URL: https://github.com/Sam-Scott-McMaster/the-team-assignment-team-35-formula4

Team Name & Number: Formula4, Team 35

1. **Title:** WorDex

2.

   A. Our app is a command-line Wordle clone programmed in C named WorDex where the user attempts to guess a randomly selected 5-letter word within six attempts. When the program starts, it chooses a target word from a provided TXT file, and the user enters one guess per attempt. Each guess is checked to ensure it's valid, exactly five letters, all alphabetical, and lowercase. After every input, the app provides clear feedback on each letter.
   B. The expected users for this project are my team members, TA/Instructors and anyone who is running it for testing purposes. To give a little more context, the project will be used on terminal/IDE consoles. Users are expected to have basic typing and reading ability and nothing more.
   C. List of possible constraints -
      a. Performance - Word list is tiny (TXT file). Random selection is trivial. There is no real performance pressure.
      b. Memory - We are only using memory for the list of words, storing the target word, and the user guesses.
      c. User Interface - Since this is a command line interface, the entire user interface will be on the terminal/IDE console with a simple design.
      d. A key constraint is the limited development time, so each module must be built independently while still integrating early to avoid last-minute issues. This requires strict boundaries between components, following a clear flow from the Random Word Generator to Validation, Comparison, and finally the interface.
      e. There are limited team resources, as each team member has a dedicated role, so no one can depend on others to complete their modules.
      f. The input constraints are that the app only accepts 5-letter lowercase words with no digits or symbols, and the user has a maximum of six attempts to guess the target word.
      g. The file constraints affect the random word generator since it depends on the quality and formatting of the TXT word list. The program must handle issues like empty lines, invalid words, or words that are not exactly 5 letters, filtering out anything that does not meet the requirements.

3.

Team Members:
Nithya Majeti, 400569912, majetin5017
Mohitkumar Gedela, 400569965, gedelm1
Lohitashwa Madhan, 400600568, madhanl
Zainab Mirza, 400586062, mirzaz13

4.

Authentication and Stats: Nithya

- Registering a new user or like for a logic for an existing user
- Storing the user credentials in a text file named userprofiles.txt
- Validate usernames and passwords

Word Comparison Logic and User Interface: Mohit

- Load the words from the stored file (words.txt) and choosing a random word
- Validating user input
    - The word is 5 letters and is an appropriate word
- Comparing each position of the letter with the target letter and colour the words exactly such as gray for non-existing letters, yellow for letters which exist in the target word but are not in the correct position and green for letters that are in the correct position
- Ensuring that there's a maximum limit of 6 guesses
- Communicating with the authentication and stats module about the amount of guesses the user had and if they won or loss
- The board

Statistics: Zainab

- Calculating and different statistics for user from the stats.txt file
    - Longest log in streak
    - Current streak
    - Average word guess count
    - How many games with each number of guesses (Games played with 1 guess: 3)
    - Wins, Losses
- Updating statistics for the user
- Fetching their Statistics
- Setting default Statistics

Leaderboard and Game History: Lohit

- Doing a full game history log system that stores every completed match with timestamps and prints it in a clean table format with coloured WIN/LOSS results (green for WIN and red for loss), ordered from most recent to oldest so users always see their latest games first.
- Setting up timestamps for every game entry

- Setting up a global leaderboard that reads all recorded games and displays all users in a ranked table based on the highest number of wins, also showing games played, losses, win rate, and average guesses.

Incremental Plan:

Increment 1.1:	**Purpose:** Complete all core modules functionality **Assignees:** Lohit, Mohit, Nithya, Zainab **Deliverables:** - Generate working logic only for parts which are not interdependent with other modules - Test cases	**Target Date:** November 28, 2025
Increment 1.2:	**Purpose:** Connect each module together and everything with the main module **Assignees:** Lohit, Mohit, Nithya, Zainab **Deliverables:** Check if each module is completely correct, connect with other modules and main	**Target Date:** November 28, 2025
Increment 2:	**Purpose:** Report and final test run **Assignees:** Lohit, Mohit, Nithya, Zainab **Deliverables:** Ensure all the requirements are met, build the entire main navigation system check for any system fails and have the report generated alongside the final version of the README.md and a tutorial section	**Target Date:** December 1, 2025