

Automated CQI

Test Document



Session: 2020-2024

Submitted by:

Rana Abdul Moiz Arif 2020-CS-674

Fatima Zafar 2020-CS-664

Munazza Rafique 2020-CS-687

Supervised by:

Dr. Zeeshan Ramzan

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

Declarations

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.

Signed : _____

Date: _____

Acknowledgments

We would like to express my special thanks of gratitude to our teacher Sir Zeeshan Ramzan as well as our Head of Department Dr.Shahzad who gave us the golden opportunity to do this wonderful project on the topic Automated CQI, which also helped me in doing a lot of Research and we came to know about so many new things I am really thankful to them.

Contents

List of Tables	v
1 Functional vs. Non-functional Testing	1
1.1 Unit Testing	2
1.1.1 Log-in page Testing	2
1.1.2 Top Nav-bar Testing	3
1.1.3 Side Nav-bar Testing	3
1.1.4 Admin Dashboard Testing	5
1.1.5 Admin page Testing	5
1.1.6 Student page Testing	6
1.1.7 Teacher page Testing	7
1.1.8 Reminder Testing	7
1.1.9 Results Testing	8
1.1.10 Profile Testing	8
1.1.11 Fill Form Testing	9
1.2 Integration Testing	10
1.2.1 Sign in	10
1.2.2 View Stats	11
1.2.3 Create Forms	11
1.2.4 View and Download Results	11
1.2.5 Add user	11
1.2.6 Add Teacher	12
1.2.7 Add Admin	12
1.2.8 Fill Form	12
1.2.9 Send Reminders	13
1.2.10 Edit Profile	13
1.3 System Testing	14
1.3.1 Sign in Testing	14
1.3.2 Create and Add User Testing	15
1.3.3 Create and Add Teacher Testing	16
1.3.4 Create and Add Admin Testing	17
1.3.5 View and Download Results Testing	18
1.3.6 Select Category and Send Reminders Testing	19
1.3.7 View and Edit Profile Testing	20
1.3.8 Fill Form Testing	21

1.4	Acceptance Testing	22
1.4.1	Sign in Testing	22
1.4.2	Create User Testing	23
1.4.3	Create Teacher Testing	24
1.4.4	Create Admin Testing	25
1.4.5	Download Results Testing	26
1.4.6	Send Reminders Testing	27
1.4.7	Edit Profile Testing	28
1.4.8	Change Mode	29
1.5	Performance Testing	29
1.5.1	Load testing	29
1.5.2	Stress testing	30
1.5.3	Endurance testing	31
1.5.4	Spike testing	31
1.6	Security Testing	32
1.6.1	Integrity	32
1.6.2	Confidentiality	33
1.6.3	Authentication	33
1.6.4	Authorization	33
1.6.5	Availability	33
1.7	Usability Testing	34
1.8	Compatibility Testing	34

List of Tables

1.1	Email Text Field Testing	2
1.2	Password Text Field Testing	2
1.3	Log-in Button Testing	3
1.4	Back Button Testing	3
1.5	Profile Display Testing	3
1.6	Log-out BUtton Testing	3
1.7	Dashboard Button Testing	4
1.8	Survey Button Testing	4
1.9	Students Button Testing	4
1.10	Teachers Button Testing	4
1.11	Admin Button Testing	4
1.12	Reminder Button Testing	5
1.13	Results Button Testing	5
1.14	Profile Button Testing	5
1.15	Survey Graph Testing	5
1.16	Subject Registration Graph Testing	5
1.17	Admin List Testing	6
1.18	Edit Admin detail Testing	6
1.19	Add/Remove Admin Testing	6
1.20	Student List Testing	6
1.21	Edit Student detail Testing	6
1.22	Add/Remove Student Testing	7
1.23	Teacher List Testing	7
1.24	Edit Teacher detail Testing	7
1.25	Add/Remove Teacher Testing	7
1.26	Selected Category Testing	8
1.27	Reminder button Testing	8
1.28	View result Testing	8
1.29	Download result button Testing	8
1.30	View/edit Profile picture Testing	9
1.31	View/edit Email Testing	9
1.32	View/edit Password Testing	9
1.33	Save changes button Testing	9
1.34	Progress bar Testing	10
1.35	Text Field Testing	10

1.36 Send Response Button Testing	10
1.37 Sign in Testing	10
1.38 View Stats Testing	11
1.39 Create Forms Testing	11
1.40 Results Testing	11
1.41 Add User Testing	12
1.42 Add Teacher Testing	12
1.43 Add Admin Testing	12
1.44 Fill form Testing	12
1.45 Send reminders Testing	13
1.46 Add Admin Testing	13
1.47 Sign in Testing	14
1.48 Create user Testing	15
1.49 Create user Testing	16
1.50 Create Admin Testing	17
1.51 Download result Testing	18
1.52 Send Reminders Testing	19
1.53 Edit Profile Testing	20
1.54 Fill Form Testing	21
1.55 Sign in Testing	22
1.56 Create user Testing	23
1.57 Create user Testing	24
1.58 Create Admin Testing	25
1.59 Download result Testing	26
1.60 Send Reminders Testing	27
1.61 Edit Profile Testing	28
1.62 Change mode Testing	29
1.63 Load testing	30
1.64 Stress testing	31
1.65 Endurance testing	31
1.66 Spike testing	32

Chapter 1

Functional vs. Non-functional Testing

The goal of utilizing numerous testing methodologies in your development process is to make sure your software can successfully operate in multiple environments and across different platforms. These can typically be broken down between functional and non-functional testing. Functional testing involves testing the application against the business requirements. It incorporates all test types designed to guarantee each part of a piece of software behaves as expected by using uses cases provided by the design team or business analyst. These testing methods are usually conducted in order and include:

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

Non-functional testing methods incorporate all test types focused on the operational aspects of a piece of 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.

1.1 Unit Testing

Unit testing involves the testing of each unit or an individual component of the software application. It is the first level of functional testing. The aim behind unit testing is to validate unit components with its performance.

1.1.1 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.

TABLE 1.1: Email Text Field Testing

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

TABLE 1.2: Password Text Field Testing

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

TABLE 1.3: Log-in Button Testing

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

1.1.2 Top Nav-bar Testing

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

- Back Button.
- Profile Display.
- Log-out Button.

TABLE 1.4: Back Button Testing

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

TABLE 1.5: Profile Display Testing

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

TABLE 1.6: Log-out BUTton Testing

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

1.1.3 Side Nav-bar Testing

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

- Dashboard Option.

- Surveys Option.
- Students Option.
- Teachers Option.
- Admins Option.
- Reminders Option.
- Results Option.
- Profile Option.

TABLE 1.7: Dashboard Button Testing

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

TABLE 1.8: Survey Button Testing

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

TABLE 1.9: Students Button Testing

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

TABLE 1.10: Teachers Button Testing

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

TABLE 1.11: Admin Button Testing

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

TABLE 1.12: Reminder Button Testing

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

TABLE 1.13: Results Button Testing

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

TABLE 1.14: Profile Button Testing

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

1.1.4 Admin Dashboard Testing

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

- Display Survey graphs.
- Display Subject Registration graphs.

TABLE 1.15: Survey Graph Testing

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

TABLE 1.16: Subject Registration Graph Testing

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

1.1.5 Admin page Testing

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

- Display Admin List.
- Edit Admin details.
- Add / Remove Admin manually.

TABLE 1.17: Admin List Testing

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

TABLE 1.18: Edit Admin detail Testing

Sr No	Test Data	Expected Result	Test Result
1	Enter Admin email	Display details to edit if Admin exists	Successful
2	Enter Admin email	Invalid email if not found	Successful

TABLE 1.19: Add/Remove Admin Testing

Sr No	Test Data	Expected Result	Test Result
1	Enter Admin details	Add Admin if details are valid	Successful
2	Enter Admin details	Admin not added if details are invalid	Rejected
3	Enter Admin details	Remove Admin if details are valid	Successful
4	Enter Admin details	Admin not removed if details are invalid	Rejected

1.1.6 Student page Testing

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

- Display Student List.
- Edit Student details.
- Add / Remove Student manually.

TABLE 1.20: Student List Testing

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

TABLE 1.21: Edit Student detail Testing

Sr No	Test Data	Expected Result	Test Result
1	Enter Student email	Display details to edit if Student exists	Successful
2	Enter Student email	Invalid email if not found	Successful

TABLE 1.22: Add/Remove Student Testing

Sr No	Test Data	Expected Result	Test Result
1	Enter Student details	Add Student if details are valid	Successful
2	Enter Student details	Student not added if details are invalid	Rejected
3	Enter Student details	Remove Student if details are valid	Successful
4	Enter Student details	Student not removed if details are invalid	Rejected

1.1.7 Teacher page Testing

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

- Display Teacher List.
- Edit Teacher details.
- Add / Remove Teacher manually.

TABLE 1.23: Teacher List Testing

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

TABLE 1.24: Edit Teacher detail Testing

Sr No	Test Data	Expected Result	Test Result
1	Enter Teacher email	Display details to edit if Teacher exists	Successful
2	Enter Teacher email	Invalid email if not found	Successful

TABLE 1.25: Add/Remove Teacher Testing

Sr No	Test Data	Expected Result	Test Result
1	Enter Teacher details	Add Teacher if details are valid	Successful
2	Enter Teacher details	Teacher not added if details are invalid	Rejected
3	Enter Teacher details	Remove Teacher if details are valid	Successful
4	Enter Teacher details	Teacher not removed if details are invalid	Rejected

1.1.8 Reminder Testing

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

- Selected Category.
- Send reminder button.

TABLE 1.26: Selected Category Testing

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

TABLE 1.27: Reminder button Testing

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

1.1.9 Results Testing

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

- View Results.
- Download Results button.

TABLE 1.28: View result Testing

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

TABLE 1.29: Download result button Testing

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

1.1.10 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.

TABLE 1.30: View/edit Profile picture Testing

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

TABLE 1.31: View/edit Email Testing

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

TABLE 1.32: View/edit Password Testing

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

TABLE 1.33: Save changes button Testing

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

1.1.11 Fill Form Testing

This page is used to Fill Form.

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

TABLE 1.34: Progress bar Testing

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

TABLE 1.35: Text Field Testing

Sr No	Test Data	Expected Result	Test Result
1	No comments.	Text Accepted	Accepted
2		Empty text field not accepted.	Rejected

TABLE 1.36: Send Response Button Testing

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

1.2 Integration Testing

After each unit is thoroughly tested, it is integrated with other units to create modules or components that are designed to perform specific tasks or activities. These are then tested as group through integration testing to ensure whole segments of an application behave as expected (i.e, the interactions between units are seamless). These tests are often framed by user scenarios, such as logging into an application or opening files. Integrated tests can be conducted by either developers or independent testers and are usually comprised of a combination of automated functional and manual tests.

1.2.1 Sign in

User must be able to Sign in and View profile.

TABLE 1.37: Sign in 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

1.2.2 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

TABLE 1.38: View Stats 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

1.2.3 Create Forms

User must be able to create form and surveys for selected category.

TABLE 1.39: Create Forms Testing

Test case	Description	Expected Result	Actual Result	Pass/Fail
1	Category is selected	User must be able to create form/survey	Able to create form	Pass
2	Category is not selected	User must not be able to create form/survey	Cannot create form	Pass

1.2.4 View and Download Results

User must be able to View and Download Results.

TABLE 1.40: Results 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

1.2.5 Add user

User must be able to Add Student with specific category.

TABLE 1.41: Add User 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

1.2.6 Add Teacher

User must be able to Add Teachers.

TABLE 1.42: Add Teacher 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

1.2.7 Add Admin

User must be able to add Admins.

TABLE 1.43: Add Admin 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

1.2.8 Fill Form

User must be able to fill forms.

TABLE 1.44: Fill form 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

1.2.9 Send Reminders

User must be able to Send reminders.

TABLE 1.45: Send reminders 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

1.2.10 Edit Profile

User must be able to edit profile.

TABLE 1.46: Add Admin 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

1.3 System Testing

System testing is a black box testing method used to evaluate the completed and integrated system, as a whole, to ensure it meets specified requirements. The functionality of the software is tested from end-to-end.

1.3.1 Sign in Testing

TABLE 1.47: Sign in Testing

Test case ID	ID_001	Test case Description	Sign in to Account		
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	3/3/2024	Pass/Fail/Not executed	Pass
----------------------	--------	--------------------	----------	-------------------------------	------

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.
-----------------------	---

1.3.2 Create and Add User Testing

TABLE 1.48: Create user 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	Semester=8	
				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.				

1.3.3 Create and Add Teacher Testing

TABLE 1.49: Create user Testing

Test case ID		ID_003	Test case Description		Create new Teacher	
Created by		Moiz	Reviewed by		Fatima	Version1.0.1
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 executedPass	
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	Semester=8	
				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.				

1.3.4 Create and Add Admin Testing

TABLE 1.50: Create Admin Testing

Test case ID		ID_004	Test case Description		Create new Admin									
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	1/3/2024	Pass/Fail/Not executed Pass									
S#	Prerequisites		<table><tr><td>S#</td><td>Test data</td></tr><tr><td>1</td><td>Upload csv=Uploaded</td></tr><tr><td>2</td><td>Name=Fatima</td></tr><tr><td>3</td><td>Email=fz1@gmail.com</td></tr></table>				S#	Test data	1	Upload csv=Uploaded	2	Name=Fatima	3	Email=fz1@gmail.com
S#	Test data													
1	Upload csv=Uploaded													
2	Name=Fatima													
3	Email=fz1@gmail.com													
1	Must have an internet connection													
2	User is signed in													
3	User has uploaded csv or entered data manually													
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.												

1.3.5 View and Download Results Testing

TABLE 1.51: Download result Testing

Test case ID		ID_005	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		<table><tr><td>S#</td><td>Test data</td></tr><tr><td>1</td><td>Category=Students</td></tr><tr><td>2</td><td>Form Name=Course and Teacher evaluation Form</td></tr></table>				S#	Test data	1	Category=Students	2	Form Name=Course and Teacher evaluation Form
S#	Test data											
1	Category=Students											
2	Form Name=Course and Teacher evaluation Form											
1	Must have an internet connection											
2	User is signed in											
3	User has entered category for results to download											
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.										

1.3.6 Select Category and Send Reminders Testing

TABLE 1.52: Send Reminders Testing

Test case ID		ID_006	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.				

1.3.7 View and Edit Profile Testing

TABLE 1.53: Edit Profile Testing

Test case ID		ID_007	Test case Description		Edit Profile											
Created by		Munazza	Reviewed by		Moiz	Version1.0.1										
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 executedPass											
S#	Prerequisites		<table><tr><td>S#</td><td>Test data</td></tr><tr><td>1</td><td>Profile picture=uploaded</td></tr><tr><td>2</td><td>Name=Munazza</td></tr><tr><td>3</td><td>Email=mz231@gmail.com</td></tr><tr><td>4</td><td>Password=Aa123@</td></tr></table>				S#	Test data	1	Profile picture=uploaded	2	Name=Munazza	3	Email=mz231@gmail.com	4	Password=Aa123@
S#	Test data															
1	Profile picture=uploaded															
2	Name=Munazza															
3	Email=mz231@gmail.com															
4	Password=Aa123@															
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.														

1.3.8 Fill Form Testing

TABLE 1.54: Fill Form Testing

Test case ID		ID_008	Test case Description		Fill Form	
Created by		Moiz	Reviewed by		Fatima	Version1.0.1
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		S#	Test data		
1	Have internet connection		1	Progress bar=FILLED		
2	User must get link to form through mail.		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.				

1.4 Acceptance Testing

Acceptance testing is the last phase of functional testing and is used to assess whether or not the final piece of software is ready for delivery. It involves ensuring that the product is in compliance with all of the original business criteria and that it meets the end user's needs. This requires the product be tested both internally and externally, meaning you'll need to get it into the hands of your end users for beta testing. Beta testing is key to getting real feedback from potential customers and can address any final usability concerns.

1.4.1 Sign in Testing

TABLE 1.55: Sign in Testing

Test case ID		ID_001	Test case Description		Sign in to Account	
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	3/3/2024	Pass/Fail/Not executed Pass	
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.				

1.4.2 Create User Testing

TABLE 1.56: Create user Testing

Test case ID		ID_002	Test case Description		Create new user	
Created by		Moiz	Reviewed by		Munazza	Version1.0.1
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 executedPass	
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	Semester=8	
				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.				

1.4.3 Create Teacher Testing

TABLE 1.57: Create user Testing

Test case ID		ID.003	Test case Description		Create new Teacher	
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	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	Semester=8		
			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.				

1.4.4 Create Admin Testing

TABLE 1.58: Create Admin Testing

Test case ID		ID_004	Test case Description		Create new Admin									
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	1/3/2024	Pass/Fail/Not executed Pass									
S#	Prerequisites		<table><tr><td>S#</td><td>Test data</td></tr><tr><td>1</td><td>Upload csv=Uploaded</td></tr><tr><td>2</td><td>Name=Fatima</td></tr><tr><td>3</td><td>Email=fz1@gmail.com</td></tr></table>				S#	Test data	1	Upload csv=Uploaded	2	Name=Fatima	3	Email=fz1@gmail.com
S#	Test data													
1	Upload csv=Uploaded													
2	Name=Fatima													
3	Email=fz1@gmail.com													
1	Must have an internet connection													
2	User is signed in													
3	User has uploaded csv or entered data manually													
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.												

1.4.5 Download Results Testing

TABLE 1.59: Download result Testing

Test case ID		ID_005	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		<table><tr><td>S#</td><td>Test data</td></tr><tr><td>1</td><td>Category=Students</td></tr><tr><td>2</td><td>Form Name=Course and Teacher evaluation Form</td></tr></table>				S#	Test data	1	Category=Students	2	Form Name=Course and Teacher evaluation Form
S#	Test data											
1	Category=Students											
2	Form Name=Course and Teacher evaluation Form											
1	Must have an internet connection											
2	User is signed in											
3	User has entered category for results to download											
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.										

1.4.6 Send Reminders Testing

TABLE 1.60: Send Reminders Testing

Test case ID		ID_006	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.				

1.4.7 Edit Profile Testing

TABLE 1.61: Edit Profile Testing

Test case ID		ID_007	Test case Description		Edit Profile											
Created by		Munazza	Reviewed by		Moiz	Version1.0.1										
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 executedPass											
S#	Prerequisites		<table><tr><td>S#</td><td>Test data</td></tr><tr><td>1</td><td>Profile picture=uploaded</td></tr><tr><td>2</td><td>Name=Munazza</td></tr><tr><td>3</td><td>Email=mz231@gmail.com</td></tr><tr><td>4</td><td>Password=Aa123@</td></tr></table>				S#	Test data	1	Profile picture=uploaded	2	Name=Munazza	3	Email=mz231@gmail.com	4	Password=Aa123@
S#	Test data															
1	Profile picture=uploaded															
2	Name=Munazza															
3	Email=mz231@gmail.com															
4	Password=Aa123@															
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.														

1.4.8 Change Mode

TABLE 1.62: Change mode Testing

Test case ID		ID_008	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		<table><tr><td>S#</td><td>Test data</td></tr><tr><td>1</td><td>Toggle button=ON</td></tr></table>				S#	Test data	1	Toggle button=ON
S#	Test data									
1	Toggle button=ON									
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.								

1.5 Performance Testing

Performance testing is a non-functional testing technique used to determine how an application will behave under various conditions. The goal is to test its responsiveness and stability in real user situations. Performance testing can be broken down into four types:

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

1.5.1 Load testing

Load testing is the process of putting increasing amounts of simulated demand on your software, application, or website to verify whether or not it can handle what it's designed to handle.

TABLE 1.63: 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

1.5.2 Stress testing

Stress testing takes this a step further and is used to gauge how your software will respond at or beyond its peak load. The goal of stress testing is to overload the application on purpose until it breaks by applying both realistic and unrealistic load scenarios. With stress testing, you'll be able to find the failure point of your piece of software.

TABLE 1.64: 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

1.5.3 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.

TABLE 1.65: 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.

1.5.4 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.

TABLE 1.66: 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.

1.6 Security Testing

With the rise of cloud-based testing platforms and cyberattacks, there is a growing concern and need for the security of data being used and stored in software. Security testing is a non-functional software testing technique used to determine if the information and data in a system is protected. The goal is to purposefully find loopholes and security risks in the system that could result in unauthorized access to or the loss of information by probing the application for weaknesses. There are multiple types of this testing method, each of which aimed at verifying five basic principles of security:

- Integrity
- Confidentiality
- Authentication
- Authorization
- Availability

1.6.1 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 successfully logged in.

1.6.2 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.

1.6.3 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 has the admin information in encrypted form.

1.6.4 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.

1.6.5 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.

1.7 Usability Testing

Usability testing is a testing method that measures an application's ease-of-use from the end-user perspective and is often performed during the system or acceptance testing stages. The goal is to determine whether or not the visible design and aesthetics of an application meet the intended workflow for various processes, such as logging into an application.

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.

1.8 Compatibility Testing

Compatibility testing is used to gauge how an application or piece of software will work in different environments. It is used to check that your product is compatible with multiple operating systems, platforms, browsers, or resolution configurations. The goal is to ensure that your software's functionality is consistently supported across any environment you expect your end users to be using. We performed compatible testing on our project by running it on different systems.