

DyslexAI

Test Plan Documentation

Project Members:

Saba Begum

Mahbubur Khan

Aaron Perel

Ishmael Gonzales

Revision History:

Date:	Description:	Author:	Comments:
06/07/2024	Version 1.0	Saba Begum	First Draft
		Mahbubur Khan	
		Aaron Perel	
		Ishmael Gonzales	
07/07/2024	Version 1.1	Saba Begum	Second Draft
		Mahbubur Khan	
		Aaron Perel	
		Ishmael Gonzales	
09/07/2024	Version 1.2	Saba Begum	Final Draft
		Mahbubur Khan	
		Aaron Perel	
		Ishmael Gonzales	

Document Approval:

Signature:	Printed Name:	Title:	Date:
Ishmool	Ishmael Gonzales	Documentation Lead	06/07/2024
Isiiiiaei		Documentation Load	00,07,2021
Gonzales			
Ishmael	Ishmael Gonzales	Documentation Lead	08/07/2024
Gonzales			
Ishmael	Ishmael Gonzales	Documentation Lead	09/07/2024
Gonzales			

Table Of Contents

1.	Introduction4
	• Purpose of the Document
2.	Test Objective4
	 Functional Testing
	• Performance Testing
	 Security Testing
3.	Test Strategy5
	 Developer Testing
	 Front-end Testing
	 Integration Testing
	• User Testing
	• System Testing
4.	Test Scope5
	• Front-end Testing
	 Backend Testing
	 API Endpoint Testing
	 Data Formatting Testing
5.	Test
	Deliverables
	• 5.1 Functional Test Cases
	 Validate User Registration
	 Validate User Login
	 Validate Password Reset
	 Validate Profile Update
	 Validate E-Book Search and Import
	 Validate Deaf Mode Functionality
	 Validate Blind Mode Functionality
	 Validate Customizable Reading Interface
	 Validate Book Recommendations
	• Validate LLM and AI Integration
	• 5.2 Non-Functional Test Cases
	Validate Performance
	 Validate Database Information Updating
	 Validate Response Time for Non-AI API Calls
	 Validate Server Error Handling
	• Validate Uptime and Availability
	• Validate Portability
	• Validate Scalability
	• 5.3 Test Scripts and Mock Data
	• Test Script TC1
	• Test Script TC2
	• Test Script TC3

	 Test Script TC5 	
	 Test Script TC6 	
	 Test Script TC8 	
	 Test Script TC10 	
6.	Test Schedule	1
	 Detailed Testing Timeline 	
7.	Test Environment	5
	• Testing Tools and Setup	
8.	Test Entry and Exit Criteria	5
	 Entry Criteria 	
	• Exit Criteria	
9.	Test Pass and Fail Criteria2	5
	• Definition of Pass and Fail	
10.	Test Suspension and Resumption Criteria25	5
	• Suspension Criteria	
	Resumption Criteria	
11.	Test Design and Execution	5
	• Manual Testing	
	 Automated Testing 	
12.	Test Data & Defect Management	Ś
	• Test Data Management	
	 Defect Management Process 	
13.	Risk Analysis	Ś
	 Identified Risks 	
	• Mitigation Strategies	
14.	Roles and Responsibilities	7
	• Team Roles	
	 Individual Responsibilities 	

1. Introduction:

The purpose of this document is to outline the testing plan for the DyslexAI web application, detailing the strategies, scope, and deliverables involved in ensuring the application meets its

functional, performance, and security requirements. This plan serves as a comprehensive guide for the testing team and developers to understand the testing objectives and processes.

2. Test Objective:

The tests outlined in this document will be used to improve and test the functionality and the responsiveness of our request structures and system overall. Functionality such as user authentication, user signup, google authentication, response time to import e-books, response time to generate images, responsiveness of the ResponsiveVoice functionality, saving and changing user settings and preferences and how data is iterated and formatted after retrieval from Firebase. Due to a mostly serverless backend our tests will focus on the structure of our requests and how that affects response time and the display of data once retrieved.

3. Test Strategy:

Developer testing will be our main approach during the testing process of all our serverless functions. This approach is more suitable for the developers as all are familiar with the code base and the aspects, they have created in the code base as well as the documentation. First testing will be done on all front-end aspects. This includes triggers, state changes, redirects, variable based rendering logic, and all displayable objects that are interactable. This ensures we catch any bugs that are prevalent in user interactable objects such as buttons, input fields and text are dealt with before they cause errors that are detrimental to the state of the application. At the same time integration testing will be taking place to validate the responsiveness of the application such as API calls, data filtering, data sorting, data creation and data retrieval via API calls. Simultaneous testing of both frontend aspects and "backend" functions / integration with the frontend will allow us to speed up the testing process as well as ensure necessary components work with their interlinked serverless functions. User testing will begin after all developer testing has been completed and confirmed by all team members. For this testing a "test user" will be introduced who has zero knowledge of the codebase and can use the application as a standard user would. This will allow us to see what issues indirectly related to the codebase are present in the application. The reason for this is as developers some may not see certain issues outside the code such as button spamming, and more. System testing will take place once all other testing has been completed since uptime data during times of testing may be skewed due to issues that arise during the testing process.

4. Test Scope:

For the front end testing of all interactable objects such as when users click the login button, signup button, the change in what is rendered based on if they are logging in signing up or changing their password. Also on the front-end, testing of all state variables and local caching will be validated to ensure minimum load and response time and to minimize visual bugs due to slow updates of these state variables. Triggers for backend functions from the frontend interactable objects are encompassed in this as well. For the "backend" example requests will be sent that mimic a request from the application to ensure data is being retrieved or stored

properly. All API endpoints being used in our serverless "backend" will be tested to test the data we are sending them as part of the request, as well as the data received from these APIs. Lastly the formatting of data before it is used in the front end will be tested as well to make sure all data is correctly displayed, and no errors arise due to missing or incomplete fields that may be required by the front end.

5. Test Deliverables:

5.1 Functional Test Cases:

ID	TC 2
Title	Validate User Registration
Description	Tests the user registration functionality to ensure that a new user can be successfully created.
Test Type	Unit
Precondition	 The DyslexAI application is accessible. The user registration page is loaded.
Test Steps	 Launch the DyslexAI application. Navigate to the user registration page. Enter the following details First Name "Mahbub" Last Name "Khan" Username: "Mahbubkh12" Email khanmahbub12@example.com Password "Mahbub12" Confirm Password "Mahbub12"
Expected Results	 The user should be able to register successfully. A confirmation message should be displayed after registration. The user should be able to redirect to the login page.
Priority	High
Pass/Fail Criteria	 Pass: The user receives a confirmation message and is redirected to the login page. The user ID should be created and stored in the database. Fail: The user receives an error message or is not redirected to the login page. The user receives an error message indicating failure to register,
	for example Username already taken or Invalid email format.3. Passwords are stored in plain text.

ID TC2 Title Validate User Registration Without First Name
--

Description	Tests	the user registration functionality to ensure that a new user cannot
	succes	stully create an account without their first name.
Test Type	Unit	
Precondition	1.	The DyslexAI application is accessible.
	2.	The user registration page is loaded.
Test Steps	1.	Launch the DyslexAI application.
	2.	Navigate to the user registration page.
	3.	Enter the following details
	4.	First Name ""
	5.	Last Name "Khan"
	6.	Username: "Mahbubkh12"
	7.	Email khanmahbub12@example.com
	8.	Password "Mahbub12"
	9.	Confirm Password "Mahbub12"
Expected Results	1.	The user should not be able to register successfully without
1		entering a first name.
	2.	An error message should be displayed indicating that the first
		name is required.
	3.	The user should remain on the registration page without being
		redirected to the login page.
Priority	High	
Pass/Fail Criteria	Pass:	
	1.	The user was not able to register the account without entering
		their first name.
	2	An appropriate error message should be displayed such as "First
		name is required "
	3	The error message should be clearly visible and not overlap with
	5.	other UI elements
	Fail	
	1	The user receives an error message and remains on the
	1.	registration nage
	2	The user is redirected to the login page or any other page
	2.	indicating successful registration
	3	No error message is displayed when the registration is attempted
	5.	without a first name.

ID	TC3	Title	Validate User Registration Without Last Name
Descriptio	n	Tests the user registration functionality to ensure that a new user cannot successfully create an account without their last name.	
Test Type		Unit	
Precondition1. The DyslexAI application is accessible. 2. The user registration page is loaded.		he DyslexAI application is accessible. he user registration page is loaded.	
Test Steps		1. L	aunch the DyslexAI application.

	2.	Navigate to the user registration page.
	3.	Enter the following details
	4.	First Name "Mahbub"
	5.	Last Name ""
	6.	Username: "Mahbubkh12"
	7.	Email khanmahbub12@example.com
	8.	Password "Mahbub12"
	9.	Confirm Password "Mahbub12"
Expected Results	1.	The user should not be able to register successfully without
-		entering a last name.
	2.	An error message should be displayed indicating that the last
		name is required.
	3.	The user should remain on the registration page without being
		redirected to the login page.
Priority	High	
Priority Pass/Fail Criteria	High Pass:	
Priority Pass/Fail Criteria	High Pass: 1.	The user was not able to register the account without entering
Priority Pass/Fail Criteria	High Pass: 1.	The user was not able to register the account without entering their last name.
Priority Pass/Fail Criteria	High Pass: 1. 2.	The user was not able to register the account without entering their last name. An appropriate error message should be displayed, such as "Last
Priority Pass/Fail Criteria	High Pass: 1. 2.	The user was not able to register the account without entering their last name. An appropriate error message should be displayed, such as "Last name is required."
Priority Pass/Fail Criteria	High Pass: 1. 2. 3.	The user was not able to register the account without entering their last name. An appropriate error message should be displayed, such as "Last name is required." The user should remain on the registration page after attempting
Priority Pass/Fail Criteria	High Pass: 1. 2. 3.	The user was not able to register the account without entering their last name. An appropriate error message should be displayed, such as "Last name is required." The user should remain on the registration page after attempting to register without a last name.
Priority Pass/Fail Criteria	High Pass: 1. 2. 3. Fail:	The user was not able to register the account without entering their last name. An appropriate error message should be displayed, such as "Last name is required." The user should remain on the registration page after attempting to register without a last name.
Priority Pass/Fail Criteria	High Pass: 1. 2. 3. Fail: 1.	The user was not able to register the account without entering their last name. An appropriate error message should be displayed, such as "Last name is required." The user should remain on the registration page after attempting to register without a last name. The user receives an error message and remains on the
Priority Pass/Fail Criteria	High Pass: 1. 2. 3. Fail: 1.	The user was not able to register the account without entering their last name. An appropriate error message should be displayed, such as "Last name is required." The user should remain on the registration page after attempting to register without a last name. The user receives an error message and remains on the registration page.
Priority Pass/Fail Criteria	High Pass: 1. 2. 3. Fail: 1. 2.	The user was not able to register the account without entering their last name. An appropriate error message should be displayed, such as "Last name is required." The user should remain on the registration page after attempting to register without a last name. The user receives an error message and remains on the registration page. The user can register without entering a last name.
Priority Pass/Fail Criteria	High Pass: 1. 2. 3. Fail: 1. 2. 3.	The user was not able to register the account without entering their last name. An appropriate error message should be displayed, such as "Last name is required." The user should remain on the registration page after attempting to register without a last name. The user receives an error message and remains on the registration page. The user can register without entering a last name. The user is redirected to the login page or any other page
Priority Pass/Fail Criteria	High Pass: 1. 2. 3. Fail: 1. 2. 3.	The user was not able to register the account without entering their last name. An appropriate error message should be displayed, such as "Last name is required." The user should remain on the registration page after attempting to register without a last name. The user receives an error message and remains on the registration page. The user can register without entering a last name. The user is redirected to the login page or any other page indicating successful registration.

ID	TC4	Title	Validate User Registration Without the Username
Descriptio	on	Tests the user registration functionality to ensure that a new user cannot	
		successfully create an account without their username.	
Test Type	•	Unit	
Precondit	ion	1. 7	The DyslexAI application is accessible.
		2. 7	The user registration page is loaded.
Test Steps	5	1. I	Launch the DyslexAI application.
		2. N	Vavigate to the user registration page.
		3. E	Enter the following details
		4. F	First Name "Mahbub"
		5. I	Last Name "Khan"
		6. U	Jsername: ""
		7. E	Email: " <u>khanmahbub12@example.com"</u>
		8. F	assword "Mahbub12"
		9. (Confirm Password "Mahbub12"

Expected Results	1.	The user should not be able to register successfully without
		entering a Username.
	2.	An error message should be displayed indicating that the
		Username is required.
	3.	The user should remain on the registration page without being
		redirected to the login page.
Priority	High	
Pass/Fail Criteria	Pass:	
	1.	The user was not able to register the account without entering
		their username.
	2.	An appropriate error message should be displayed, such as
		"Username is required."
	3.	The error message should be clearly visible and not overlap with
		other UI elements.
	Fail:	
	1.	The user receives an error message and remains on the
		registration page.
	2.	The user can register without entering a username.
	3.	The error message is incorrect, unclear, or partially displayed.

ID	TC5	Title	Validate User Registration Without the email
Descripti	on	Tests the user registration functionality to ensure that a new user cannot	
		successfully create an account without their email.	
Test Type	e	Unit	
Precondi	tion	1.	The DyslexAI application is accessible.
		2.	The user registration page is loaded.
Test Step	s	1.	Launch the DyslexAI application.
		2.	Navigate to the user registration page.
		3.	Enter the following details
		4.	First Name "Mahbub"
		5.	Last Name "Khan"
		6.	Username: "Mahbubkh12"
		7.	Email: ""
		8.	Password "Mahbub12"
		9.	Confirm Password "Mahbub12"
Expected	Results	1.	The user should not be able to register successfully without entering an email.
		2.	An error message should be displayed indicating that the email is required.
		3.	The user should remain on the registration page without being redirected to the login page.
Priority		High	
Pass/Fail	Criteria	Pass:	

	1.	The user was not able to register the account without entering
		their email.
	2.	The user should be prevented from registering if the email field
		is empty.
	3.	An appropriate error message should be displayed, such as
		"Email is required."
	4.	The error message should be clearly visible and not overlap with
		other UI elements.
	Fail:	
	1.	The user receives an error message and remains on the
		registration page.
	2.	The user can register without entering an email.
	3.	The error message is incorrect, unclear, or partially displayed.
	4.	Only some of the missing fields result in error messages, while
		others do not.

ID	TC6	Title	Validate User Registration Without the password			
Description		Tests t	the user registration functionality to ensure that a new user cannot			
-		successfully create an account without their password.				
Test Type	;	Unit				
Precondit	ion	1.	The DyslexAI application is accessible.			
		2.	The user registration page is loaded.			
Test Steps	3	1.	Launch the DyslexAI application.			
		2.	Navigate to the user registration page.			
		3.	Enter the following details			
		4.	First Name "Mahbub"			
		5.	Last Name "Khan"			
		6.	Username: "Mahbubkh12"			
		7.	Email <u>khanmahbub12@example.com</u>			
		8.	Password ""			
		9.	Confirm Password "Mahbub12"			
E (1 D 1 (1	The year should not be able to register avecagefully without			
Expected Results		1.	antering a password			
		2	An arrow massage should be displayed indicating that the			
		۷.	nassword is required			
		3	The user should remain on the registration page without being			
		5.	redirected to the login page.			
Priority High						
Pass/Fail Criteria		Pass:				
		1.	The user was not able to register the account without entering			
			their password.			
		2.	The user should be prevented from registering if the password			
			field is empty.			
		3.	An appropriate error message should be displayed, such as			
			"Password is required.			

	4.	The focus should return to the password field, or the field should
		be highlighted to guide the user.
	Fail:	
	1.	The user receives an error message and remains on the
		registration page.
	2.	The user can register without entering a password.
	3.	The user is redirected to the login page or any other page
		indicating successful registered.
	4.	The error message is incorrect, unclear, or partially displayed.

ID	TC7	Title	Validate User Registration Without the Confirm Password				
Description		Tests the user registration functionality to ensure that a new user cannot successfully create an account without their Confirm Password.					
Test Type	;	Unit	Unit				
Precondit	ion	1. 7	The DyslexAI application is accessible.				
		2. The user registration page is loaded.					
Test Steps	8	1. 1	Launch the DyslexAI application.				
		2. 1	Navigate to the user registration page.				
		3. 1	Enter the following details				
		4. 1	First Name "Mahbub"				
		5. 1	Last Name "Khan"				
		6. 1	Jsername: "Mahbubkh12"				
		7. 1	Email: " <u>khanmahbub12@example.com"</u>				
		8. 1	Password "Mahbub12"				
		9. (Confirm Password ""				
Expected	Results	1. 7	The user should not be able to register successfully without				
			entering a confirm password.				
		2. 4	An error message should be displayed indicating that the				
			confirm password is required.				
		3. 7	The user should remain on the registration page without being				
		1	edirected to the login page.				
Priority		High					
Pass/Fail	Criteria	Pass:					
		1. 7	The user was not able to register the account without entering				
		t	heir confirm password.				
		2. 7	The user should be prevented from registering if the confirm				
		1	bassword field is em				
		3. 4	An appropriate error message should be displayed, such as				
		Fail:					
		1. 7	The user receives an error message and remains on the				
		1	egistration page.				
		2.	The user can register without entering a confirm password.				
		3. 7	The user is redirected to the login page or any other page				
		i	ndicating successful registered.				

4. No error message is displayed when the registration is attempted
without a confirm password.

ID	TC8	Title	Validate User Registration by leaving all fields blank			
Description		Tests the user registration functionality to ensure that a new user cannot successfully create an account without leaving all fields blank.				
Test Type	;	Unit				
Preconditi	ion	1.	The DyslexAI application is accessible.			
		2. The user registration page is loaded.				
Test Steps	3	1.	Launch the DyslexAI application.			
		2. Navigate to the user registration page.				
		3.	Enter the following details			
		4. 7	First Name ""			
		5.	Last Name ""			
		6. 7	Username: ""			
		/.	Email: ""			
		8. 0	Password Confirm Decement ""			
		9.	Commin Password			
Expected	Results	1.	The user should not be able to register successfully without			
1			entering any fields.			
		2.	An error message should be displayed indicating the required			
			fields.			
		3.	The user should remain on the registration page without being			
		TT: 1	redirected to the login page.			
Priority		High				
Pass/Fail Criteria		Pass:				
		1.	The user was not able to register the account without leaving all fields blank.			
		2.	The user should be prevented from registering if all fields are			
		3	Appropriate error messages should be displayed for each			
		5.	required field, such as "First name is required." "Last name is			
			required," "Username is required," "Email is required,"			
			"Password is required," and "Confirm password is required."			
		4.	The error messages should be clearly visible and not overlap			
			with other UI elements.			
		Fail:				
		1.	The user receives an error message and remains on the			
			registration page.			
		2.	The user can register without entering any details.			
		3.	The user is redirected to the login page or any other page			
			indicating successful registration.			
		4.	No error messages are displayed when the registration is attempted with all fields left blank.			

ID	TC9	Title	Validate User Registration Without a Complete Email				
			Address				
Description		Tests the cannot staddress.	Tests the user registration functionality to ensure that a new user cannot successfully create an account without a complete email address.				
Test Type		Unit					
Precondition		1. T	The DyslexAI application is accessible.				
		2. T	The user registration page is loaded.				
Test Steps		1. L 2. N 3. E 4. F 5. L 6. U 7. E 8. P 9. C 10. C	Launch the DyslexAI application. Navigate to the user registration page. Enter the following details First Name "Mahbub" Last Name "Khan" Jsername: "Mahbubkh12" Email: khanmahbub12 Password "Mahbub12" Confirm Password "Mahbub12" Click on Submit Button				
Expected Results		1. T	The user should not be able to register successfully without a				
		c	complete email.				
		2. T	The user should get a message indicating to enter a valid email address.				
		3. T	The user should remain on the registration page without being edirected to the login page.				
Priority		High					
Pass/Fail Cri	teria	Pass: 1. T c 2. T e 3. T	The user was not able to register the account without a complete email. The user receives an error message indicating to enter a valid email. The user remains on the registration page.				
		Fail: 1. T 2. T c 3. T e	The user was redirected to the login page. The user is redirected to the login page without entering a complete email. The user does not receive an error message to enter a valid email.				

ID	TC10	Title	Validate User Registration with a Numeric value in the First Name Text Field	
Description	•	Tests the user registration functionality to ensure that a new user cannot successfully create an account without a valid First Name.		

Test Type	Unit
Precondition	3. The DyslexAI application is accessible.
	4. The user registration page is loaded.
Test Steps	 Launch the DyslexAI application. Navigate to the user registration page. Enter the following details First Name "12345" Last Name "Khan" Username: "Mahbubkh12" Email: khanmahbub12@gmail.com Password "Mahbub12" Confirm Password "Mahbub12" Click on Submit Button
Expected Results	 The user should not be able to register successfully without a valid First Name. The user should remain on the registration page without being redirected to the login page.
Priority	High
Pass/Fail Criteria	 Pass: The user was not able to register the account without a valid First Name. The user receives an error message indicating the need to enter a valid first name. The user remains on the registration page without being redirected. Fail: The user can register using a numeric value in the First Name field. The user is redirected to the login page despite using an invalid first name. The user does not receive an error message indicating the need to enter a valid first name.

ID	TC11	Title	Validate User Registration with a Numeric value in the		
			Last Name Text Field		
Description		Tests the user registration functionality to ensure that a new user			
		cannot su	ccessfully create an account without a valid Last Name.		
Test Type		Unit			
Precondition		1. T	1. The DyslexAI application is accessible.		
		2. T	he user registration page is loaded.		
Test Steps		1. L	aunch the DyslexAI application.		

2.	Navigate to the user registration page.
3.	Enter the following details
4.	First Name "Mahbubur"
5.	Last Name "1234"
6.	Username: "Mahbubkh12"
7.	Email khanmahbub12@gmail.com
8.	Password "Mahbub12"
9.	Confirm Password "Mahbub12"
10	. Click on Submit Button
1.	The user should not be able to register successfully without a
	valid Last Name.
2.	The user should remain on the registration page without being
	redirected to the login page.
High	
Pass:	
1.	The user was not able to register the account without a valid Last Name.
2.	The user receives an error message indicating the need to
	enter a valid last name.
3.	The user remains on the registration page without being
	redirected.
Fail:	
1.	The user can register without a valid last name.
2.	The user is redirected to the login page despite not entering a
	valid last name
	2. 3. 4. 5. 6. 7. 8. 9. 10. 1. 2. High Pass: 1. 2. 3. Fail: 1. 2.

ID	TC12	Title	Validate User Registration with a Numeric value in the		
			Username Text Field.		
Description		Tests the	Tests the user registration functionality to ensure that a new user		
		cannot s	uccessfully create an account without a valid Username.		
Test Type		Unit			
Precondition		1. 7	The DyslexAI application is accessible.		
		2. 7	The user registration page is loaded.		
Test Steps		1. I	Launch the DyslexAI application.		
		2. N	Javigate to the user registration page.		
		3. E	Enter the following details		
		4. F	First Name "Mahbubur"		
		5. I	Last Name "Khan"		
		6. U	Jsername: "1234567"		
		7. E	Email: khanmahbub12@gmail.com		
		8. F	Password "Mahbub12"		
		9. 0	Confirm Password "Mahbub12"		
		10. 0	Click on Submit Button		

Expected Results	1.	The user should not be able to register successfully without a
		valid Username.
	2.	The user should remain on the registration page without being redirected to the login page.
Priority	High	~ * *
Pass/Fail Criteria	Pass:	
	1.	The user was not able to register the account without a valid
		Username.
	2.	The user receives an error message indicating the need to enter a valid username.
	3.	The user remains on the registration page without being redirected.
	Fail:	
	1.	The user was able to register without a valid Username.
	2.	The user is redirected to the login page despite not entering a
		valid username.
	3.	The user does not receive an error message indicating the
		need to enter a valid username.

ID	TC13	Title	Validate User Registration with a Numeric value in the
			Email Text Field.
Description		Tests the	user registration functionality to ensure that a new user
		cannot su	accessfully create an account without a valid Email Address.
Test Type		Unit	
Precondition		1. T	he DyslexAI application is accessible.
		2. T	he user registration page is loaded.
Test Steps		1. L	aunch the DyslexAI application.
		2. N	lavigate to the user registration page.
		3. E	Inter the following details
		4. F	'irst Name "Mahbubur"
		5. L	ast Name "Khan"
		6. U	Jsername: "Mahbub12"
		7. E	Email: 123456789
		8. P	assword "Mahbub12"
		9. C	Confirm Password "Mahbub12"
		10. C	lick on Submit Button
Expected Rea	sults	1. T	The user should not be able to register successfully without a
		v	alid email.
		2. T	The user should remain on the registration page without being edirected to the login page.
Priority		High	
Pass/Fail Cri	teria	Pass:	

1.	The user was not able to register the account without a valid
	email.
2.	The user receives a message indicating the need to enter a valid email.
3.	The user remains on the registration page without being redirected.
Fail:	
1.	The user can register without a valid email.
2.	The user is redirected to the login page despite not entering a valid email.
3.	The user does not receive an error message indicating the need to enter a valid email.

ID	TC14	Title	Validate User Registration Without the Submit Button		
Description		Tests the user registration functionality to ensure that a new user cannot			
-		success	fully create an account without clicking the submit button.		
Test Type		Unit			
Preconditi	ion	1.	The DyslexAI application is accessible.		
		2.	The user registration page is loaded.		
Test Steps	5	1.	Launch the DyslexAI application.		
		2.	Navigate to the user registration page.		
		3.	Enter the following details		
		4.	First Name "Mahbub"		
		5.	Last Name "Khan"		
		6.	Username: "Mahbubkh12"		
		7.	Email <u>khanmahbub12@example.com</u>		
		8.	Password "Mahbub12"		
		9.	Confirm Password "Mahbub12"		
		10.	Click on Submit Button		
Expected	Results	1.	The user should not be able to register successfully without		
		2	clicking the submit button.		
		Ζ.	redirected to the login page		
Priority		High	redirected to the togin page.		
Deco/Feil	Critorio	Decas			
F a85/1 all v	Cillena	газ <u>5</u> . 1	The user was not able to register the account without clicking		
		1.	the submit button		
		2	The system remains idle without providing any feedback unless		
		2.	the submit button is clicked ensuring no premature submission		
			occurs		
		3	The user remains on the registration page without being		
			redirected.		
		Fail:			
		1.	The user remains on the registration page.		
		2.	The user can register without clicking the submit button.		

3. The user is redirected to the login page or another page without
clicking the submit button.

ID	TC15	Title	Validate User Login
Description		Tests the user login functionality to ensure that a registered user can log	
		in succ	essfully.
Test Type		Unit	
Preconditi	on		 The DyslexAI application is accessible. A user with the email "khanmahbub12@example.com" and password "Mahbub12" is registered.
Test Steps	5	1. 2. 3. 4.	 Launch the DyslexAI application. Navigate to the user login page. Enter the following details Email "<u>khanmahbub12@example.com</u>" Password "Mahbub12" Press the Login button.
Expected	Results		 The user is successfully logged in. The user is redirected to the dashboard.
Priority		High	
Pass/Fail (Criteria	Pass: 1. 2. 3. 4. Fail:	The user is redirected to the dashboard. The user can log in using the correct email and password. The user is redirected to the dashboard upon successful login. A user session is initiated, and the user remains logged in across different pages until they log out.
		1. 2. 3.	The user receives an error message or is not redirected to the dashboard. The user receives an error message indicating incorrect credentials or other login issues despite entering the correct email and password. The user session is not properly initiated, or the user is logged out unexpectedly.

ID	TC16	Title	Validate User Login Without the email		
Description		Tests the user login functionality to ensure that a user can't log in			
		successfully without entering an email.			
Test Type		Unit			
Precondition	l	1. The	DyslexAI application is accessible.		
		2. The	login page is loaded		

Test Steps	1.	Launch the DyslexAI application.
	2.	Navigate to the user login page.
	3.	Enter the following details
		• Email ""
		Password "Mahbub12"
	4.	Press the Login button.
Expected Results	1.	The user should not be able to login successfully without
		entering an email.
	2.	An error message should be displayed indicating that the email is
		required.
	3.	The user should remain on the login page without being
		redirected to the homepage as a logged in user.
Priority	Mediu	m
Pass/Fail Criteria	Pass:	
	1.	The user was not able to login without entering an email.
	2.	An error message is displayed indicating that the email is required.
	3.	The user remains on the login page without being redirected to the homepage.
	4.	The focus should return to the email field, or the field should be
		highlighted to guide the user.
	Fail:	
	1.	The user was able to login in without entering an email.
	2.	No error message is displayed when attempting to log in without an email.
	3.	The user is redirected to the homepage or another page despite not entering an email.

ID	TC17	Title	Validate User Login Without the password
Description		Tests the	user login functionality to ensure that a user can't log in
		successf	ally without entering a password.
Test Type		Unit	
Precondition		1. T	he DyslexAI application is accessible.
		2. T	he login page is loaded
Test Steps		1. L	aunch the DyslexAI application.
		2. N	lavigate to the user login page.
		3. E	inter the following details

	• Email: " <u>khanmahbub12@example.com</u> "	
	• Password: ""	
	4. Press the Login button.	
Expected Results	1. The user should not be able to login successfully without	
	entering a password.	
	2. An error message should be displayed indicating the user to)
	enter in a password.	
	3. The user should remain on the login page without being	
	redirected to the homepage as a logged in user.	
Priority	Medium	
Pass/Fail Criteria	Pass:	
	1. The user is not able to login without entering a password.	
	2. An error message is displayed indicating that the password required	is
	3. The user remains on the login page without being redirected	1
	to the homepage.	
	4. The focus should return to the password field, or the field should be highlighted to guide the user.	
	Fail:	
	1. The user was able to login in without entering a password.	
	2. No error message is displayed when attempting to log in	
	without a password.	
	3. The user is redirected to the homepage or another page despite not entering a password.	

ID	TC18	Title	Validate User Login Without the email and password	
Description		Tests the user login functionality to ensure that a user can't log in		
		successfully without entering an email and password.		
Test Type		Unit		
Precondition		1. Т	he DyslexAI application is accessible.	
		2. Т	he login page is loaded	
Test Steps		1. I	aunch the DyslexAI application.	
		2. N	lavigate to the user login page.	
		3. E	Inter the following details	
		•	Email ""	
		•	Password ""	
		4. F	ress the Login button.	

Expected Results	1.	The user should not be able to login successfully without
		entering an eman and password.
	2.	An error message should be displayed indicating the user to enter
		in an email and password.
	4.	The user should remain on the login page without being
		redirected to the homepage as a logged in user.
Priority	Mediu	Im
Pass/Fail Criteria	Pass:	
	1.	The user was not able to login without entering an email and password.
	2.	The email and password fields should remain empty, and the focus should return to the first empty field or an error indication
	3.	The error message should be clearly visible and not overlap with other UI elements.
	Fail:	
	1.	The user was able to login in without entering an email and password.
	2.	The user is redirected to the homepage or any other page
		indicating a successful login.
	3.	Only one of the empty fields (email or password) results in an
		error message, while the other does not.

ID	TC19	Title		Validate User Login with unregistered email
Description		Tests the user login functionality to ensure that a user can't log in		
		successfully using an unregistered email.		
Test Type		Unit		
Precondition		1.	The l	DyslexAI application is accessible.
		2.	The l	login page is loaded
Test Steps		1.	Laun	ch the DyslexAI application.
		2.	Navi	gate to the user login page.
		3.	Enter	r the following details
			• E	Email " <u>Random@example.com</u> "
			• P	assword "Mahbub12"
		4.	Press	s the Login button.
Expected Re	esults	1.	The u	user should not be able to login successfully with an
			unreg	gistered email.
		2.	An e	rror message should be displayed indicating that we cannot
			find a	an account with that email address.

	3. The user should remain on the login page without being	
	redirected to the homepage as a logged in user.	
Priority	Medium	
Pass/Fail Criteria	Pass: The user was not able to login with an unregistered email address.	
	Fail: The user was able to login with an unregistered email address.	

ID	TC20	Title	Validate User Login with incorrect password		
Description		Tests t	he user login functionality to ensure that a user can't log in		
		successfully using an incorrect password.			
Test Type		Unit			
Precondition		1.	The DyslexAI application is accessible.		
		2.	The login page is loaded		
Test Steps		1.	Launch the DyslexAI application.		
		2.	Navigate to the user login page.		
		3.	Enter the following details		
			• Email " <u>Random@example.com</u> "		
			• Password "Mahbub@1"		
		4.	Press the Login button.		
Expected Results		1.	The user should not be able to login successfully with an		
			incorrect password.		
		2.	An error message should be displayed indicating that the		
			password is incorrect.		
		3.	The user should remain on the login page without being		
			redirected to the homepage as a logged in user.		
Priority		Medium			
Pass/Fail Crit	eria	Pass:			
		1.	The user was not able to login with an incorrect password.		
		2.	An error message is displayed indicating that the system cannot		
			find an account with that email address.		
		3.	The user remains on the login page without being redirected to		
			the homepage.		
		1.	The user was able to login with an incorrect password.		
		2.	The user can log in using an unregistered email address.		
		3.	No error message is displayed when attempting to log in with		
			an unregistered email address.		

	ID	TC21	Title	Validate Password Reset
--	----	------	-------	-------------------------

Description	Tests the password reset functionality to ensure that a user can reset their password		
Test Type	Unit		
Precondition	 The DyslexAI application is accessible. A user with the email <u>khanmahbub12@example.com</u> is registered. 		
Test Steps	 Launch the DyslexAI application. Navigate to the login page. Click on the "Forgot Password" link. Enter the email "khanmahbub12@example.com". Press the Submit button. Check the email for the reset link. Click the reset link. Enter the new password "Mahbub13". Confirm the new password. Press the "Reset Password" button. 		
Expected Results	 The user receives a password reset email. The password is successfully reset. The user is redirected to the login page with a success message. 		
Priority	Medium		
Pass/Fail Criteria	Pass: The user receives a reset email, successfully resets the password, and is redirected to the login page with a success message. Fail: The user does not receive the reset email, cannot reset the password, or is not redirected to the login page.		

ID	TC22	Title	Validate User Profile Management		
Descriptio)n	Tests the profile update functionality to ensure that a user can update			
1		their profile information.			
Test Type	;	Unit			
Precondit	ion	1	1. The DyslexAI application is accessible.		
		2	2. A user is logged in.		
Test Steps		1. I	Launch the DyslexAI application.		
		2. I	log in with valid credentials.		
		3. N	Vavigate to the profile page.		
		4. U	Jpdate the following details:		
		5. F	First Name: "Mahbubur"		
		6. I	_ast Name: "Rahman"		
		7. E	Email: " <u>khanmahbub12@example.com</u> "		
		8. F	Press the "Update" button.		
Expected	ected Results 1. The profile information is successfully updated.		The profile information is successfully updated.		
		2. A	A confirmation message is displayed.		
Priority		Medium			

Pass/Fail Criteria	Pass: The user receives a confirmation message, and the profile information is updated.
	not updated.

ID	TC23	Title	Validate User Profile Management by leaving First Name		
			text field empty		
Description		Verifies the profile update functionality to ensure that users cannot			
_		update their profile information by leaving the First Name text field			
		empty.			
Test Type		Unit			
Precondition		1. The DyslexAI application is accessible.			
		2. A user is logged in.			
Test Steps		1. Launch the DyslexAI application.			
		2. Log	in with valid credentials.		
		3. Nav	igate to the profile page.		
		4. Update the following details:			
		5. First	t Name: ""		
		6. Last	Name: "Rahman"		
		7. Ema	il: " <u>khanmahbub12@example.com</u> "		
		8. Pres	s the "Update" button.		
Expected Res	ults	1. The	profile information should not be successfully updated.		
		2. An e	error message will be displayed indicating to the user that		
		First	t Name is required.		
Priority		Medium			
Pass/Fail Criteria		Pass: The user doesn't receive a confirmation message, and the profile			
		information will not be updated.			
		Fail: The user receives a confirmation message, and the profile			
		information is updated.			

ID	TC24	Title	Validate User Profile Management by leaving Last Name		
			text field empty		
Description		Verifies the profile update functionality to ensure that users cannot			
		update their profile information by leaving the Last Name text field			
		empty.	empty.		
Test Type		Unit			
Precondition		1. The DyslexAI application is accessible.			
		2. A us	ser is logged in.		

Test Steps	1. Launch the DyslexAI application.	
	2. Log in with valid credentials.	
	3. Navigate to the profile page.	
	4. Update the following details:	
	5. First Name: " Mahbubur "	
	6. Last Name: ""	
	7. Email: " <u>khanmahbub12@example.com</u> "	
	8. Press the "Update" button.	
Expected Results	1. The profile information should not be successfully updated.	
	2. An error message will be displayed indicating to the user that	
	Last Name is required.	
Priority	Medium	
Pass/Fail Criteria	Pass: The user doesn't receive a confirmation message, and the profile	
	information will not be updated.	
	Fail: The user receives a confirmation message, and the profile	
	information is updated.	

ID	TC25	Title	Validate User Profile Management by leaving white	
			spaces in the First Name text field	
Description		Varifias	be profile undete functionality to ansure that users cannot	
Description		vermes	ne prome update functionality to ensure that users cannot	
		update their profile information by leaving only white spaces in the		
		First Name text field.		
Test Type		Unit		
Precondition		1. T	he DyslexAI application is accessible.	
		2. A	user is logged in.	
Test Steps		1. L	aunch the DyslexAI application.	
		2. L	og in with valid credentials.	
		3. N	avigate to the profile page.	
		4. U	pdate the following details:	
		5. F	irst Name: " "	
		6. L	ast Name: "Rahman"	
		7. E	mail: " <u>khanmahbub12@example.com</u> "	
		8. P	ress the "Update" button.	
Expected Results		1. T	he profile information should not be successfully updated.	
		2. A	n error message will be displayed indicating to the user that a	
		v	alid First Name is required.	
Priority	ity Medium			

Pass/Fail Criteria	Pass: The user doesn't receive a confirmation message, and the profile
	information will not be updated.
	Fail: The user receives a confirmation message, and the profile
	information is updated.

ID	TC26	Title	Validate User Profile Management by leaving white		
			spaces in the Last Name text field.		
Description		Verifies the profile update functionality to ensure that users cannot			
		update their profile information by leaving only white spaces in the			
		Last Name text field.			
Test Type		Unit			
Precondition		1. The	DyslexAI application is accessible.		
		2. A us	ser is logged in.		
Test Steps		1. Lau	nch the DyslexAI application.		
		2. Log	in with valid credentials.		
		3. Nav	3. Navigate to the profile page.		
		4. Update the following details:			
		5. First Name: " Mahbubur"			
		6. Last	Name: " "		
		7. Ema	il: " <u>khanmahbub12@example.com</u> "		
		8. Pres	s the "Update" button.		
Expected Results		1. The	profile information should not be successfully updated.		
		2. An e	error message will be displayed indicating to the user that a		
		valio	l Last Name is required.		
Priority		Medium			
Pass/Fail Crite	eria	Pass: The user doesn't receive a confirmation message, and the profile			
		information will not be updated.			
		Fail: The user receives a confirmation message, and the profile			
		information is updated.			

ID	TC27	Title	Validate User Profile Management by leaving Email text	
			field empty	
Description		Verifies the profile update functionality to ensure that users cannot		
update their profile information by leaving the Email text field em		profile information by leaving the Email text field empty.		
Test Type		Unit		
Precondition		1. The	DyslexAI application is accessible.	

	2. A user is logged in.		
Test Steps	1. Launch the DyslexAI application.		
	2. Log in with valid credentials.		
	3. Navigate to the profile page.		
	4. Update the following details:		
	5. First Name: " Mahbubur "		
	6. Last Name: "Rahman"		
	7. Email: "'		
	8. Press the "Update" button.		
Expected Results	3. The profile information should not be successfully updated.		
	4. An error message will be displayed indicating to the user that an		
	Email is required.		
Priority	Medium		
Pass/Fail Criteria	Pass: The user doesn't receive a confirmation message, and the profile		
	information will not be updated.		
	Fail: The user receives a confirmation message, and the profile		
	information is updated.		

ID	TC28	Title	Validate User Profile Management by entering numeric		
			values in the First Name text field.		
Description		Verifies the profile update functionality to ensure that users cannot			
		update their profile information using numeric values in the First N			
		text field.			
Test Type		Unit			
Precondition		1. 7	The DyslexAI application is accessible.		
		2. <i>A</i>	A user is logged in.		
Test Steps		1. I	1. Launch the DyslexAI application.		
		2. I	2. Log in with valid credentials.		
		3. 1	Navigate to the profile page.		
		4. U	4. Update the following details:		
		5. I	5. First Name: " 123 "		
		6. I	5. Last Name: "Rahman"		
		7. I	Email: "khanmahbub12@example.com"		
		8. Press the "Update" button.			
Expected Res	ults	1. The profile information should not be successfully upo			
			2. An error message will be displayed indicating to the user to		
		enter in a valid First Name.			
Priority		Medium			

Pass/Fail Criteria	Pass: The user doesn't receive a confirmation message, and the profile
	information will not be updated.
	Fail: The user receives a confirmation message, and the profile
	information is updated.

ID	TC29	Title	Validate User Profile Management by entering numeric		
			values in the Last Name text field.		
Description		Verifies the profile update functionality to ensure that users cannot			
		update their profile information using numeric values in the First Name			
		text field.			
Test Type		Unit			
Precondition		1. The	1. The DyslexAI application is accessible.		
		2. A u	ser is logged in.		
Test Steps		1. Lau	nch the DyslexAI application.		
		2. Log	in with valid credentials.		
		3. Nav	3. Navigate to the profile page.		
		4. Update the following details:			
		5. Firs	5. First Name: " Mahbubur "		
		6. Las	t Name: "233"		
		7. Ema	ail: " <u>khanmahbub12@example.com</u> "		
		8. Pres	ss the "Update" button.		
Expected Results		1. The	profile information should not be successfully updated.		
		2. An	error message will be displayed indicating to the user to		
		ente	er in a valid Last Name.		
Priority		Medium			
Pass/Fail Criteria Pass: The user doesn't receive a confirm		ser doesn't receive a confirmation message, and the profile			
		information will not be updated.			
		Fail: The user receives a confirmation message, and the profile			
		information is updated.			

ID	TC30	Title	Validate User Profile Management by entering only	
			numeric values in the Email text field.	
Description		Verifies the profile update functionality to ensure that users cannot		
		update their profile information using numeric values in the Email text		
		field.		
Test Type		Unit		
Precondition		1. The	DyslexAI application is accessible.	
		2. A us	er is logged in.	

Test Steps	3. Launch the DyslexAI application.		
	4. Log in with valid credentials.		
	5. Navigate to the profile page.		
	6. Update the following details:		
	7. First Name: " Mahbubur "		
	8. Last Name: "Khan"		
	9. Email: "12345678"		
	10. Press the "Update" button.		
Expected Results	1. The profile information should not be successfully updated.		
	2. An error message will be displayed indicating to the user to		
	enter in an Email.		
Priority	Medium		
Pass/Fail Criteria	Pass: The user doesn't receive a confirmation message, and the profile		
	information will not be updated.		
	Fail: The user receives a confirmation message, and the profile		
	information is updated.		

ID	TC31	Title	Validate User Profile Management by entering only		
			punctuations in the First Name text field.		
Description		Verifies the profile update functionality to ensure that users cannot			
		update their profile information using only punctuation in the First			
		Name text field.			
Test Type		Unit			
Precondition		1. The	DyslexAI application is accessible.		
		2. A us	ser is logged in.		
Test Steps		1. Lau	nch the DyslexAI application.		
		2. Log	in with valid credentials.		
		3. Navigate to the profile page.			
		4. Upd	4. Update the following details:		
		5. First	t Name: " () "		
		6. Last	Name: "Khan"		
		7. Email: " <u>khanmahbub12@example.com</u> "			
		8. Press the "Update" button.			
Expected Res	ults	1. The	profile information should not be successfully updated.		
		2. An	error message will be displayed indicating to the user to		
		enter in a valid First Name.			
Priority		Medium			
Pass/Fail Crit	eria	Pass: The u	ser doesn't receive a confirmation message, and the profile		
		information will not be updated.			

Fail: The user receives a confirmation message, and the profile
information is updated.

ID	TC32	Title	Validate User Profile Management by entering only		
			punctuations in the Last Name text field.		
Description		Verifies the profile update functionality to ensure that users cannot			
_		update their profile information using only punctuation in the Last			
		Name text field.			
Test Type		Unit			
Precondition		1. The DyslexAI application is accessible.			
		2. A u	ser is logged in.		
Test Steps		1. Lau	nch the DyslexAI application.		
		2. Log	in with valid credentials.		
		3. Nav	3. Navigate to the profile page.		
		4. Update the following details:			
		5. First Name: " Mahbubur "			
		6. Las	t Name: "@@@@@"		
		7. Ema	ail: " <u>khanmahbub12@example.com</u> "		
		8. Pres	ss the "Update" button.		
Expected Res	ults	1. The	profile information should not be successfully updated.		
		2. An	error message will be displayed indicating to the user to		
		ente	er in a valid First Name.		
Priority		Medium			
Pass/Fail Crite	Pass/Fail Criteria Pass: The user doesn't receive a confirmation message, and		ser doesn't receive a confirmation message, and the profile		
		information will not be updated.			
		Fail: The user receives a confirmation message, and the profile			
		information is updated.			
1					

ID	TC33	Title	Validate User Profile Management by entering only	
			punctuations in the Email text field.	
Description		Verifies the profile update functionality to ensure that users cannot		
		update their	profile information using only punctuation in the Email	
		text field.		
Test Type		Unit		
Precondition		1. The	DyslexAI application is accessible.	
		2. A us	er is logged in.	
Test Steps		1. Lau	hch the DyslexAI application.	
		2. Log	in with valid credentials.	

	3. Navigate to the profile page.		
	4. Update the following details:		
	First Name: " Mahbubur "		
	5. Last Name: "Khan"		
	6. Email: "@@@@@@@@"		
	7. Press the "Update" button.		
Expected Results	1. The profile information should not be successfully updated.		
	2. An error message will be displayed indicating to the user to		
	enter in a valid email.		
Priority	Medium		
Pass/Fail Criteria	Pass: The user doesn't receive a confirmation message, and the profile		
	information will not be updated.		
	Fail: The user receives a confirmation message, and the profile		
	information is updated.		

ID	TC34	Title	Validate User Profile Management by entering the		
			incomplete email address in the Email text field.		
Description		Verifies the profile update functionality to ensure that users cannot			
		update their profile information using incomplete email address.			
Test Type		Unit			
Precondition		1. The	1. The DyslexAI application is accessible.		
		2. A u	ser is logged in.		
Test Steps		1. Lau	nch the DyslexAI application.		
		2. Log	in with valid credentials.		
		3. Nav	vigate to the profile page.		
		4. Upo	4. Update the following details:		
		5. First Name: " Mahbubur "			
		6. Las	6. Last Name: "Khan"		
		7. Em	ail:"khanmahbub12"		
		8. Pres	ss the "Update" button.		
Expected Res	ults	3. The	profile information should not be successfully updated.		
		4. An	error message will be displayed indicating to the user to		
		ente	er in a valid email.		
Priority		Medium			
Pass/Fail Criteria Pass: The user doesn't receive a confirmation message, as		ser doesn't receive a confirmation message, and the profile			
information will not be updated.		n will not be updated.			
Fail: Th		Fail: The u	ser receives a confirmation message, and the profile		
information is updated.		information	n is updated.		

ID	TC35	Title	Validate E-Book Search and Import
Description Tests the functionality of searching for e-books from an onlin and importing e-books into the application. The system must various e-book formats and ensure they are accessible within application		functionality of searching for e-books from an online library orting e-books into the application. The system must support -book formats and ensure they are accessible within the on.	
Test Type		Integratio	Dn
Preconditi	on	 The DyslexAI application is accessible. A user is logged in. The user has a valid search query or e book file to import 	
Expected	Results	 The application displays a formatted list of all search results The selected e-book is successfully imported into the application. The imported e-books PDF are accessible within the application. Error messages are displayed for any failed searches or unsupported file formats. The user is provided with guidance on how to find supported formats 	
Priority		Medium	
Pass/Fail (Criteria	 Pass: The application displays the search results, allows the user to import selected e-books, and supports PDF and ePub formats. Error messages and user guidance are displayed appropriately. Fail: The application fails to display search results, does not allow import of selected e-books, does not support PDF and ePub formats does not display error messages and user guidance as expected. 	

ID	TC36	Title	Validate Deaf Mode Functionality	
Description		Tests the Deaf Mode functionality that converts selected text into images to aid deaf users in understanding content through visual representations. Utilizes DALL-E to generate relevant images based on the text, provided line by line to generate images using lines of text as prompts.		
Test Type	;	Integratio	Dn	
Precondition		1 2 3 4	 The DyslexAI application is accessible. A user is logged in. The user has selected text for conversion. The DALL-E API is functional and accessible. 	
Test Steps		1. L 2. L 3. N 4. A 5. C 6. N te 7. A	aunch the DyslexAI application. og in with valid credentials. avigate to the reading page with text content. ctivate Deaf Mode by clicking the "Generate Images" button. confirm the conversion process. fonitor the generation of images corresponding to the selected ext. ttempt to convert text into images again.	

Expected Results	1. The application converts the selected text into images using DALL-E.
	2. The generated images are displayed, representing the current text.
	3. If there is an API request error or a server error, an appropriate error message is displayed.
	4. Detailed error messages are shown if an image cannot be generated.
Priority	High
Pass/Fail Criteria	Pass: The application successfully converts the current text into images and displays them. Error messages are displayed appropriately for any API request errors, server errors, or image generation failures. The speed of image generation matches the selected input. Fail: The application fails to convert the text into images, does not display generated images, or does not show error messages appropriately for API request errors, server errors, or image generation failures.

ID	TC37	Title Validate Blind Mode Functionality
Descriptio	on	Tests the Blind Mode functionality that provides text-to-speech functionality to aid blind users. Users will have all text, including buttons and titles, read to them. Upon load, an explanation will be read to the user and a slight ticking sound that speeds up upon nearing an object/interface is used to guide users.
Test Type		Integration
Preconditi	on	 The DyslexAI application is accessible. A user is logged in. The user has a mouse or touchpad for interaction.
Test Steps		 Launch the DyslexAI application. Log in with valid credentials. Do not disable "Blind mode" on website load. Navigate to any page with text content or any object on screen. Move the mouse over different text elements, including buttons and titles. Listen to the audio output as the mouse hovers over text elements. Move the mouse towards an interactive object and listen for the ticking sound. Attempt to interact with the text elements again.
Expected	Results	 The application reads all text, including buttons and titles, to the user. Upon loading Blind Mode, an explanation of the functionality is read to the user.

	 A ticking sound is heard, speeding up as the mouse nears an interactive object. If a task cannot be completed or nothing is interacted with, an audio-based error message is displayed.
Priority	High
Pass/Fail Criteria	Pass: The application successfully reads all text to the user, provides an initial explanation, and the ticking sound guides the user as expected. Audio-based error messages are displayed appropriately when a task cannot be completed, or nothing is interacted with. Fail: The application fails to read text, does not provide an initial explanation, the ticking sound does not function as expected, or audio- based error messages are not displayed appropriately.

ID	TC38	Title	Validate Customizable Reading Interface
Descriptio	n	Tests the to adjust changing paragrap	Customizable Reading Interface functionality that allows users reading settings to improve readability. Options include font type, font size, background color, line spacing, and h spacing.
Test Type		Integration	on
Preconditi	Precondition1. The DyslexAI application is accessible. 2. A user is logged in.		
Test Steps	3	1. L 2. L 3. N 4. A 5. F 6. F 7. B 8. L 9. P 10. S 11. C 12. A	aunch the DyslexAI application. og in with valid credentials. lavigate to the reading settings page. djust the following reading settings: ont type (Arial) ont size (18pt) ackground color (black) ine spacing (1.5) aragraph spacing (10px) ave the changes. Open an e-book to verify the applied settings. Attempt to apply a new set of reading settings again.
Expected	Results	 The reading settings are applied, and the customized U are displayed based on user preferences. If customization fails, the settings are restored to default. 	
Priority		Medium	
Pass/Fail (Criteria	Pass: The displays customiz Fail: The the custo customiz	e application successfully applies the reading settings and the customized UI and UX based on user preferences. If ation fails, the settings are restored to default. application fails to apply the reading settings, does not display mized UI and UX, or does not restore the default settings when ation fails.

ID	TC39	Title	Validate Book Recommendations		
Descriptio	on	Tests the Book Recommendations functionality that uses AI to analyze			
		user read	user reading history and recommend books. The system provides		
Test Trues		Interneti	personalized recommendations for enhanced user experience.		
Test Type		Integratio			
Preconditi	on	1	. The DyslexAI application is accessible.		
		2	. A user is logged in.		
Test Steps	5	1. L	aunch the DyslexAI application.		
		2. L	og in with valid credentials.		
		3. N	lavigate to the book recommendations page.		
		4. V	iew the list of recommended books based on user reading		
		h	istory and preferences.		
		5. S	imulate an AI recommendation failure by disabling internet		
			onnection.		
		0. K	erresh the book recommendations page.		
Expected Results		1. T	The application displays a formatted list of recommended books		
		b	ased on the user's reading history and preferences.		
		2. If	AI recommendations fail, default recommendations are		
		d	isplayed.		
		3. U	Isers are notified of any issues with the recommendation		
		S	ystem.		
Priority		Low			
Pass/Fail	Criteria	Pass: The	e application successfully displays personalized book		
		recommendations based on user reading history and preferences. If AI			
		recommendations fail, default recommendations are displayed, and the			
		user is notified of the issue.			
		Fail: The application fails to display personalized book			
		recommendations, does not show default recommendations when AI			
		recommendations fail, or does not notify the user of issues with the			
		recomme	endation system.		

ID	TC40	Title	Validate LLM and AI Integration	
Description Tests the LLM and AI Integration functionality		LLM and AI Integration functionality that uses OpenAI's API		
		to manag	e all AI-related requests, including querying DALL-E for	
images, GPT for finding and suggesting e-books, and GPT-4 for			GPT for finding and suggesting e-books, and GPT-4 for	
searching the web to find documents.				
Test Type Integration		on la		
Precondition 1. Th		1. T	he DyslexAI application is accessible.	
		2. A user is logged in.		
3. OpenAI API is functional and accessible.			penAI API is functional and accessible.	

Test Steps1. Launch the DyslexAI application.				
2. Log in with valid credentials.				
3. Navigate to a page where text can be selected.				
4. Select a block of text for image generation.				
5. Use the OpenAI API (DALL-E) to generate images the selected text.	based on			
6. Verify the generated images are displayed correctly.				
7. Enter a search query for e-books using GPT.				
8. Verify the list of suggested e-books based on user readers	ading			
history.	U			
9. (Optional) Simulate an API error by disabling internet	et			
connection.				
10. Attempt to use the AI functionalities again.				
Expected Results 1. The application successfully generates images based on	user-			
selected text.				
2. A list of suggested e-books is displayed based on user re	2. A list of suggested e-books is displayed based on user reading			
history.	history.			
3. If AI request errors occur, a dynamic modal displays the	3. If AI request errors occur, a dynamic modal displays the error			
messages.				
Priority Low				
Pass/Fail Criteria Pass: The application successfully integrates with OpenAI's AP	I,			
generates images, and suggests e-books. Error messages are dis	generates images, and suggests e-books. Error messages are displayed			
appropriately for AI request errors.	-			
Fail: The application fails to generate images or suggest e-books	s or does			
not display error messages appropriately for AI request errors.				

5.2 Non-Functional Test Cases:

ID	TC41	Title	Validate Performance	
Description		Tests the performance of the application to ensure pages load and respond to user actions within 2 seconds. The system should handle simultaneous user interactions efficiently without significant delays.		
Test Type		Non-Fund	ctional	
Precondition		1. 2.	The DyslexAI application is accessible. Multiple users are logged in and interacting with the application.	
Test Steps		1. 2. 3. 4. 5. 6.	Launch the DyslexAI application. Log in with valid credentials. Navigate through various pages of the application. Measure the page load times and response times for user actions. Simulate multiple users interacting with the application simultaneously. Monitor the performance under load.	
Expected	Results	1. 2.	Pages load and respond to user actions within 2 seconds. The application handles simultaneous user interactions efficiently without significant delays.	

Priority	Low
Pass/Fail Criteria	Pass: The application meets the performance requirements with pages
	loading and responding within 2 seconds and handles simultaneous user
	interactions efficiently.
	Fail: The application fails to meet the performance requirements with
	pages loading or responding in more than 2 seconds or does not handle
	simultaneous user interactions efficiently.

ID	TC42	Title	Validate Database Information Updating	
Description		Tests the system's ability to update database information (user profiles, preferences, reading history, account deletion or creation, and e-book data) within 2 seconds of the user's request.		
Test Type		Non-Fun	ctional	
Preconditi	on	1 2	The DyslexAI application is accessible.A user is logged in and interacting with the application.	
Test Steps		1 2 3 4 5 6	 Launch the DyslexAI application. Log in with valid credentials. Navigate to the profile management page. Update the user profile information. Save the changes and measure the time taken for the update to be processed. Verify that the changes are reflected immediately. 	
Expected	Results	 The database updates are processed within 2 seconds of the user's request. Changes are reflected immediately to support data consistency. 		
Priority		Medium		
Pass/Fail Criteria		Pass: The seconds a Fail: The not reflect	e system processes updates to the database information within 2 and reflects the changes immediately. e system takes more than 2 seconds to process updates or does ct the changes immediately.	

ID	TC43	Title	Validate Response Time for Non-AI API Calls	
Descriptio	n	Tests the system's response time for non-AI API calls to ensure a		
		maximun	n of 3 seconds for fetching user data, book information, and	
		preference settings.		
Test Type		Non-Functional		
Preconditi	ion	1.	The DyslexAI application is accessible.	
		2.	A user is logged in and interacting with the application.	
Test Steps		1.	Launch the DyslexAI application.	
		2.	Log in with valid credentials.	
		3.	Navigate to a page that requires fetching user data, book	
			information, or preference settings.	
		4.	Measure the response time for the API calls.	
Expected Results		The syste	m responds to non-AI API calls within a maximum of 3	
		seconds.		

Priority	Medium
Pass/Fail Criteria	Pass: The system responds to non-AI API calls within 3 seconds. Fail: The system takes more than 3 seconds to respond to non-AI API calls.

ID	TC44	Title	Validate Server Error Handling	
Description		Tests the system's ability to handle server errors and request errors.		
Test Type		Non-Functional		
Preconditi	on	1.	The DyslexAI application is accessible.	
		2.	A user is logged in and interacting with the application.	
Test Steps		1.	Launch the DyslexAI application.	
		2.	Log in with valid credentials.	
		3.	Simulate a server error by disabling the internet connection	
			or causing a server-side error.	
		4.	Attempt to perform an action that requires server interaction.	
Expected Results		1.	The system displays appropriate error messages for server	
			errors and request errors.	
		2.	The system handles errors gracefully and recovers when the	
			connection is restored.	
Priority		Medium		
Pass/Fail (Criteria	Pass: The	system displays appropriate error messages and handles errors	
		gracefully	У.	
		Fail: The	system does not display error messages or handle errors	
		gracefull	y.	

ID	TC45	Title	Validate Uptime and Availability		
Description		Tests the server's uptime and availability to ensure it is supported and			
		live 24/7.			
Test Type		Non-Fun	ctional		
Preconditi	ion	1.	1. The DyslexAI application is accessible.		
		2. Server monitoring tools are set up.			
Test Steps		1. Monitor the server uptime and availability over time.			
		2. Record any downtime or unavailability incidents.			
Expected Results		The serve	er is supported and lives 24/7 with minimal downtime.		
Priority		High			
Pass/Fail Criteria Pass: The serv		Pass: The	e server is live 24/7 with no significant downtime.		
		Fail: The server experiences significant downtime or is frequently			
unavailable.		ole.			

ID	TC46	Title	Validate Portability
Description		Tests the application's ability to run on a web server hosted within a Linux environment.	
Test Type Non-		Non-Fun	ctional

Precondition	The DyslexAI application is deployed on a Linux-based web server.
Test Steps	 Deploy the DyslexAI application on a Linux-based web server. Access the application from various devices and browsers.
	 Verify that the application functions correctly on all tested environments.
Expected Results	The application runs smoothly on a web server hosted within a Linux environment.
Priority	Medium
Pass/Fail Criteria	Pass: The application runs smoothly on a Linux-based web server across all tested devices and browsers.Fail: The application fails to run or experiences issues on a Linux-based web server.

ID	TC47	Title	Validate Scalability	
Description		Tests the system's scalability to accommodate more users and the		
		volume o	f data used.	
Test Type		Non-Fun	ctional	
Preconditi	ion	1.	The DyslexAI application is accessible.	
		2.	Multiple user accounts and data sets are available.	
Test Steps		1.	Simulate an increasing number of users interacting with the application.	
		2.	Monitor the system's performance and responsiveness.	
		3.	Verify that the application handles the increased load	
			without significant performance degradation.	
Expected Results		The appli	cation scales effectively to accommodate more users and data	
		volume.		
Priority		Medium		
Pass/Fail Criteria		Pass: The	e application scales effectively, handling increased users and	
		data volume without performance degradation.		
		Fail: The application experiences performance issues or fails to handle		
		the increased load.		

5.3 Test Scripts and Mock Data:

Test Script TC1	<pre>import { signUpLogic } from '/Signup';</pre>
	<pre>import { getAuth, createUserWithEmailAndPassword, updateProfile } from</pre>
	'firebase/auth';
	<pre>import { getDatabase, set, ref } from 'firebase/database';</pre>
	jest.mock('firebase/auth');
	jest.mock('firebase/database');
	<pre>describe('signUpLogic', () => {</pre>
	it('should sign up a user and update profile', async () => {
	const mockUser = { uid: '123' };

<pre>createUserWithEmailAndPassword.mockResolvedValue({ user: mockUser }); updateProfile.mockResolvedValue();</pre>
set.mockResolvedValue();
await signUpLogic('test@example.com', 'testuser', 'password', 'First', 'Last');
expect (create User With Email And Password). to Have Been Called With (expect. any this is the second se
ng(), 'test@example.com', 'password');
expect(updateProfile).toHaveBeenCalledWith(mockUser, {
'testuser' });
expect(set).toHaveBeenCalledWith(ref(expect.anything(), 'users/123'), {
firstname: 'First',
lastname: 'Last',
});
});
});

Test Script TC2	<pre>import { loginLogic } from '/Login';</pre>
	<pre>import { getAuth, signInWithEmailAndPassword } from 'firebase/auth';</pre>
	jest.mock('firebase/auth');
	<pre>describe('loginLogic', () => {</pre>
	<pre>it('should log in a user with email and password', async () => { const mockUser = { uid: '123' };</pre>
	signInWithEmailAndPassword.mockResolvedValue({ user: mockUser });
	await loginLogic('test@example.com', 'password');
	<pre>expect(signInWithEmailAndPassword).toHaveBeenCalledWith(expect.anything() , 'test@example.com', 'password');</pre>
	<pre>}); });</pre>

Test Script TC3	<pre>import { forgotPasswordLogic } from '/Forgotpassword';</pre>						
	<pre>import { getAuth, sendPasswordResetEmail } from 'firebase/auth';</pre>						
	jest.mock('firebase/auth');						
	<pre>describe('forgotPasswordLogic', () => {</pre>						
	it('should send a password reset email', async () $=>$ {						
	sendPasswordResetEmail.mockResolvedValue();						

await forgotPasswordLogic('test@example.com');
expect(sendPasswordResetEmail).toHaveBeenCalledWith(expect.anything(), 'test@example.com');
<pre>}); });</pre>

Test Script TC4	Test Script N/A, function not ready for testing.

Test Script TC5	TC5 import { eBookSearch } from '/EBookImport';						
	import axios from 'axios';						
	<pre>import { getLastBookId } from '/EBookImport';</pre>						
	jest.mock('axios');						
	jest.mock('/EBookImport', () => ({						
	jest.requireActual('/EBookImport'),						
	getLastBookId: jest.fn(),						
	}));						
	<pre>describe('eBookSearch', () => {</pre>						
	it('should search for e-books and return a list of books', async () $=>$ {						
	const mockResponse = {						
	data: {						
	items: [
	{						
	link: 'http://example.com/book1.pdf',						
	title: 'Book 1',						
	snippet: 'Description 1',						
	<pre>pagemap: { cse_thumbnail: [{ src: 'http://example.com/image1.jpg' }] },</pre>						
	},						
	{						
	link: 'http://example.com/book2.pdf',						
	title: 'Book 2',						
	snippet: 'Description 2',						
	<pre>pagemap: { cse_thumbnail: [{ src: 'http://example.com/image2.jpg' }] },</pre>						
	},						
],						
	queries: {						
	nextPage: [{ startIndex: 11 }],						
	},						

axios.get.mockResolvedValue(mockResponse); getLastBookId.mockResolvedValue(0); const searchQuery = 'Harry+Potter+pdf'; const books = await eBookSearch(searchQuery); expect(axios.get).toHaveBeenCalledWith(`https://www.googleapis.com/customsearch/v1?key=AIzaSyDLM-FFzf1ZKg1_SI1yW98xv3qqDCbhwPE&cx=b005aa8a990ec4e7a&q=\${searchQuery &filetype=pdf&start=1` id: 'book1', title: 'Book 1', description: 'Description 1', imageUrl: 'http://example.com/image1.jpg', pdfUrl: 'http://example.com/book1.pdf', id: 'book2'. title: 'Book 2', description: 'Description 2', imageUrl: 'http://example.com/image2.jpg', pdfUrl: 'http://example.com/book2.pdf', it('should handle errors during the search', async () => { axios.get.mockRejectedValue(new Error('API error')); const searchQuery = 'Harry+Potter+pdf'; await expect(eBookSearch(searchQuery)).rejects.toThrow('Error searching PDFs:');

import OpenAI from 'openai':
jest.mock('openai');
<pre>describe('generateImage', () => { it('should generate an image URL from a prompt', async () => { const mockResponse = { data: [{ url: 'http://example.com/image.png' }], }; }</pre>
OpenAI.prototype.images.generate.mockResolvedvalue(mockResponse);
<pre>const prompt = 'A beautiful sunset over the mountains'; const imageUrl = await generateImage(prompt);</pre>
<pre>expect(OpenAI.prototype.images.generate).toHaveBeenCalledWith({ model: 'dall-e-3', prompt: prompt, n: 1,</pre>
size: '1024x1024',
});
<pre>expect(imageUrl).toBe('http://example.com/image.png'); });</pre>
<pre>it('should throw an error if image generation fails', async () => { OpenAI.prototype.images.generate.mockRejectedValue(new Error('API error'));</pre>
const prompt = 'A beautiful sunset over the mountains';
<pre>await expect(generateImage(prompt)).rejects.toThrow('Error generating image:'); }); });</pre>

Test Script TC7	Test Script N/A, function not ready for testing.			

Test Script TC8	import React from 'react';					
	<pre>import { render, screen, fireEvent } from '@testing-library/react';</pre>					
import Settings from '/Settings';						
	<pre>import { getDatabase, ref, set, get } from 'firebase/database';</pre>					
	<pre>import { getAuth, onAuthStateChanged } from 'firebase/auth';</pre>					
	jest.mock('firebase/database');					

```
jest.mock('firebase/auth');
```

```
describe('Settings', () \Rightarrow
  getAuth.mockReturnValue({
   currentUser: { uid: '123' },
  onAuthStateChanged.mockImplementation((auth, callback) => {
   callback({ uid: '123' });
  get.mockResolvedValue({
   exists: () => true,
    theme: 'light',
 it('renders settings page with current preferences', async () => {
  render(<Settings />);
  expect(await screen.findByLabelText('Font Size')).toHaveValue('16px');
  expect(screen.getByLabelText('Theme')).toHaveValue('light');
 it('updates font size preference', async () => {
  render(<Settings />);
  const fontSizeInput = await screen.findByLabelText('Font Size');
  fireEvent.change(fontSizeInput, { target: { value: '18px' } });
  expect(fontSizeInput).toHaveValue('18px');
 it('updates theme preference', async () => {
  render(<Settings />);
  const themeSelect = await screen.findByLabelText('Theme');
  fireEvent.change(themeSelect, { target: { value: 'dark' } });
  expect(themeSelect).toHaveValue('dark');
 it('saves preferences to the database', async () => {
  render(<Settings />);
  const saveButton = await screen.findByText('Save Preferences');
  fireEvent.click(saveButton);
  expect(set).toHaveBeenCalledWith(ref(getDatabase(), 'users/123/preferences'), {
   fontSize: '16px',
```

theme: 'light', });
<pre>});</pre>
});

Test Script TC9	Test Script N/A, function not ready for testing.

Test Script TC10	<pre>import { generateImage } from '/OpenAI';</pre>						
	import OpenAI from 'openai';						
	jest.mock('openai');						
	describe('generateImage', () => {						
	it('should generate an image URL from a prompt', async () => {						
	const mockResponse = {						
	data: [{ url: 'http://example.com/image.png' }],						
	};						
	OpenAI.prototype.images.generate.mockResolvedvalue(mockResponse);						
	const prompt $- A$ beautiful subset over the mountains'						
	const prompt = A beautiful subset over the mountains,						
	const mage off a war generaternage (prompt),						
	expect(OpenAI.prototype.images.generate).toHaveBeenCalledWith({						
	model: 'dall-e-3',						
	prompt: prompt,						
	n: 1,						
	size: '1024x1024',						
	});						
	expect(imageUrl).toBe('http://example.com/image.png');						
	});						
	$it/ahould throw an arrow if image concretion fails' as 0 \rightarrow 1$						
	(1) (should throw an error in image generation rans, async () => {						
	open/ii.prototype.inages.generate.inoexitejeeted value(new Error(/ii renor)),						
	const prompt = 'A beautiful sunset over the mountains':						
	await expect(generateImage(prompt)).rejects.toThrow('Error generating image:');						
	});						
	});						

5.4 Traceability Matrix:

Please double click on the Excel Sheet and scroll to see the rest of the Matrix.



6. Test Schedule:

Sun	Mon	Tue	Wed	Thu	Fri	Sat
7/20/24	7/15/24	7/9/24	7/10/24	7/11/24	7/12/24	7/13/24
Write test script	Write test script	Write test script				

Start testing: User Preferences Manageme nt		Start testing: User Registrati on (TC1)	Start testing: Customiz able Reading Interface (TC8)	Start testing: E-Book Search and Import (TC5)	Start testing: Deaf Mode (TC6)	Start testing: Large Language Model (LLM) and AI Integratio n (TC10)
	Start Testing Database Informati on Updating (TC12)	Start testing: User Login (TC2)	Start testing: Blind Mode (TC7)	Start testing: Book Recomm endations (TC9)	Start Testing on Uptime and availabili ty	Response Time for Non-AI API Calls (TC13)
Start Testing on Portability (TC16)		Start testing: Password Reset (TC3)		Start Testing Server Error Handling (TC14)		Uptime and Availabil ity (TC15)
Start Testing on Scalability (TC17)		Start testing: Profile Managem ent (TC4)	Start Testing Performa nce (TC11)			
Final Review and Documenta tion	Final systems test (All test cases)					

7. Test Environment:

Testing of DyslexAI web application. To successfully test the environment, we will employ dedicated testing computers. While there are no specific requirements regarding the type of operating system used by these computers, they must be capable of running Visual Studio Code for testing purposes. We will leverage Visual Studio Code's Test Explorer feature to conduct unit testing on backend functions, including sign-in and sign-out processes. Additionally, for the test environment, we will utilize Postman for both suite and unit testing of our implemented backend and frontend APIs.

8. Test Entry and Exist Criteria:

The entry criteria for the testing phase of DyslexAI's occur after the completion of prototype 2 of the DyslexAI's web application. While the existing criteria of the testing phase are dependent on the successful execution of each of the test cases outlined in our test plan with their expected outcomes and yielding a passing result.

9. Test Pass and Fails Criteria:

In the testing system for DyslexAI, there are two potential outcomes following the execution of a test. These outcomes fall into two distinct categories of pass and fail, for a test to be labeled as a pass in the DyslexAI web applications, its first must meet the criteria of having each of their results after execution match with the expected results specified in its respective test case. If the test doesn't not meet this criterion by producing an unexpected result, it is deemed as a failed test. If a test returns as a failed test, it will mark for a review. The review process includes an investigation into why the test failed, if a specific reason is identified, the subsequent step involves implementing necessary modifications to correcting the issue

10. Test Suspension and Resumption Criteria:

Testing for DyslexAI will stop if three tests in a row fail. Since one problem can cause others to fail, we will fix issues after three consecutive failures to avoid wasting time. Testing will start again once these problems are fixed, and we've checked that nothing else was broken by the changes. For example, if login tests fail three times in a row, we'll stop testing to fix the login issues. After fixing and verifying that login works and other features aren't affected, we'll resume testing the rest of the project.

11. Test Design and Execution: Manual/Automated

For DyslexAI, we will use both manual and automated testing. Our backend tests will be manual, and most of our frontend tests will be automated. We will use Jest for automated frontend tests and script/unit tests for our serverless API calls, while manual testing will cover backend functions. A special testing branch will be created in our GitHub project to handle these tests.

We will organize our tests into different groups based on features like user login, e-book searches, library, and accessibility modes. Automated tests will be created for each feature, allowing us to quickly and efficiently check for problems. If a test fails or we find a bug, we can easily rerun the test after fixing the issue. This makes retesting fast and thorough.

Manual testing will be used for parts that automated tests can't cover, such as certain user interface elements or serverless methods. In these cases, we will have pass or fail logs to manually record whether the test succeeded. This way, we ensure that both backend and frontend parts of DyslexAI are tested effectively, giving us a strong and reliable testing process.

12. Test Data & Defect Management;

For DyslexAI, we will manage our test data by storing it within our live database during testing to ensure that the data used is valid when making calls to the server. We will create test user profiles, including a test user account, a test admin account, and sample e-books to cover various functionalities of the application. This test data will mimic real-life data that the program will encounter, ensuring accurate and reliable testing. The test data will be created manually to ensure it meets our specific requirements. After each test, we will regularly check and validate that the functionality works correctly with the test data.

For defect management, each defect found during testing will be documented with a unique identifier, a description, severity level, and status to ensure all of us are aware of the bug. This documentation will help us track and prioritize fixes. When a bug is found, we will meet to discuss potential fixes and assign each bug to a. High-priority bugs will be addressed first. Once bugs are fixed, they will be retested to ensure the fix works and that no new issues have arisen. This systematic approach will help us manage test data effectively and ensure a smooth defect resolution process for DyslexAI.

13. Risk Analysis:

During testing for DyslexAI, we need to identify potential risks that may impact the user experience. We must ensure that the application can manage user data, e-book searches, library, and accessibility settings effectively. One major risk is the potential for security vulnerabilities, particularly in protecting user credentials and personal information. We need to verify that our database and authentication mechanisms are secure and robust.

Another risk is the possibility of functional bugs that could affect core features such as login, preference settings, and e-book searches. We must ensure that our application remains within the project's scope and meets all functional requirements and objectives to satisfy user needs.

Testing with real data is crucial to ensure that DyslexAI functions correctly in a production environment. This includes checking that all features work as intended and that the application handles real-world data accurately.

Additionally, we must assess the maintainability and scalability of DyslexAI to ensure it remains functional and efficient under different scenarios, such as increased user load or changes in user preferences.

To mitigate these risks, we will implement comprehensive security testing to identify and fix any vulnerabilities. We will conduct thorough functional testing to catch and resolve any bugs. Regular review meetings will ensure we stay within the project's scope and meet all requirements. By testing with real data and focusing on scalability and maintainability, we will ensure that DyslexAI is reliable and performs well in various conditions.

14. Role and Responsibilities:

Mahbubur is responsible for writing and executing unit tests for the front-end components of the DyslexAI application, ensuring that each individual component works as expected in isolation. After unit tests, he will move on to integration testing, combining different front-end components to ensure they work together seamlessly. Aaron works alongside Mahbubur in writing and executing unit tests for additional front-end components and will also participate in front-end integration testing, focusing on user interface consistency and functionality. Both will document any defects found during testing and follow the Test Data and Defect Management protocols. Saba focuses on integration testing for the backend components, ensuring that various backend services and databases interact correctly. She will also perform system testing to validate the entire system's functionality, performance, and reliability. Ishmail partners with Saba in backend integration testing, verifying the interoperability of backend services, and will also be responsible for system testing, ensuring the backend and front-end components function together as a whole system. Both will meticulously document any issues discovered during testing according to the established defect management process. The workflow involves Mahbubur and Aaron conducting unit tests, followed by front-end integration testing, while Saba and Ishmail concurrently conduct backend integration testing.