# React Assignment – Wordle

**Assignment**: Create a [Wordle](https://www.nytimes.com/games/wordle/index.html) duplicate using React

**References**:

Keyboard

<https://hodgef.com/simple-keyboard/getting-started/react/>

<https://hodgef.com/simple-keyboard/documentation/options/layout/>

CSS

<https://tailwindcss.com/docs/guides/create-react-app>

Local Storage:

<https://developer.mozilla.org/en-US/docs/Web/API/Window/localStorage>

Install react-simple-keyboard via npm

npm install react-simple-keyboard

Import Keyboard into your project

import ***Keyboard*** from 'react-simple-keyboard'

Use the following code to initialize a keyboard with the correct layout (you may need to implement button themes and the onKeyPress function)

<Keyboard  
 onKeyPress={onKeyPress}  
 layout={{  
 'default': [  
 'Q W E R T Y U I O P',  
 'A S D F G H J K L',  
 '{enter} Z X C V B N M {bksp}',  
 ],  
 }}  
 buttonTheme={[]}  
/>}

**Instructions:**

* Fork the jumpstart repository and use your directory as a working directory
  + <https://github.com/sdinteractive/jumpstart/tree/master/2022-01>
* Create wordle-assignment-1 directory to use for your application
* Use Create React App to get a basic frontend application running
* Use Tailwinds CSS (or a similar CSS framework of your choice that works well with react. The framework is not as important here; what’s important is the exercise in using the framework inside a react application)
* Create a dictionary of at least 5 words of your choice, each word must be exactly 5 letters long and be valid words in the English dictionary
* If the application already has a word for the day, use the value from localStorage
* If the application does not yet have a word for that day, allow the application to choose one word from the list
  + Allow the application to randomly choose one word from the list
  + Set that word as the current word of the day in localStorage on the user’s browser, associated with the current date
* Once the user types the first letter, start a timer. Keep track of when the user started the game. Save this value in local storage
* On page load, render a grid of light grey bordered squares, 6 rows and 5 columns
* Below the 6x5 grid, display a QWERTY keyboard using `react-simple-keyboard`, including the Enter key and a Backspace
* Clicking on a letter should add it to the next spot on the current row
* Clicking the enter key should submit the user’s selection for that row
  + The enter key should be disabled if there are not 5 letters entered into the row
  + Identify any green tiles – letters in the correct spot
  + Identify any black tiles – letters that are not in the word
  + Identify any yellow tiles – letters that are in the word but in the wrong spot
* If the user guesses the word correctly, display an alert messaging indicating they have won, calculating their score, and showing them the overall time it took for them to complete the puzzle
* A backspace should remove a letter from the active row
  + The backspace key should be disable if there is not an active letter to remove
* Refreshing the page should restore the progress
* At the start of every new day (per user’s time zone) the progress and word should be reset, along with the win-status of the day and the time the user started playing

# Acceptance Criteria

As a user

When I open wordle for the first time on a given day

Then I should see a blank wordle board

As a user

Given I have already won wordle that day

When I navigate to the page or refresh the page

Then I should see the current game status and I should not be able to interact with the game board

As a user

Given I have already won wordle that day

When I navigate to the page or refresh the page

Then I should see a share button

As a user

Given I have already won wordle that day

When I click the share button

Then I should see an option to share my board (using emojis of colored squares, just like original wordle)

As a user,

When I type in 5 letters and hit enter,

Then I should see the characters highlighted, matching the status of the letter

As a user

Given I have not yet completed the wordle for the day

When I navigate to the wordle board

Then my progress should be restored, preserving the letters I have already entered and the space on the board I am on

And the game should remember whether I have already submitted a row or whether the row has not yet been submitted

As a user

When I load the page and it is a new day for my time zone

Then my progress from the previous day should be reset

As a user

When I type in a letter

Then the letter should be stored in uppercase

As a user

When I hit the key on my keyboard

Then the game should react as though I had pressed the virtual keyboard

And all characters other than the letters, backspace, and enter should be ignored

As a user

When I get a letter that is in the correct space in the word of the day

Then the tile should turn green

And the keyboard tile should turn green

As a user

When I enter a letter that is not in the word of the day

Then the tile should turn dark grey for that tile or tiles

And the keyboard tile should also go dark grey and become disabled

As a user

When I have not entered 5 letters

Then the enter button should be disabled

As a user

When I have entered 5 letters

Then the enter button should become enabled

As a user

When I have not entered any letters on my row

Then the backspace button should be disabled

As a user

When I have entered letters onto my board

Then the backspace button should be enabled

As a user

When I have won the wordle of the day

Then the keyboard should no longer be displayed

# Bonus: Implement keydown listener

Create an event listener hook

import {useRef, useEffect} from "react";  
  
const useEventListener = (eventName, handler, element = ***window***) => {  
 const savedHandler = useRef();  
  
 useEffect(() => {  
 savedHandler.current = handler;  
 }, [handler]);  
  
 useEffect(() => {  
 const eventListener = (event) => savedHandler.current(event);  
 element.addEventListener(eventName, eventListener);  
 return () => {  
 element.removeEventListener(eventName, eventListener);  
 };  
 }, [eventName, element]);  
};  
  
export default useEventListener;

Create a keydown listener

useEventListener("keydown", ({key}) => {  
 if (key === 'Enter') {  
 key = '{enter}'  
 } else if(key === 'Backspace') {  
 key = '{bksp}'  
 } else {  
 key = key.toUpperCase();  
 }  
 onKeyPress(key)  
});

# Brian’s Notes for developing the exam

30 minutes in

Chart

Description automatically generated

1:15 in and can add letters to a row

1:35 in and enable/disable backspace and enter keys

2:00 in and can add color to tiles / map them to a matrix for matches on g/y/b

2:45 in and have random words per day using local storage

3:15 in and I restoring preserved status from local storage

3:20 in and I can check whether a word is a real word in our dictionary

3:30 the game was playable to determine whether it won – was able to show the tiles with colors in he keyboard below

A picture containing table

Description automatically generated