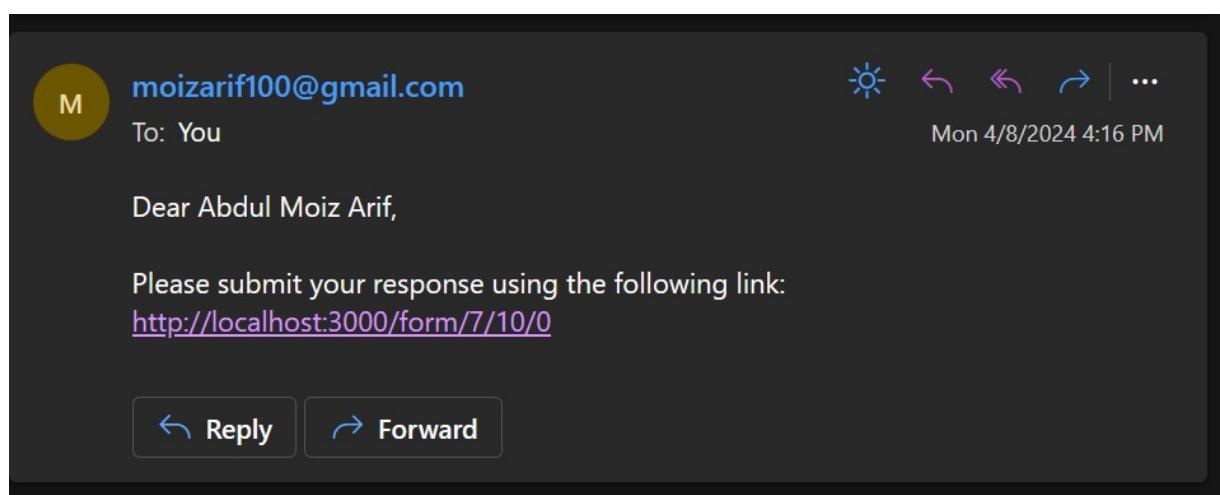


FIGURE 7.15: Fill Form Email

7.0.16 View/Edit Profile

Allows Admin to View/edit Profile. View Figure 7.16 for reference.

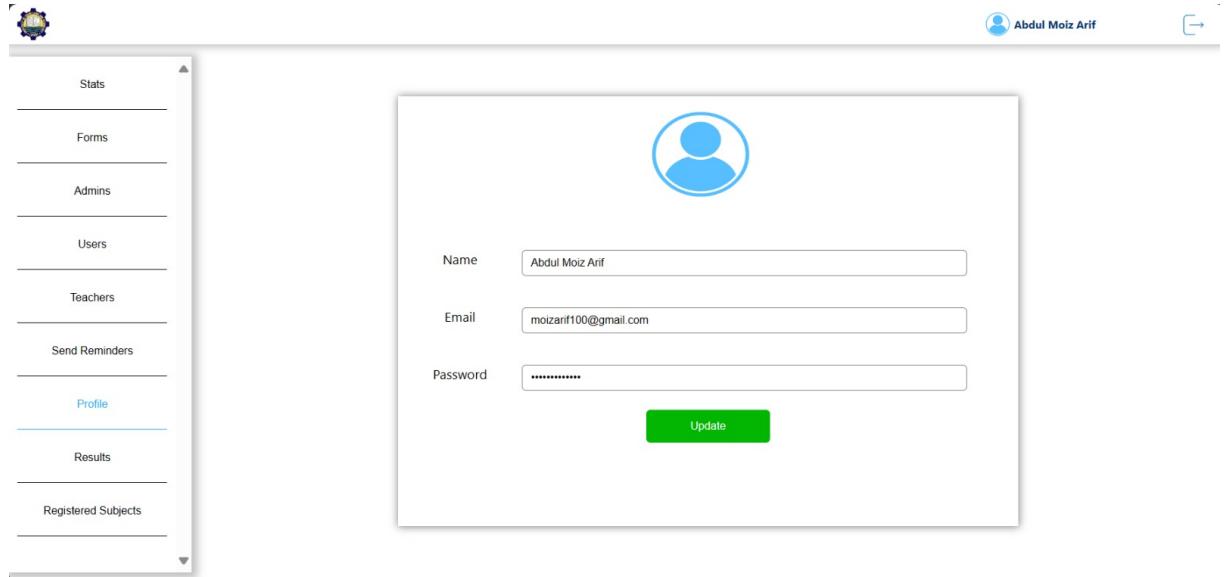


FIGURE 7.16: View/Edit Profile

7.0.17 Change Username Email

Sends Email to Admin when Username is changed. View Figure 7.17 for reference.

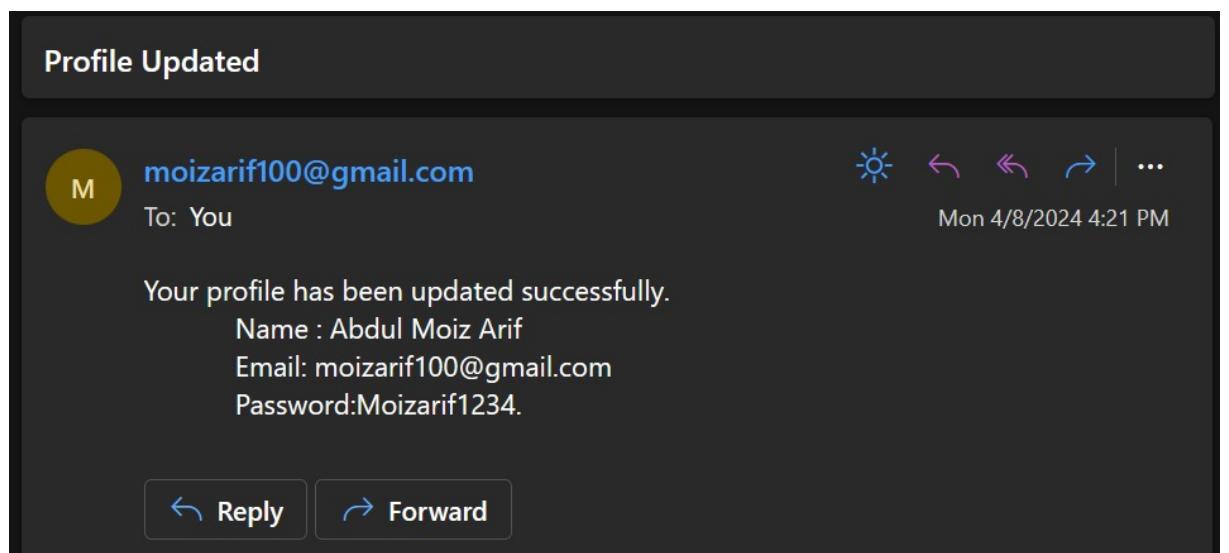


FIGURE 7.17: Change Username Email

7.0.18 Change Password Email

Sends Email to Admin when password is changed. View Figure 7.18 for reference.

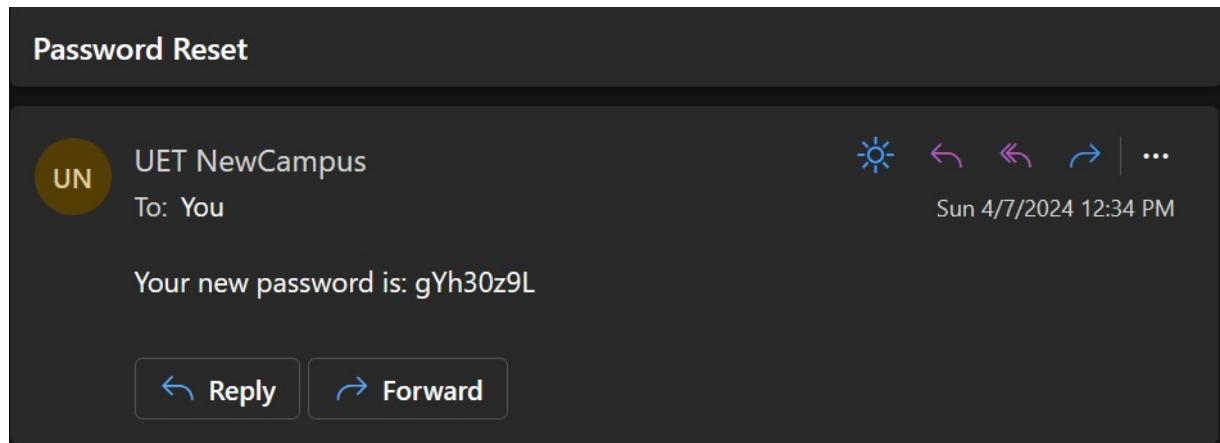


FIGURE 7.18: Change Password Email

7.0.19 View and Download Results

Allows Admin to Download results of selected form. View Figure 7.19 for reference.

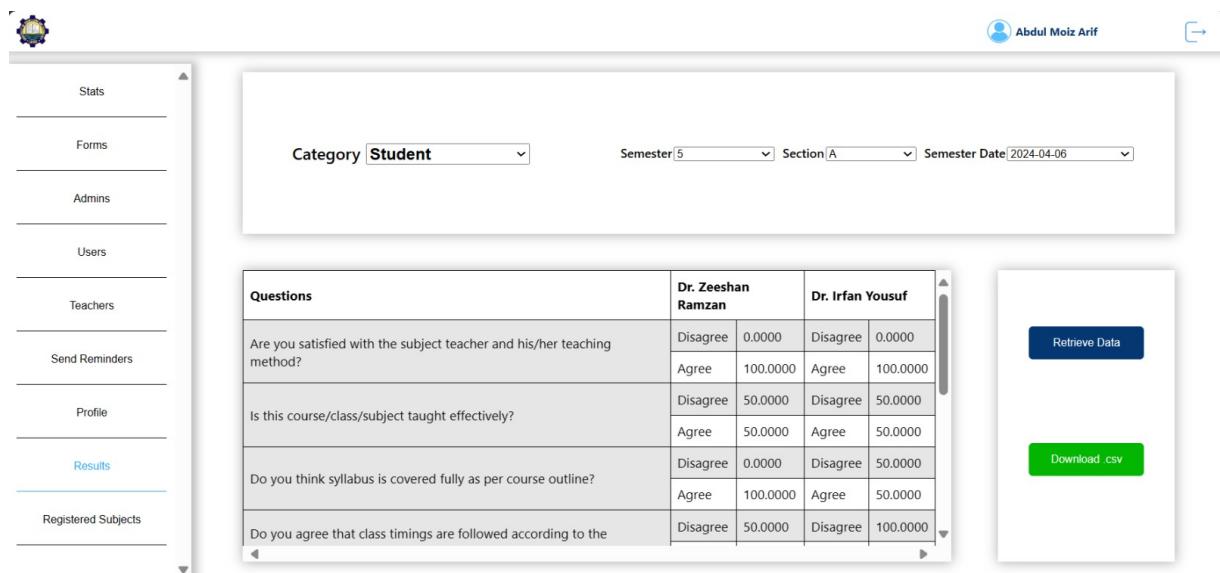


FIGURE 7.19: View/Download Results

7.0.20 Registered Subjects

Displays Admin, details of student registered in specific subject. View Figure 7.20 for reference.

The screenshot shows a user interface for managing registered subjects. On the left, a sidebar menu includes options like Stats, Forms, Admins, Users, Teachers, Send Reminders, Profile, Results, and Registered Subjects. The 'Registered Subjects' option is currently selected. The main content area has two dropdown menus: 'Select Subject: Enterprise Application Development' and 'Select Semester Date: 2024-04-06'. Below these is a table with columns: Name, Email, Semester, and Section. One row is shown for 'Abdul Moiz Arif' with email 'moizarif100@gmail.com', Semester '7', and Section 'A'. To the right of the table are two buttons: 'Retrieve Data' (blue) and 'Download csv' (green).

Name	Email	Semester	Section
Abdul Moiz Arif	moizarif100@gmail.com	7	A

FIGURE 7.20: Registered Subjects

7.0.21 Change Mode

Allows Admin to Change Mode of system. View Figure 7.21 for reference.

The screenshot shows a dashboard titled 'Change Mode' with various data visualizations. On the left, a sidebar menu lists Stats, Forms, Admins, Users, Teachers, Send Reminders, Profile, Results, and Registered Subjects. The 'Results' option is selected. The main area contains three pie charts under 'Course & Teacher Evaluation Survey', 'Alumni Survey', and 'Student Exit Survey'. Below these are two bar charts: 'Semester 5' comparing 'Web Development' and 'Introduction to Data Science' in terms of 'Percentage of Reserved Seats', and 'Semester 6' comparing four subjects (Subject 1 to Subject 4) in the same metric. A note 'No responses yet' is visible above the Semester 6 chart.

FIGURE 7.21: Change Mode

Chapter 8

Conclusions and Future Direction

8.1 Conclusion

The implementation of the "Automated Continuous Quality Improvement" system has significantly enhanced the evaluation process of teachers based on feedback from students. By automating the collection and analysis of data through forms filled by students, we have achieved a more streamlined and efficient approach to assessing teacher performance. Additionally, the system's capability to allow students to register subjects has facilitated a smoother enrollment process, contributing to overall organizational effectiveness.

Through this project, we have demonstrated the potential of leveraging technology to drive continuous quality improvement in educational institutions. The insights gained from student feedback have enabled targeted interventions and improvements in teaching methodologies, ultimately leading to a better learning experience for students.

8.2 Future Direction

Moving forward, the future direction of this project includes:

- Enhanced Data Analytics
- Integration with Learning Management Systems
- Feedback Mechanism Refinement
- Scalability and Expansion

8.2.1 Enhanced Data Analytics

Implementing advanced data analytics techniques to derive deeper insights from the collected feedback data, enabling more informed decision-making and proactive improvements.

8.2.2 Integration with Learning Management Systems

Integrating the system with existing learning management systems to provide a seamless experience for both students and faculty, consolidating data and processes for greater efficiency.

8.2.3 Feedback Mechanism Refinement

Continuously refining the feedback mechanism based on user feedback and evolving educational best practices to ensure the system remains relevant and impactful.

8.2.4 Scalability and Expansion

Scaling the system to accommodate larger user bases and expanding its capabilities to cover additional aspects of quality assessment and improvement across various educational domains.

By focusing on these future directions, we aim to further enhance the educational ecosystem and contribute to ongoing efforts in fostering continuous quality improvement in teaching and learning processes.

References

- [1] Alchemer. <https://www.alchemer.com/surveygizmo-is-now-alchemer/>. Last accessed on February 10, 2024.
- [2] Anthology. <https://www.campuslabs.com/campus-labs-platform/improvement-and-accountability/studies-and-instruments/>. Last accessed on April 26, 2023.
- [3] Cqi system. <https://pubmed.ncbi.nlm.nih.gov/12390261/>, . Last accessed on August 18, 2023.
- [4] Cqi system. <https://www.ncbi.nlm.nih.gov/books/NBK559239/>, . Last accessed on August 18, 2023.
- [5] Crowd signal. <https://crowdsignal.com/>. Last accessed on March 2, 2024.
- [6] Formstack. <https://www.formstack.com/>. Last accessed on February 16, 2024.
- [7] Google forms. <https://www.google.com/forms/about/>. Last accessed on April 26, 2023.
- [8] Json to csv. <https://www.npmjs.com/package/react-json-to-csv>. Last accessed on August 18, 2023.
- [9] Jotform. <https://www.jotform.com/>. Last accessed on February 12, 2024.
- [10] Metrics for learning. <https://www.metricsforlearning.com/>. Last accessed on April 25, 2023.
- [11] Qualtrics. <https://www.qualtrics.com/>. Last accessed on April 25, 2023.
- [12] Quicktap survey. <https://www.quicktapsurvey.com/>. Last accessed on February 11, 2024.

- [13] React hooks. <https://legacy.reactjs.org/docs/hooks-overview.html>. Last accessed on August 18, 2023.
- [14] Surveysparrow. <https://surveysparrow.com/>, . Last accessed on February 12, 2024.
- [15] Survey monkey. <https://www.surveymonkey.com/>, . Last accessed on April 25, 2023.
- [16] Typeform. <https://www.typeform.com/>. Last accessed on March 18, 2024.
- [17] Zohosurvey. <https://www.zoho.com/survey/>. Last accessed on February 15, 2024.
- [18] Getfeedback. <https://www.getfeedback.com/en/>. Last accessed on February 2, 2024.
- [19] Microsoft forms. <https://forms.office.com/Pages/DesignPageV2.aspx?subpage=creationv2>. Last accessed on February 26, 2024.
- [20] Question pro. <https://www.questionpro.com/tour/>. Last accessed on February 8, 2024.
- [21] Smart survey. <https://www.smartsurvey.co.uk/>. Last accessed on February 20, 2024.
- [22] Sogosurvey. <https://www.sogalytics.com/static/welcome.aspx>. Last accessed on February 22, 2024.
- [23] Survey anyplace. <https://www.discovercloud.com/products/survey-anyplace#/overview>, . Last accessed on February 20, 2024.
- [24] Survey legend. <https://www.surveylegend.com/>, . Last accessed on February 22, 2024.
- [25] Survey planet. <https://surveyplanet.com/>, . Last accessed on February 8, 2024.
- [26] J. Antonycand R. J. M. M. Does B. A. Lameijera, H. Boerb et al. Continuous improvement implementation models: a reconciliation and holistic metamodel. *PRODUCTION PLANNING CONTROL*, 34(11): 1062–1081, 2023. Last accessed on Feb 16, 2024.

- [27] St. Petersburg Bataev A. V., I S Davydov et al. The role of automation in improving the quality of enterprise business processes. *IOP Conference Series: Materials Science and Engineering*, 986, 2020. Last accessed on Feb 19, 2024.
- [28] Steve Graham and Karen R Harris. Self-regulation and strategy instruction for students who find writing and learning challenging. *The science of writing: Theories, methods, individual differences, and applications*, pages 347–360, 1996. Last accessed on May 16, 2023.
- [29] Robert V. Hogg and Mary C. Hogg. Continuous quality improvement in higher education. *International Statistical Review / Revue Internationale de Statistique*, 63(1):35–48, 1995. ISSN 03067734, 17515823. URL <http://www.jstor.org/stable/1403776>. Last accessed on May 15, 2023.
- [30] Madeleine Rohlin, Rob MH Schaub, Peter Holbrook, Edvitar Leibur, Gérard Lévy, Lenka Roubalikova, Maria Nilner, Valerie Roger-Leroi, Gunter Danner, Haluk Iseri, et al. 2.2 continuous quality improvement. *European journal of dental education*, 6:67–77, 2002. Last accessed on May 16, 2023.
- [31] KY Tshai, J-H Ho, EH Yap, and HK Ng. Outcome-based education—the assessment of programme educational objectives for an engineering undergraduate degree. *Engineering Education*, 9(1):74–85, 2014. Last accessed on May 15, 2023.