MIA Task 6

NOTE:

- There are 5 files (main - logic - backend) those 3 are the first 3 I made and there was no GUI
- Then in the extra days I added GUI in (main with gui - logic gui) and used the same backend file
- This documentation was made before adding GUI

Folder structure:

The folder contain 3 files:

1. Backend.py: where fitching the database occurs to generate the answer and to validate user guess
2. logic.py: it contains
   A function which takes the guess from the user
   A function which compare user guess with the correct answer
   A function to process each letter one by one and give each letter a color according to its state then make a list of the colors
   A function which take the list of the colors and the guess to combine them and return colored guess
3. main.py: easy-to-read code containing a loop which breaks after 6 tries unless the user guessed the word correctly before
   It runs all the needed functions and prints the suitable message for each condition

Logical structure:

The code starts in backend.py where there is a function responsible for fetching the data from the database and creating a list containing all the words in it

After that another function choose one word of this list randomly and assign it as the answer
Then we go to main.py where an infinite loop will be started
That's when the user is supposed to write his first guess and after each guess there is a counter which increases by one with each guess
After the user type his guess it gets snt to a function in the Backend to validate if it's a real word or a made up word by searching for it in the data list
If it was not found in the data list the compiler tells the user that he is only allowed to type real 5-letter English word and counter is reduced by 1 as this guess won't count
If it was found in the data list, It will be sent to a function in logic.py called check guess to check if it is actually the correct answer
If it turned out to be the correct answer we start celebrating with the user, break from the loop, end the program and go home
If it is not the correct answer "most likely It won't be" it goes back to logic file to check matches
Check matches main purpose is to make a list of 5 color codes with default being grey or white
First it starts by looping on each letter of the guess and comparing it with the letter with the same index in the answer. if it sees a match it goes to the colors list and change the color code with the same index to green
And if the 2 letters didn't match we add the letter of the guess to a dictionary with its index as its key
After checking every letter we enter a nested loop
First loop loops around each key and value in the dictionary second loop is to loop around each letter in the answer

Then we compare the letter in the dictionary to every letter in the answer and if it sees a match it goes to the colors list and change the color code with index equals to its key to yellow
And to prevent letters which are already correct and in their right places from being processed again in second stage, I made a variable called ans which is equal to the answer and whenever a letter is matched in first stage, its index in ans is changed to “-” then in second stage instead of comparing the real answer i compare “ans” with letters in the dictionary
After all of that the function returns the list of colors to another function whose purpose is to combine the letters of user guess with the list of colors to create a colored string
At the end, the colored word is printed for the user then he starts his next guess

Things to be added:

1. A script to sort the data set by letters so when we need to check if the user entered a valid word we can use binary search to search for it faster in the data set
2. Make a gui for the game using game.py
3. Making a json file where we can save highscores like least guesses solve
4. Also we can use time to save the fastest solve in the json file

Sadly I don’t have enough time to implement them as I still need to finish Task5 as I had an excuse