Testing

	Input testing	Results	Test purpose
username (Sign up)	Empty entries	username field empty	Check empty input to request for re-entry
	Entries more than 11 characters	Username is too long	Check long input to request for re-entry
	Entries less than 6 characters	Username is too short	Check short input to request for re-entry
	All entries are alphanumeric or digit	Weak username	Check input with letters or digits only to request for re-entry
	Entries need upper and lower case letters and digits.	Strong username	check if strong entry to considered as sucssefully sign up
	When all conditions become false	Medium username	excute when all condition fasle
password (Sign up)	Empty entries	password field empty	Check empty input to request for re-entry
	Entries more than 11. characters	Password is too long	Check long input to request for re-entry
	Entries less than 6 Characters	Password is too short	Check short input to request for re-entry
	All entries are alphanumeric or digit	Weak password	Check input with letters or digits only to request for re-entry
	Entries need upper and lower case letters and digits.	Strong password	Check if strong entry to considered as sucssefully sign u
	When all conditions become false	Medium password	Excute when all condition fasle
username and password (Sign up)	any field exist in file	Display "Username or password already exists" and "Sign up failed" and enter new inputs	Check exist of any entry in file to ask for different inputs

username and password (Sign up)	all field does not exist in file	Display "Sign up successful" and execute rest of condition	Check no exist of any entry in file to sign up successful
username and password (Log in)	-Username/password not found in file - Empty fields	Display "Incorrect username or password"	check empty fields or non-exist username/password in any file line.
	Username/password found in file	Display "Login successful" and execute rest of condition	check exist username/password in any file line
Words Guessing game	Empty guess	Display Please fill the inputs and execute rest of condition	Check empty input to request for re-entry
	Letters Guessed before	Display "You already guessed that letter" and execute rest of condition	Check letters guessed before to input to request for re-entry
	Correct guess word letter	Display "Correct letter!" and execute rest of condition	Check the guessed letter if inside the word
	Incorrect guess word letter	Display "Incorrect guess" and execute rest of condition	Check the guessed letter if not inside the word
	Correct word guess	Display "Correct!" and execute rest of condition	Check if word guess is correct
	incorrect word guess	Display "Incorrect guess" and execute rest of condition	Check if word guess is incorrect
	Enter all word letters	Display "You've guessed all the letters! The word is:", random_word) and update the score with execute rest of condition	Check that all letters guessed

	Enter 6 wrong guesses	Update the score and display "current guess score:{guess_score}" with execute rest of condition	Check if the guess has reached its maximum limit
	User enter correct letters	Display "Guessed Letters:" (the letters guessed with their location)	Display regularly to show the user progress
	User answer to play again==yes	Play the game again	Check if the user answers yes to play again
	User answer to play again==anything else	Stop the game and unhide the menu window to choose an option	Check if the user answers not yes to return to menu window
Definitions matching game	all three answers corresponding with Definitions	Display "Correct!" and update the score with execute rest of condition	Check that all answers match their corresponding definitions
	Answers don't have "abc" like j,k,l letters or may contain repeated letter	Display "Inputs must have (A, B, C)"	Check no repeated letters or any letters other than abc
	Any empty answer	Display "Fill all the inputs"	Check if any answer empty
	When all conditions become false	Display "Incorrect answers" with execute rest of condition	Excute when all condition fasle
	When all conditions become false and after two wrong guesses	Update the score and display f"Current matching score: {matching_score}" with execute rest of condition	Check if the guess has reached its maximum limit
	User answer to play again==yes	Play the game again	Check if the user answers yes to play again
	User answer to play again==anything else	Stop the game and unhide the menu window to choose an option	Check if the user answers not yes to return to menu window

Score update	When user reach max attempts without the correct answer (Words Guessing game)	Decrease the score of the specified game by one point.	To decrease the score of Words guessing game in the file
	When the user answers the question correctly (Words Guessing game)	Increase the score of the specified game by one point.	To increase the score of Words guessing game in the file
	When user reach max attempts without the correct answer (Definitions matching game)	Decrease the score of the specified game by one point.	To decrease the score of Words definitions matching game in the file
	When the user answers the question correctly (Words Guessing game)	Increase the score of the specified game by one point.	To increase the score of Words definitions matching game in the file

Checking test

File content storing

- Test if the scores, username, and password are stored correctly for different users.
- Test if the scores are properly updated after playing games.

User Interface Testing

- Test functionality of buttons and entry fields.
- Test window transitions such as sign in and log out.

Error Handling

- Test unexpected inputs to ensure the program handles errors.

System Testing

- Test the entire flow of the program by testing multiple users interacting.