Blossom Game Helper Project

The first coding project I did without any provided steps or guidelines was creating a Wordle clone. I decided that, for my second project, I would like to try to make a page to help players maximize their scores in Merriam-Webster’s Blossom daily word game. I did not want to completely automate the process, as I wanted to leave room for the player to still make decisions. I wanted the site to merely display some of the highest-value words for each round.

Background: Gameplay

Merriam-Webster’s Blossom game is similar to the NYT’s Spelling Bee and other casual word games. Each day, a new set of seven letters is provided, depicted as petals of a flower, with one letter in the middle and one petal highlighted. Players have twelve turns to create a “bouquet” of twelve words.

For each turn:

* The word must contain at least four letters.
* The center letter must be used at least once.
* The word may not contain letters other than the seven provided.
* Provided letters may be used more than once.
* No proper nouns, hyphenations, or foul language are permitted.
* Only US English spellings are permitted (as Merriam-Webster is US-based).
* The highlighted letter changes with each turn, and earns bonus points for each use.

Scoring:

* Four-letter words = 2 points
* Five-letter words = 4 points.
* Six-letter words = 6 points.
* Seven-letter words = 12 points.
* Each additional letter = +3 points.
* The petal outlined in yellow (which changes with every word played) contains a bonus letter. Bonus letter = +5 points per use.
* Pangrams (words which contain all 7 letters) = +7 points.

Logic

I want my project to contain a single HTML page with a form element that has three input fields and a submit button. The first field will be for the player to input the day’s provided letters, and the second will be for the player to provide the required letter. The third will be for the player to provide the bonus letter. Since the bonus letter changes for every round, the player will have to fill out this form for every round.

The Blossom game does not choose bonus letters randomly. Rather, it cycles through the six petals in order, twice, to make twelve “blossoms.” So it is possible that I could make my project provide the bonus letter automatically, but I think that sounds like an additional challenge to play with later.

Once the player hits the submit button, I would like my project to save that input. I think an array is likely the best option, though a set is also a possibility. Then I would like my project to iterate through Merriam-Webster’s dictionary API and include all words that contain only the provided letters. Then I would like it to compare that list to the rules of the game and sort the list according to how high each word can score, and provide that list to the player in ranked order. The player is free to choose the highest-scoring word, or another word entirely. I deliberately do not want that choice made for him or her.

I am going to begin, for ease of testing purposes, with a small hard-coded array of words rather than fetching from Merriam-Webster. That part I will implement later in the process. So the first few steps of this process are as follows:

1. Create HTML page containing a form element with three input fields and a submit button.
2. Link to an external .js file and, in that file, create the necessary JS to save the values from the HTML inputs to variables. The required and bonus letters will be single-character strings. The full set of provided letters will be saved as an array.
3. The provided letters array must also contain the required and bonus letter. To ensure this, I will do two things. First, write clear instructions to the user to include those in the required letters. Second, to add some script that looks at the array and, if these values are not in it, add them. This serves as a backup in case of human error.

Now I will have a hard-coded global array containing 10-20 words, as well as an array of single letters. I want to create a new array that excludes all words from the global array that contain letters other than the permissible ones. This will require a loop of some kind, possibly nested loops.

Problem #1: I need to make sure the array of provided letters contains both the required and bonus letters, and since there are three input fields that may not be clear to the user. Ways to combat this:

* I can try to write clear directions to the user, but this still does not completely safeguard against user error.
* I can constrain the provided letters field to only accept inputs of 7 letters, ensuring the user must provide all seven letters (including the required and bonus letter).
* I can create some “if” statements to check whether the provided letters contain the required and bonus letter, but this likely is not necessary if I constrain the input field. It may be best practice, however.