Word Game

Create an application to deliver a word game

This challenge is to implement functions of a word game which takes user-submitted words, scores them, and maintains a high-score list. As the challenge is only to test your PHP skills, rather than storing the high-score table in a database, we'll just save it in memory, and we'll simulate the user entering words by calling the relevant functions within the script.

1. The game is constructed with a random base string of letters a to z, e.g. the string **areallylongword** or **dgeftoikbvxuaa.**
2. This base string can contain repeated letters.
3. The player attempts to create words out of the letters, and scores one point for each letter used.
4. The maximum score is therefore the length of the starting string (IF a valid English word can be made using ALL the letters).
5. Individual letters can be used as many times as they appear in the base string.
6. To score, a submission must be a valid English word.
7. Validity should be checked by loading the word list file wordlist.txt and confirming that the submitted word is present in the list.
8. The game should maintain in memory a list of the top ten highest-scoring submissions (word and score).
9. Duplicate words are NOT allowed in the high score list i.e. a word can only appear once in the high score list.
10. Throughout this test, you need not worry about whitespace, special characters, or uppercase: you may assume that both the base string of letters and all submissions contain none of these.
11. Here are some examples of valid and invalid submissions, when starting with the base string "areallylongword". You may like to use these to test your code.
    - **"no"** is an acceptable submission, with score 2
    - **"grow"** is an acceptable submission, with score 4
    - **"bold"** is NOT an acceptable submission (there is no "b" in the base string)
    - **"glly"** is NOT an acceptable submission (it is not present in the word list)
    - **"woolly"** is an acceptable submission with score 6
    - **"adder"** is NOT an acceptable submission (there is only one "d" in the base string)

12. There is no need to make a user interface or to store the words in a way that will persist between invocations of the program.
13. It is sufficient simply to implement the functions in the provided file wordgame.php file (click to download). You may like to add some additional code to test or demonstrate the functionality, but this is not mandatory.
14. You may choose whether your program will be run via a web server or using the php command-line

tool, and format any output accordingly. There is no need to provide any fancy formatting - as long as it is readable.

15. If any of the instructions is unclear to you, please make a reasonable assumption and document your assumption in the code with a comment.
16. You have two hours to complete the test.
17. Please submit your solution by following the instructions provided.
18. Your implementation should be production quality. Use any tools, approaches, resources, and information you would normally use to complete production-quality code.
19. However, please bear in mind that this test is intended to allow you to show your ability with pure PHP, rather than your skill in utilising frameworks and libraries.