Final Project: Hangman Game

1. Project Overview

- Develop a simple Hangman game using Python.
- The game should allow a single player to guess letters in a hidden word.

2. Basic Requirements

- **Word List**: The game should use a predefined list of words from which a random word is selected for each game.
- **Input**: The player should be able to input one letter at a time.
- **Display**: Show the word with guessed letters revealed and unguessed letters as underscores.
- **Guess Limit**: The player should have a limited number of incorrect guesses (e.g., 6).
- **Win/Loss Condition**: Notify the player if they win (all letters guessed) or lose (exceeded guess limit).

3. Game Mechanics

- **Random Word Selection**: Select a random word from the word list at the start of each game.
- **Guess Validation**: Ensure the player inputs only one letter and handle invalid inputs gracefully.
- **Repeated Guesses**: Keep track of guessed letters and notify the player if they guess a letter they have already guessed.
- **Update Display**: Update the displayed word and remaining guesses after each guess.

4. User Interface

- **Game Prompts**: Provide clear instructions and feedback (e.g., incorrect guesses left, current word state, letters guessed so far).

5. Additional Features

- **Hint System**: Provide hints to the player if they are stuck.
- **Score Tracking**: Track and display the player's score across multiple games.

6. Code Quality

- **Readability**: Write clean, readable, and well-commented code.
- **Modularity**: Organize the code into functions as appropriate.
- **Error Handling**: Implement error handling for edge cases (e.g., non-alphabetical input).

7. Evaluation Criteria

- **Functionality**: The game meets all the basic requirements.
- **Code Quality**: The code is clean, well-structured, and well-documented.
- **User Experience**: The game is user-friendly and provides clear feedback.
- **Optional Features**: Implementation of any additional features.