Technical Overview

	Functions	Description
User-defined Functions	check(username)	The function responsible for verifying the security of a username before sign up
	check2(password):	The function responsible for verifying the security of a password before sign up
	open_signup_window()	The function excute when user click on sign up button which hide the main window and open window with sign up title
	signup()	validates user registration inputs, ensuring username availability and password criteria before storing data, completing sign-up.
	go_back()	The function responsible for returning from sign up window to the main window displaying the choice between signing up or log in
	open_login_window()	The function excute when user click on Log in button which hide the main window and open window with Log In title
	login()	Check username and password in data file for successful login. If located, open file with write username, assign scores to zero with header if not found .
	go_back()	The function responsible for returning from Log in window to the main window displaying the choice between signing up or log in
	open_menu_window(username)	Upon successful login, the function opens a menu window providing three options: guess game, definition matching, and log out.
	games_rules(username, game)	The function responsible for determining the user's level in the game is also tasked with identifying which specific game's score needs to be checked.
	word_guessing(username, menu)	The function executes upon clicking Words Guessing in the menu, a game initiates where the user guesses words or letters, with their score updating based on guesses.
	definitions_matching(username, menu)	The function executes when selecting definitions matching initiates a game where users match words with their letters in definitions, updating scores based on guesses.
	score_update(username, point, point2)	The function adjusts game scores based on in-game calls, either increasing or decreasing the score of one game while leaving the other game's score unchanged.
	log_out(menu)	The function responsible for logging out of a user account and returning to the menu window to allow interaction with different users
Built-in functions	all()	Help in checking by iterates on the items and return true if all item true
	isdigit() -- isalpha()	- checking if charcters have numbers - checking if charcters have letters

Built-in functions	any()	Help in checking by iterates on the items and return true if any item true
	islower() -- isupper()	- checking if charcters have lower case charcters - checking if charcters have upper case charcters
	set()	used to store multiple items in a single variable.
	withdraw()	Used to hide the windows which help to make program go smoothly
	Toplevel()	Create a window overlaying all other windows such as a signup window or login window.
	title()	Used to name title for the window
	get()	Retrieve a specific value associated with a particular key, for instance, when entering a password, enable access to that value
	readlines()	Return the list of lines and save it , also can return specific line
	strip()	Remove extra space to handle sensitive checking
	split()	Split the list for example to seprate the username and password by | operateor to be able acess easily
	lower() – upper()	- Convert chracters to lower case - Convert chracters to upper case.
	write()	Used to wite a content in file
	deiconify()	Used to unhide the windows which help to make program go smoothly
	destroy()	Used to destroy the windows which help to make program go smoothly
	Label()	Used to write a lables in window to claify it such as label beside useranme enrtry field
	pack()	Declares the position of widgets in relation to each other before placing them in the parent widget.
	Entry()	Used to take entries form user as input but the entries inside a window
	Button()	Create a button with the ability to display text indicating its function, and when clicked for example it initiates a predefined action function.
	lambda	Anonymous functions that can accept any number of arguments helped prevent immediate execution of the function without requiring a button click beforehand.
	DictReader()	Help to read the file as dictionary

Built-in functions	next()	Function responsible used to skip the header in the file
	open()	function is responsible for opening files in Python with a specific mode
	choice()	Function returns a randomly selected element from the specified sequence.
	len()	Function returns number of items in an object or characters in the string.
	input()	function takes input from the user
	print()	function prints a message to display it to user screen
	add()	Function is responsible adds an element to the set.
	join()	function is used to concatenate the elements of an iterable object
	sample()	Return a list of randomly specified number items from a sequence.
	shuffle()	Method takes a sequence and rearrange it in random order
	max()	returns the item with the highest value and can specify a minimum value using the key parameter
	append()	method appends an element to the end of the list.
	seek()	function allows to change the position of this file pointer within the file.
	writelines()	function writes the items of a list to the file
	read()	This function reads the entire file
	TK()	Function used while use GUI to create main window
	grid()	A function that arranges widgets by specifying their positions in rows and columns.
	mainloop()	The method starts an event loop, continuously monitoring user input (e.g., mouse clicks, keypresses), and remains activeuntil the user exits the program

Modules

from tkinter import *: This line imports all classes and functions from the Tkinter module

import csv: Imports the CSV module, which provides functions to read and write data in CSV format.

import random: Imports the random module, which provides functions to generate random numbers, select random elements from lists, and shuffle sequences.

Data structures	Description
Lists	Lists are used to store multiple items in a single variable such as - guessed_letters: It stores the guessed letters in the word guessing game. - random_rows: It stores randomly row form list of Words from the CSV file for the definitions matching game. - definitions: It stores definitions for the words in the definitions matching game. - updated_lines: It stores updated lines for the user's score in the score_update() function.
Dictionaries	Dictionaries are used to store data in key-value pairs such as - options: It maps options ('A', 'B', 'C') to shuffled definitions for the definitions matching game. - random_row: It stores a randomly chosen row from the CSV file - parts: It stores parts of each line split by the '|' character in various functions.
Sets	Sets are used to store unique elements. Such as - guessed_letters: It stores guessed letters in the word guessing game to ensure each letter is counted only once. - guessed_word: It stores the guessed letters of the word in the word guessing game.
CSV Reader.	The CSV reader is used to read data from CSV files
string	Strings are used to represent data such as - Storing usernames, passwords, words, definitions, topics - Displaying messages to users. - checking user inputs.
int	Integers are used to represent numeric data such as - Keeping track of scores in the games (e.g., guesses_left, guess_score,) - Counting and comparing numeric values (e.g., comparing points, levels)

programming techniques	Description
Functions	The code is divided into multiple functions, each function have specific purpose, which enhances code organization and structure.
console window & Graphical User Interface (GUI)	Combining a GUI with a console window lets users start actions via the GUI and play games in the window, enhancing professionalism and user interaction.
Conditional Statements	used to make decisions based on conditions, such as checking the validity of username and password inputs or controlling the flow of the program.
Looping Structures	The code use while which repeatedly execute certain parts of the code until a condition is met such as game over or when user exit
File Handling	Using CSV and TXT files to read and store data, which are responsible for generating program outputs
Randomization	The random module assist in generates words for the guessing game and three words with their definitions for the matching activity.
Input Validation	The program validates user inputs like usernames, passwords, and game answers, ensuring correct program flow based on specified conditions.

Reasoned decisions

Break the code to functions: Divide the code into chunks to efficiently manage errors without disrupting other functionalities, with a particular focus on handling user button clicks

Using dictionary and list to read from files: Dictionaries are organizing data with multiple fields, like CSV files to read it easily. Lists work for storing and accessing lots of data, especially when you don't need to name each field, like iterating over lines.

Using GUI: provide an easy way for users to interact with the program with navigate through menus, buttons, and other graphical elements.

Using random module: Enhance the game to introduce elements that less predictable and reduce repetitive patterns.

Variable Names: choosed descriptive variable names make code more readable and understandable for both the programmer and anyone else who may need to review or modify the code

Games Logic: Clear and understandable guidelines for dividing game’s part into sections, such as game rules, updating scores, guessing words, and matching definitions, each managed within its designated function with connection between them.