

# DSA Project Proposal

## **Team Title:**

Trie Word Game

## **Team Members:**

Sufiyan Aman

Shayan Raza

## **Data Structure:**

### **Trie (Prefix Tree)**

Trie is used for efficient retrieval of a key in a dataset of strings. It stores strings as keys and their associated values as nodes. Each node has a character and a pointer to its children nodes. Trie supports efficient prefix search, which means it can find all the words that start with a given prefix in linear time.

## **Application:**

### **Word Game:**

This is a fun and interactive game that allows the player to form valid words from a given set of letters. The player can earn points, time, or hints based on the words they form. The game has different levels that require the player to form words with certain criteria, such as length, prefix, or containing a specific letter. The game uses Trie to store all the valid words and their scores, and to check if the words meet the criteria or not. The game will also have a feature for hints that the player can enter a prefix and check if there is any valid word with that prefix. Since, Trie supports efficient prefix search, which means it can find all the words that start with a given prefix in linear time, it will be a useful data structure for checking if a partial word is valid or not, or for suggesting some possible words to the player (if needed). Another benefit of using Trie for this game is that Trie can also store additional information associated with each word, such as its score or difficulty. This is useful for calculating the points, time, or hints for the player. Trie can also save space by compressing the common prefixes of the words into a single node. This reduces the number of nodes and pointers needed to store the words, and improves the memory efficiency of the data structure.