1. Problem Description:

The problem is to create a game of Hangman. The game should have a login system for both users and administrators, and should also keep track of high scores.

1. Distinguishing features of the project:

* Fully dedicated GUI for the game.
* The game has a login system for both users and administrators, allowing for different levels of access to the game.
* The game keeps track of high scores, and allows administrators to reset the high scores.
* The game has a user registration system, allowing new users to create an account and play the game.

1. Flow of Project

* The game starts with a login window for both users and administrators.
* For the user:
  + Once logged in, the user can play the game.
  + The user can guess a single alphabet at a time.
  + If the alphabet is in the secret word, it will be displayed on the screen.
  + If the alphabet is not in the secret word, the user will lose a guess.
  + The game will end when the user has guessed all the letters in the secret word or has run out of guesses.
  + If the user successfully guesses all the letters in the secret word before running out of guesses, they will be declared the winner and the game will end. The user will then be prompted to play again or exit the game, returning to the login window if they choose to play again.
  + However, if user is unsuccessful in guessing the word before the guesses run out, they will lose.
  + After the game for both cases of win or lose It will ask the user if he/she wants to play again? If not then it will display a goodbye message screen.
  + If the user wants to play again the game will return to the login window allowing the user to start again.
* For the admin:
  + After Authentication of admin with relevant password and username.
  + A window specific to the admin will open up and there will reside methods to Add words and reset the high scores.
* The game also keeps track of high scores.

1. Most challengi[[1]](#endnote-1)ng part for you while working on the project:

There were two parts that would be classified as Challenging:

1. The Most challenging part was actually implementing the Tkinter module as it was very lengthy and I had to be very careful in defining certain objects.
2. Developing some good algorithms for data processing was also very challenging.
3. Any new thing learnt in Python while working on the project:

I learnt a lot during the process of making this project:

1. During the project, I learned how to use the Tkinter library to create GUI in Python.
2. We also learned how to work with file I/O to keep track of high scores and user login information.
3. Individual contributions of each individual contributions of each group member in the project:
4. **Muneeb Ahmed (CS-22048):**
   1. Group Leader
   2. Lead Programmer.
   3. Asset Design.
5. **Ahzam Rehan Bari (CS-22046):**
   1. Research
   2. GUI concept.
   3. Game Tester.
6. **Abdullah Athar (CS-22041):** 
   1. Game Tester.
   2. GUI concept.
   3. Ideas for improvement.
7. Future expansions:

* Expanding the game to the web for better reach.
* Adding a leaderboard feature to display top scores of all the players.
* Adding a feature for users to play with friends by sharing a game code.
* Adding a feature for users to choose the level of difficulty of the game.
* Adding a feature for users to change their password.
* [[2]](#endnote-2)Adding a feature that would allow admins to manipulate the words(deleting or replacing words).

1. List of references:

* Tkinter documentation:
  + https://docs.python.org/3/library/tk.html
* High Scores and File I/O in Python:
  + <https://realpython.com/working-with-files-in-python/>
* Tkinter Tutorial:
  + <https://www.youtube.com/watch?v=TuLxsvK4svQ&t=3143s&ab_channel=BroCode>
* W3Schools Docs:
  + https://www.w3schools.com/python/

1. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

   TEST CASES:

   1. **WINNING CASE:**
      1. Game Window

   [↑](#endnote-ref-1)
2. [↑](#endnote-ref-2)