

# Assignment 3

---

This file document the Wordle game updes with php and Ajax.

## Overview

---

We used php to build a top 5 best score board, which will use `$_SESSION` array to store the best 5 attempts to solve the Wordle game. (We choose 5 since Wordle is different from the game Yatzy). And a global variable `attempts` to keep track the total number of attempts of user, when a user win the wordle, the attempts will increase by one, and decrease by 1 if the player loses all the chances.

## Ajax

We use the axis to call perform the Ajax functions, the details are in the comment of the code.

```
import axios from 'axios'

// Create a object to send the ajax request to my php server
const apiClient = axios.create({
  baseURL: 'http://localhost:8080',
  // Include the cookies,
  withCredentials: true,
  // Format is json
```

```
headers: {
  Accept: 'application/json',
  'Content-Type': 'application/json'
}
}

export default {
  // Get the $_SESSION array from the player.php, to get all the user attempts
  getMessage() {
    return apiClient.get('/player.php')
  },
  // Put new attempts of a player into the $_SESSION array
  sendMessage(player) {
    return apiClient.post('/player.php', player)
  },
  // Close and destroy the $_SESSION array, to refresh the score board
  destroySession() {
    return apiClient.get('/quit.php')
  },
  // To update the global variable attempts into the $_SESSION array
  postAttempts(attempts) {
    apiClient.post('/attempts.php', JSON.stringify({ attempts }))
  },
  // Get the global variable attempts from $_SESSION array
  getAttempts() {
    return apiClient.get('/attempts.php')
  },
  // Check if a word matches the answer
  checkWord(word) {
    return apiClient.post('/checkWord.php', word)
  },
  // Get a random word
  getWord() {
    return apiClient.get('/word.php')
  }
}
```

```

getAttempts() {
    return apiClient.get('/attempts.php')
},
// Check if a word matches the answer
checkWord(word) {
    return apiClient.post('/checkWord.php', word)
},
// Get a random word
getWord() {
    return apiClient.get('/word.php')
}
}

```

## Global Variable: Attempts

This variable is aim to track the total number of attempts the player since the game starts. Since it's stored in the `$_SESSION` array, when u refresh the page, it will not be reset to 0.

We update the variable when the user either win or lose a game:

```

const handleKeydown = async (event: KeyboardEvent) => {
    ...
    // Pass the variable to the attempts.php
    api.postAttempts(php_attempts.value++)
    ...
}

```

## attempts.php

```

if ($_SERVER["REQUEST_METHOD"] === "POST") {

```

## attempts.php

```
if ($_SERVER["REQUEST_METHOD"] === "POST") {
    $json = file_get_contents("php://input");
    // Get the updated variable's value
    $data = json_decode($json, true);

    // Initialize Attempt_Number if not already set
    if (!isset($_SESSION["Attempt_Number"])) {
        $_SESSION["Attempt_Number"] = 1;
    }

    if (json_last_error() !== JSON_ERROR_NONE) {
        http_response_code(400);
        die(json_encode(["error" => "Invalid JSON: " . json_last_error_msg()]));
    }

    if (isset($data["attempts"])) {
        // Update the variable
        $_SESSION["Attempt_Number"] += $data["attempts"];
    }
}

} elseif ($_SERVER["REQUEST_METHOD"] === "GET"){
    echo json_encode($_SESSION);
}
```

# Scoreboard

In the `player.php` we store a object:

```
let player = {
  // Global variable
  a_number: php_attempts.value,
  // How many times player has tried
  a_attempts: 7 - life.value
}
```

into the session. In the method `handleKeyDown`, we pass this object to the `player.php`:

```
// Sending the input word to the php server
const word = inputWord.value[currentRow.value].join('')
/*
Sends the word, correct answer, row and column information to the checkWord.php,
which will return the game state: (win, lose, continue), and the index array to
display the correct color of the div.
*/
const response = await api.checkWord({
  word,
  answer: answer.value,
  currentRow: currentRow.value,
  currentCol: currentCol.value
})
// Updates the index array
index.value = response.data.index
// Update the div color
evaluateBoard()
```

```
evaluateBoard()
// If user's word is correct
if (response.data.result === 'win') {
  congrats()
  let player = {
    a_number: php_attempts.value,
    a_attempts: 7 - life.value
  }
  // Sends the current attempt number and attempts
  api.sendMessage(player)
  // Update the total attempt number (global variable)
  api.postAttempts(1)
  setTimeout(() => {
    resetBoard()
    resetRowCol()
    fetchMessage()
  }, 3000)
} else if (response.data.result === 'lose') {
  // Lost, waste one life
  lose()
  api.postAttempts(1)
  setTimeout(() => {
    resetBoard()
    resetRowCol()
    fetchMessage()
    life.value = 6
  }, 3000)
} else {
  // Go to the next line
  popUp()
  currentRow.value++
  currentCol.value = 0
  life.value--
```

```
currentCol.value = 0
life.value--
index.value = response.data.index
}
}
```

In the checkWord.php:

```
<?php
if ($_SERVER["REQUEST_METHOD"] === "POST") {
    $json = file_get_contents("php://input");
    $data = json_decode($json, true);

    if (json_last_error() !== JSON_ERROR_NONE) {
        http_response_code(400);
        die(json_encode(["error" => "Invalid JSON: " . json_last_error_msg()]));
    }

    if (isset($data['word']) && isset($data['answer']) && isset($data['currentRow']) &&
        isset($data['currentCol'])) {
        // Get the word, answer, row, and col.
        $word = strtoupper($data['word']);
        $answer = strtoupper($data['answer']);
        $currentRow = $data['currentRow'];
        $currentCol = $data['currentCol'];
        // Initialize the index array with -1
        $index = array_fill(0, 5, -1);
        // Check the word if match
        for ($i = 0; $i < strlen($word); $i++) {
            if ($word[$i] === $answer[$i]) {
                $index[$i] = 1;
            } elseif (strpos($answer, $word[$i]) !== false) {

```

```
$currentRow = $data['currentRow'];
$currentCol = $data['currentCol'];
// Initialize the index array with -1
$index = array_fill(0, 5, -1);
// Check the word if match
for ($i = 0; $i < strlen($word); $i++) {
    if ($word[$i] === $answer[$i]) {
        $index[$i] = 1;
    } elseif (strpos($answer, $word[$i]) !== false) {
        $index[$i] = 0;
    }
}
// Win
if ($word === $answer) {
    echo json_encode(["result" => "win", "index" => $index]);
}
// Lose
elseif ($data['currentRow'] >= 5) {
    echo json_encode(["result" => "lose", "index" => $index]);
}
// Continue
else {
    echo json_encode(["result" => "continue", "index" => $index]);
}
} else {
    http_response_code(400);
    echo json_encode(["error" => "Missing required fields"]);
}
} else {
    echo json_encode(["error" => "Life not set"]);
}
?>
```

In the player.php:

```
if ($_SERVER["REQUEST_METHOD"] === "POST") {
    $json = file_get_contents("php://input");
    // Get the player object
    $data = json_decode($json, true);

    if (json_last_error() !== JSON_ERROR_NONE) {
        http_response_code(400);
        die(json_encode(["error" => "Invalid JSON: " . json_last_error_msg()]));
    }

    if (isset($data['a_number']) && isset($data['a_attempts'])) {
        // Store the number of attempts and the sucess attempts into new variables
        $attempt_number = $data['a_number'];
        $attempts = $data['a_attempts'];
        // Put into the $_SESSION array
        $_SESSION["Attempt_{$attempt_number}"] = $attempts;
        // Return this array (optional)
        $response = ["Attempt " . $attempt_number => $_SESSION["Attempt_{$attempt_number}"]];
        echo json_encode($response);

    }
} elseif ($_SERVER["REQUEST_METHOD"] === "GET") {
    // Get the $_SESSION array
    echo json_encode($_SESSION);
}
```

Lastly, we will retrieve the both the global variable and the player attempts from the session:

```
async function fetchMessage() {  
  try {  
    const session = await api.getMessage()  
    const session_object = session.data  
    console.log(session_object)  
    const tbody = document.getElementsByTagName('tbody')[0]  
    // Filter out 'Attempt_Number' and then sort the attempts in ascending order  
    const session_array = Object.entries(session_object)  
      .filter((player) => player[0] !== 'Attempt_Number' && player[0] !== 'answer')  
      .sort((a, b) => a[1] - b[1])  
  
    // Only keep the top 5 attempts  
    const top_5 = session_array.slice(0, 5)  
  
    top_5.forEach((player) => {  
      if (player[0] === 'Attempt_Number' || player[0] === 'answer') {  
        return  
      }  
      // Create the table elements  
      const tr = document.createElement('tr')  
      const td_key = document.createElement('td')  
      const td_value = document.createElement('td')  
  
      td_key.innerText = player[0].substring(8)  
      td_value.innerText = player[1]  
  
      tr.appendChild(td_key)  
      tr.appendChild(td_value)  
  
      tbody.appendChild(tr)  
    })  
  } catch (error) {  
    console.error(error)  
  }  
}
```

```
        tr.appendChild(td_key)
        tr.appendChild(td_value)

    tbody.appendChild(tr)
    })
} catch (error) {
    console.error('Error fetching message:', error)
}
}
```

## Get Random Words

We have a `word.php` to generate a random word

```
<?php
...
$words = [
    'apple', 'about', 'above', 'actor', 'acute', 'adopt', 'asain', 'aside', 'avoid', 'aware',
    'baker', 'bland', 'blunt', 'broad', 'brush', 'brief', 'bread', 'break', 'broke', 'below',
    'carry', 'catch', 'cause', 'cedar', 'chant', 'claim', 'class', 'climb', 'clear', 'cendy',
    'dance', 'dandy', 'death', 'debit', 'decoy', 'depth', 'delay', 'daddy', 'dirty', 'doubt',
    'eagle', 'early', 'earth', 'easel', 'eject', 'ethic', 'equal', 'event', 'every', 'exact',
    'fable', 'facet', 'faith', 'fancy', 'feast', 'floor', 'first', 'final', 'flame', 'floor'
];
...
if ($_SERVER["REQUEST_METHOD"] === "GET") {
    $random_word = $words[array_rand($words)];
    $_SESSION['answer'] = $random_word;
```

```

$words = [
    'apple', 'about', 'above', 'actor', 'acute', 'adopt', 'asain', 'aside', 'avoid', 'aware',
    'baker', 'bland', 'blunt', 'broad', 'brush', 'brief', 'bread', 'break', 'broke', 'below',
    'carry', 'catch', 'cause', 'cedar', 'chant', 'claim', 'class', 'climb', 'clear', 'cendy',
    'dance', 'dandy', 'death', 'debit', 'decoy', 'depth', 'delay', 'daddy', 'dirty', 'doubt',
    'eagle', 'early', 'earth', 'easel', 'eject', 'ethic', 'equal', 'event', 'every', 'exact',
    'fable', 'facet', 'faith', 'fancy', 'feast', 'floor', 'first', 'final', 'flame', 'floor'
];

...
if ($_SERVER["REQUEST_METHOD"] === "GET") {
    $random_word = $words[array_rand($words)];
    $_SESSION['answer'] = $random_word;
    echo json_encode(["answer" => $_SESSION['answer']]);
}
?>

```

We call this in our JS:

```

const fetchWord = async () => {
    try {
        const response = await api.getWord()
        answer.value = response.data.answer
    } catch (error) {
        console.error('Error fetching word:', error)
    }
}

```

# Clean Scoreboard

We also include a button to clean the session, so the player can refresh their score board!

```
<?php
// Start
session_start();
...

if ($_SERVER["REQUEST_METHOD"] === "GET") {
    // Unset first
    session_unset();
    // Destroy the session
    session_destroy();
}
echo json_encode(["status" => "All session variables destroyed"]);
?>
```

## Button

```
function destroy() {
    // Use the api object to call the method
    api.destroySession()
}
```

# Api

```
import axios from 'axios'

const apiClient = axios.create({
  baseURL: 'http://localhost:8080',
  withCredentials: true,
  headers: {
    Accept: 'application/json',
    'Content-Type': 'application/json'
  }
})

export default {
  ...
  // Method to destroy the session
  destroySession() {
    return apiClient.get('/quit.php')
  }
  ...
}
```

# State of the game

---

## 1 Initial State

Run the following command to start the game

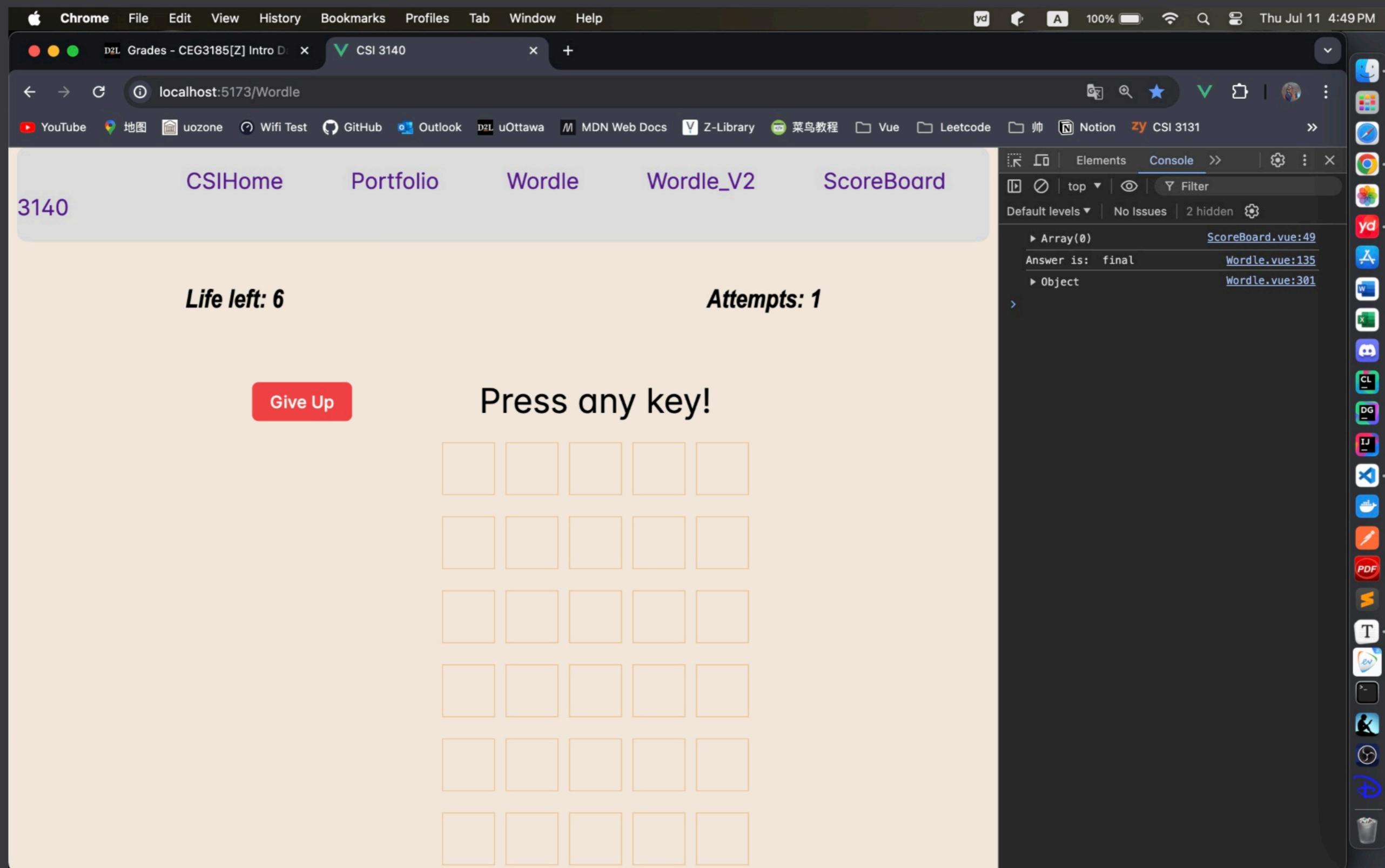
```
Wordle git:(main) ✘ npm run dev
```

Run this to start the php server

```
→ Wordle git:(main) ✘ php -S localhost:8080 -t src/api
[Wed Jul 10 15:49:04 2024] PHP 8.3.8 Development Server (http://localhost:8080) started
```

The game should look like

(The view Wordle(from assignment 2) and Wordle\_V2 are exactly the same, but using different server. Wordle use the JS to compute the logic, but Wordle\_V2 use Php. So Wordle is just a **old version**)



**Table**

Attempt Number	Attempts
----------------	----------

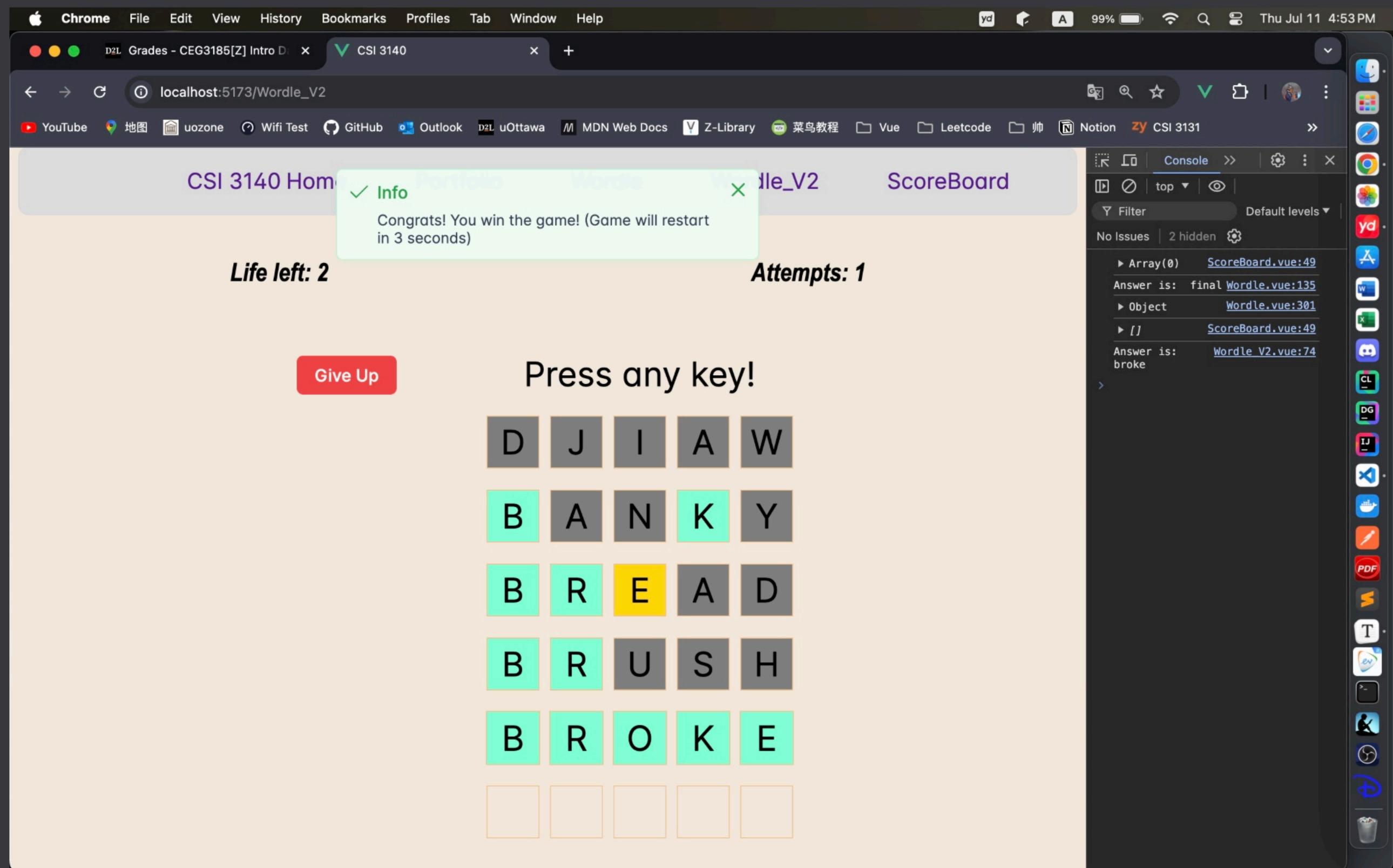
### Clean Board

```
▶ Array(0)      ScoreBoard.vue:49
Answer is: final Wordle.vue:135
▶ Object        Wordle.vue:301
▶ []            ScoreBoard.vue:49
>
```

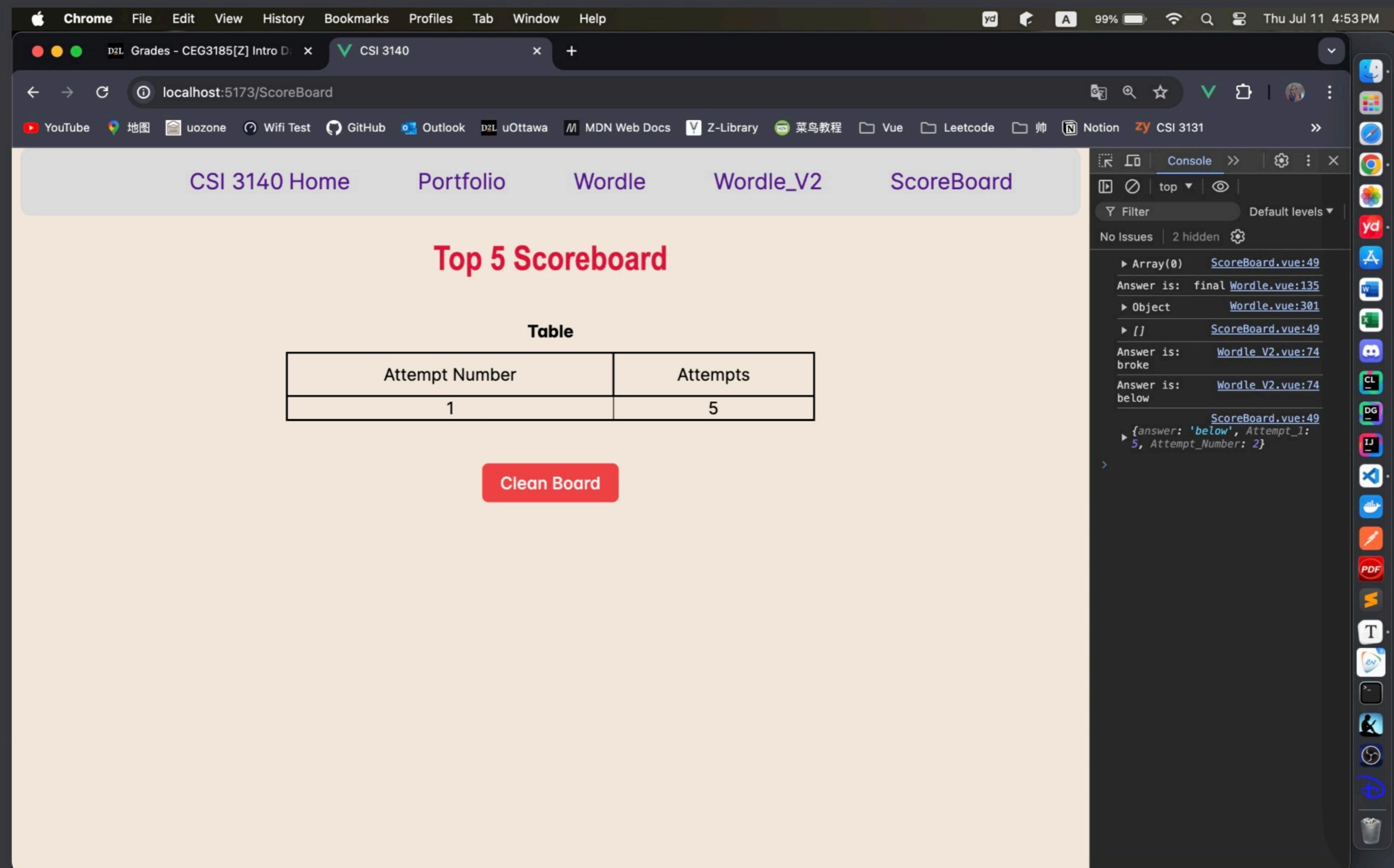
## 2 Playing

Now start our first trial

You can see on the terminal, the correct answer is: broke, and we have used 4 lives, if we enter the correct answer in the 5th line, then our attempt number is 1(1st round), and the attempts is 5.



Lets check out scoreboard, you can see the table has been updated, the Attempt number indicates n-th round of player's result, and the right column is the number of attempts player used.



CSI 3140 Home    Portfolio    Wordle    Wordle\_V2    ScoreBoard

## Top 5 Scoreboard

Table

Attempt Number	Attempts
1	5

Clean Board

CSI 3140 - localhost:5173/ScoreBoard

Console

```
Array(0) ScoreBoard.vue:49
Answer is: final Wordle.vue:135
> Object Wordle.vue:301
> [] ScoreBoard.vue:49
Answer is: Wordle V2.vue:74
broke
Answer is: Wordle V2.vue:74
below
ScoreBoard.vue:49
> {answer: 'below', Attempt_1: 5, Attempt_Number: 2}
```

Let's play again, now attempt number is 2 since we are in our 2nd round. And we are try to enter the correct answer in the first try!

Chrome File Edit View History Bookmarks Profiles Tab Window Help

99% Thu Jul 11 4:54 PM

D2L Grades - CEG3185[Z] Intro D... x CSI 3140 x

localhost:5173/Wordle\_V2

YouTube 地图 uozone Wifi Test GitHub Outlook uOttawa MDN Web Docs Z-Library 菜鸟教程 Vue Leetcode 帅 Notion CSI 3131

CSI 3140 Home

Info

Congrats! You win the game! (Game will restart in 3 seconds)

Life left: 4

Attempts: 2

Give Up

Press any key!

H A P P Y

B E T T E

A W A R E

Console > Filter Default levels

No Issues 2 hidden

Array() ScoreBoard.vue:49

Answer is: final Wordle.vue:135

Object Wordle.vue:301

[] ScoreBoard.vue:49

Answer is: Wordle V2.vue:74

broke

Answer is: Wordle V2.vue:74

below

ScoreBoard.vue:49

↓ {answer: 'below', Attempt\_1: 5, Attempt\_Number: 2}

Answer is: Wordle V2.vue:74

aware

Now on the scoreboard, the table will display the 2nd try **first** since it has a lower(better) attempts.

CSI 3140 Home    Portfolio    Wordle    Wordle\_V2    ScoreBoard

## Top 5 Scoreboard

Table

Attempt Number	Attempts
2	3
1	5

Clean Board

CSI 3140

localhost:5173/ScoreBoard

Chrome    File    Edit    View    History    Bookmarks    Profiles    Tab    Window    Help

99%    Thu Jul 11 4:55 PM

Console

No Issues | 2 hidden

Array(0)    ScoreBoard.vue:49

Answer is: final Wordle.vue:135

Object    Wordle.vue:301

[]    ScoreBoard.vue:49

Answer is: Wordle V2.vue:74

broke

Answer is: Wordle V2.vue:74

below

ScoreBoard.vue:49

{answer: 'below', Attempt\_1: 5, Attempt\_Number: 2}

Answer is: Wordle V2.vue:74

aware

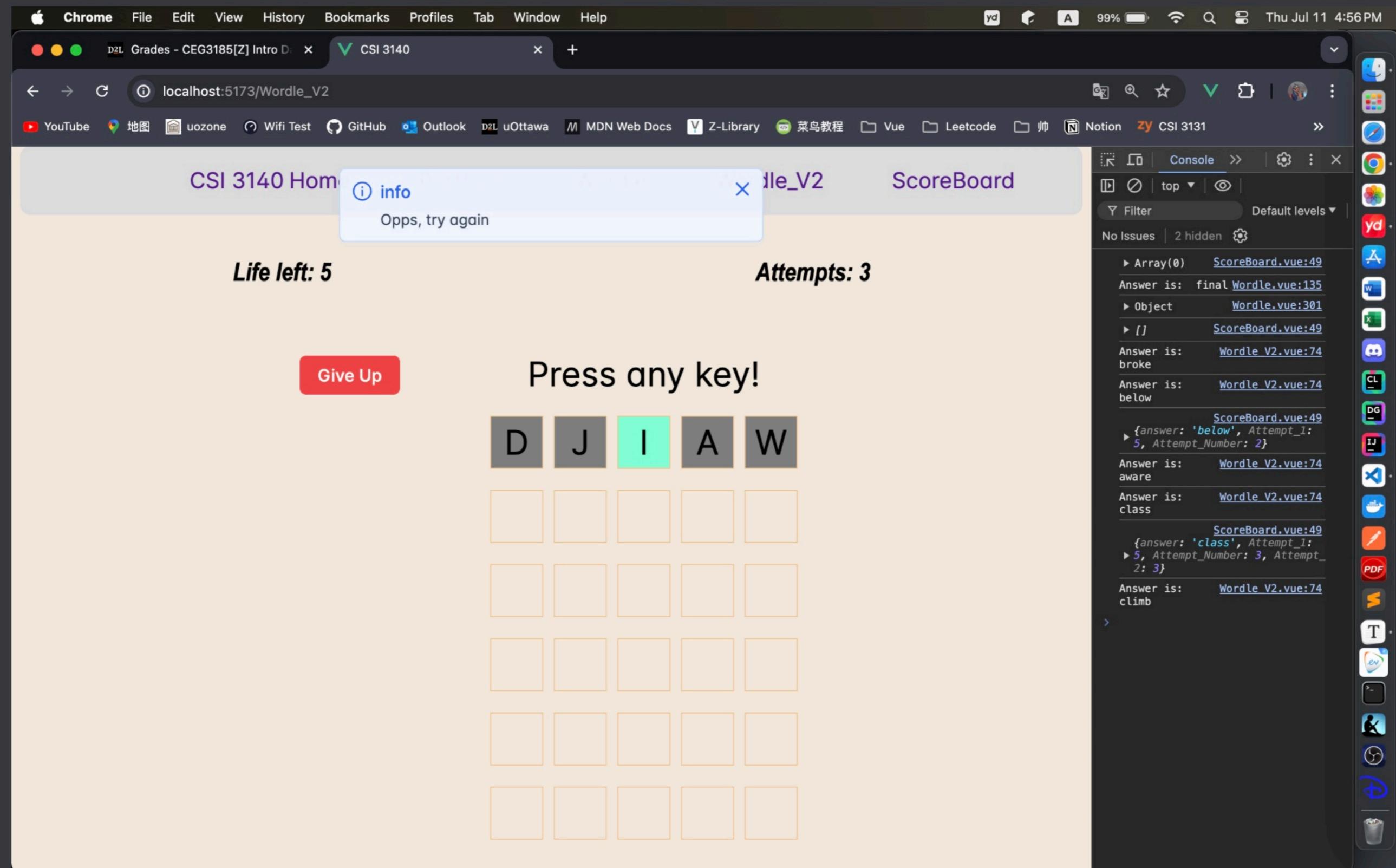
Answer is: Wordle V2.vue:74

class

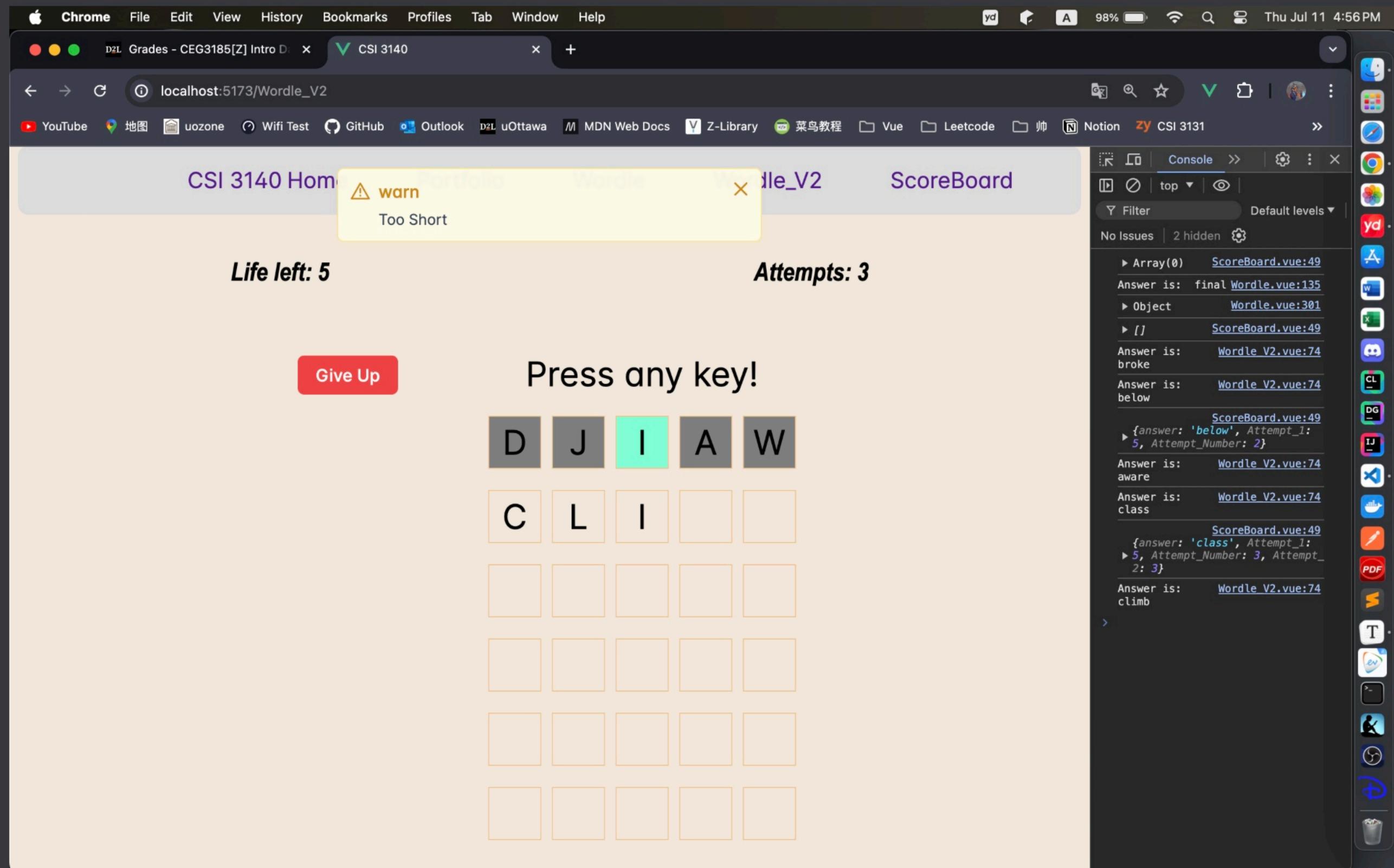
ScoreBoard.vue:49

{answer: 'class', Attempt\_1: 5, Attempt\_Number: 3, Attempt\_2: 3}

Of course, if you enter a wrong answer, it will prompt you:



Or not enough characters:



Chrome File Edit View History Bookmarks Profiles Tab Window Help

98% Thu Jul 11 4:57PM

Grades - CEG3185[Z] Intro D x CSI 3140

localhost:5173/Wordle\_V2

YouTube 地图 uozone Wifi Test GitHub Outlook uOttawa MDN Web Docs Z-Library 菜鸟教程 Vue Leetcode 帅 Notion CSI 3131

CSI 3140 Home Wordle\_V2 ScoreBoard

Info You lost!

Life left: 1

Attempts: 5

Give Up

Press any key!

D J I A J

D J A I W

D J A I D

D N K A N

N D A W I

D N A I W

Console

top ▾

Filter Default levels ▾

No Issues 2 hidden

▶ Array(0) ScoreBoard.vue:49

Answer is: final Wordle.vue:135

▶ Object Wordle.vue:301

▶ [] ScoreBoard.vue:49

Answer is: Wordle V2.vue:74

broke

Answer is: Wordle V2.vue:74

below

ScoreBoard.vue:49

▶ {answer: 'below', Attempt\_1: 5, Attempt\_Number: 2}

Answer is: Wordle V2.vue:74

aware

Answer is: Wordle V2.vue:74

class

ScoreBoard.vue:49

▶ {answer: 'class', Attempt\_1: 5, Attempt\_Number: 3, Attempt\_2: 3}

Answer is: Wordle V2.vue:74

climb

Answer is: Wordle V2.vue:74

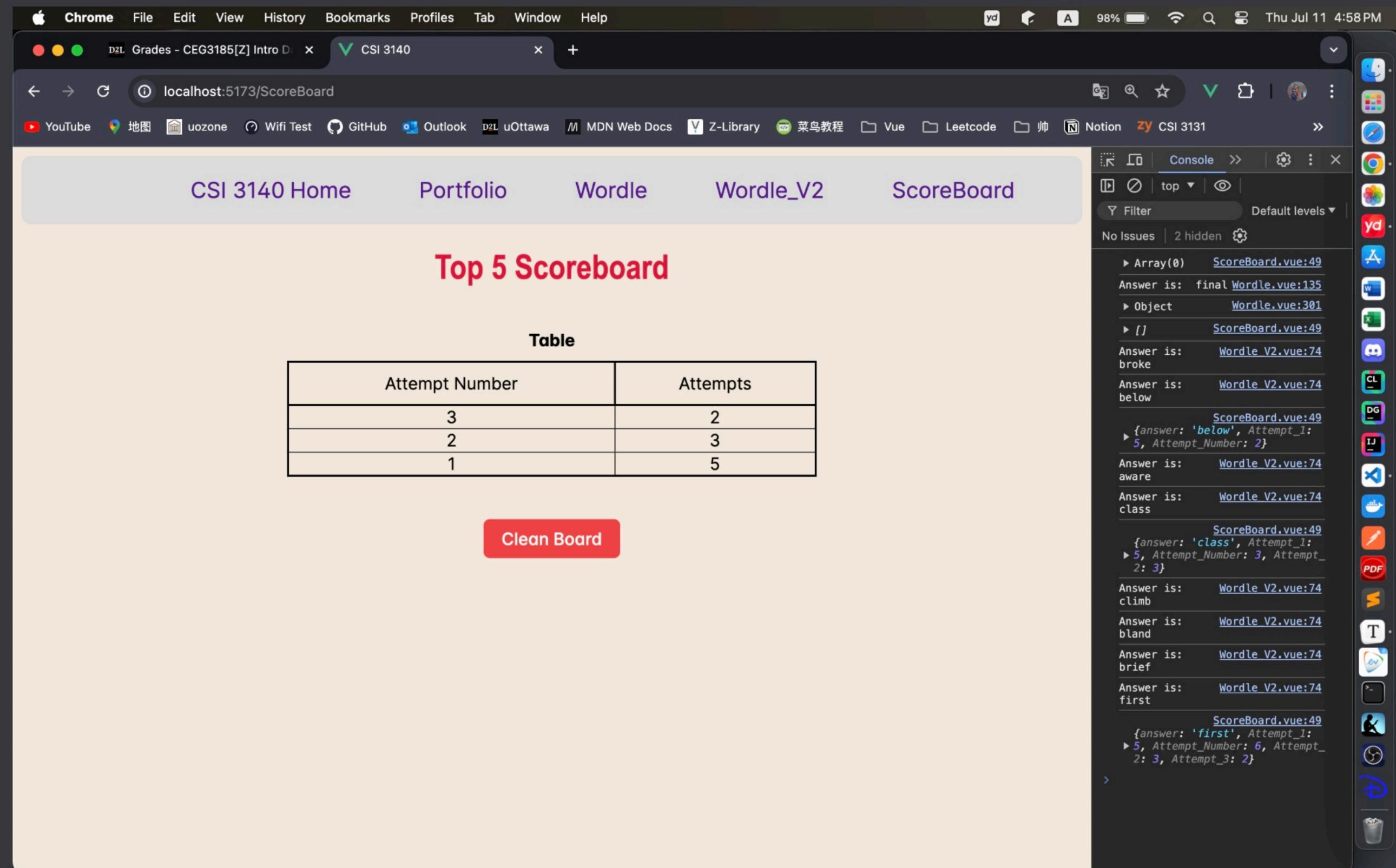
bland

Answer is: Wordle V2.vue:74

brief

...

As can be seen, there's no record for attempt number 5:



CSI 3140 Home    Portfolio    Wordle    Wordle\_V2    ScoreBoard

## Top 5 Scoreboard

Table

Attempt Number	Attempts
3	2
2	3
1	5

Clean Board

CSI 3140

localhost:5173/ScoreBoard

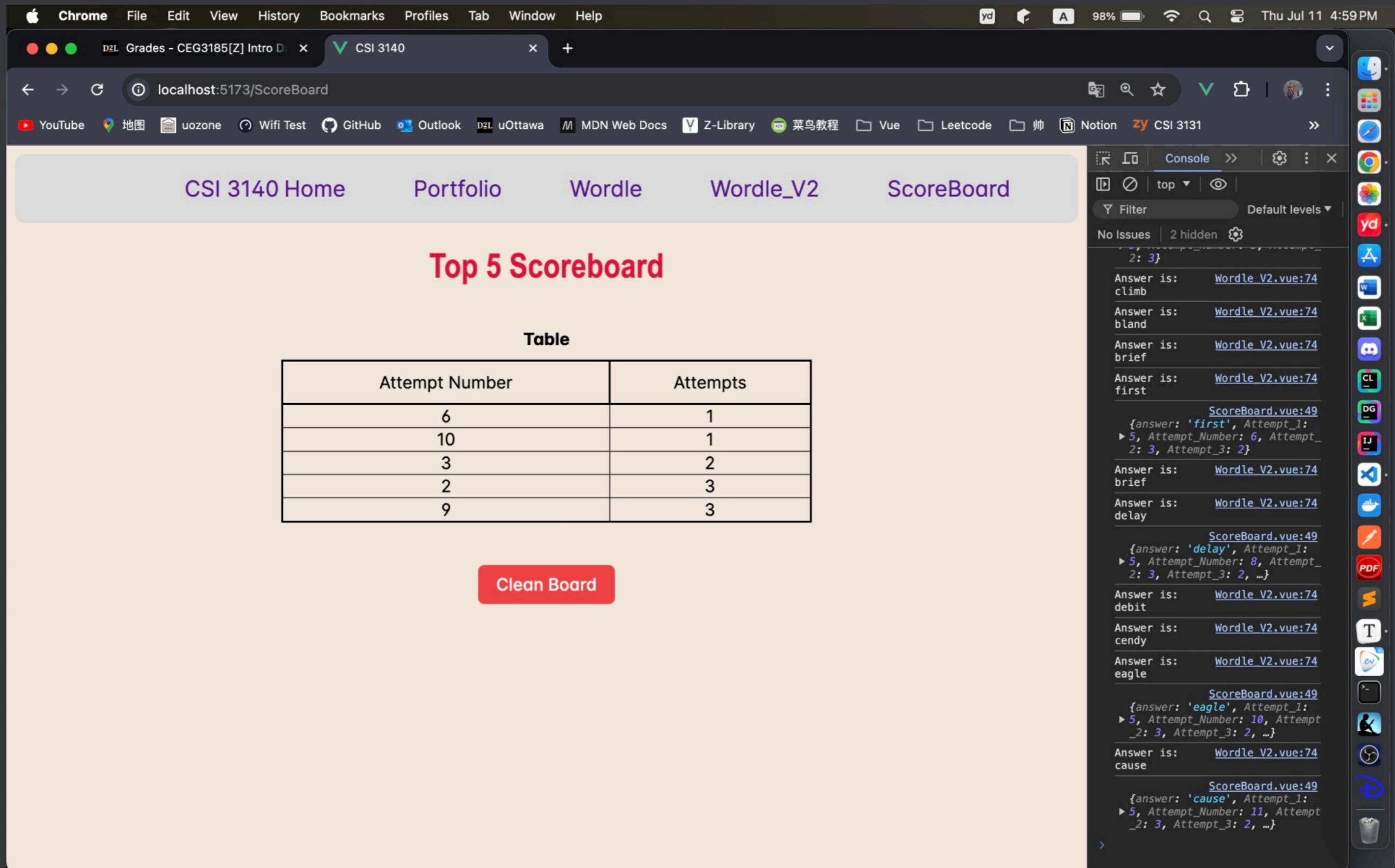
Console

No Issues | 2 hidden

- ▶ Array()    ScoreBoard.vue:49  
Answer is: final Wordle.vue:135
- ▶ Object    Wordle.vue:301
- ▶ []    ScoreBoard.vue:49  
Answer is: Wordle V2.vue:74  
broke
- ▶ Answer is: Wordle V2.vue:74  
below
- ▶ ScoreBoard.vue:49  
{answer: 'below', Attempt\_1: 5, Attempt\_Number: 2}
- ▶ Answer is: Wordle V2.vue:74  
aware
- ▶ Answer is: Wordle V2.vue:74  
class
- ▶ ScoreBoard.vue:49  
{answer: 'class', Attempt\_1: 5, Attempt\_Number: 3, Attempt\_2: 3}
- ▶ Answer is: Wordle V2.vue:74  
climb
- ▶ Answer is: Wordle V2.vue:74  
bland
- ▶ Answer is: Wordle V2.vue:74  
brief
- ▶ Answer is: Wordle V2.vue:74  
first
- ▶ ScoreBoard.vue:49  
{answer: 'first', Attempt\_1: 5, Attempt\_Number: 6, Attempt\_2: 3, Attempt\_3: 2}

# 3 Keep playing

Lets keep try a few time, and see the results.



The screenshot shows a Chrome browser window with a dark theme. The address bar shows the URL `localhost:5173/ScoreBoard`. The page content is a scoreboard application. At the top, there is a navigation bar with links: [CSI 3140 Home](#), [Portfolio](#), [Wordle](#), [Wordle\\_V2](#), and [ScoreBoard](#). The main section is titled **Top 5 Scoreboard** and contains a table with the following data:

Attempt Number	Attempts
6	1
10	1
3	2
2	3
9	3

Below the table is a red button labeled **Clean Board**.

On the right side of the browser window, the developer tools are open, specifically the **Console** tab. The console output shows a list of answers and their corresponding attempts:

```
2: 3}
Answer is: Wordle V2.vue:74
climb
Answer is: Wordle V2.vue:74
bland
Answer is: Wordle V2.vue:74
brief
Answer is: Wordle V2.vue:74
first
ScoreBoard.vue:49
{answer: 'first', Attempt_1:
▶ 5, Attempt_Number: 6, Attempt_
2: 3, Attempt_3: 2}
Answer is: Wordle V2.vue:74
brief
Answer is: Wordle V2.vue:74
delay
ScoreBoard.vue:49
{answer: 'delay', Attempt_1:
▶ 5, Attempt_Number: 8, Attempt_
2: 3, Attempt_3: 2, ...}
Answer is: Wordle V2.vue:74
debit
Answer is: Wordle V2.vue:74
cendy
Answer is: Wordle V2.vue:74
eagle
ScoreBoard.vue:49
{answer: 'eagle', Attempt_1:
▶ 5, Attempt_Number: 10, Attempt_
2: 3, Attempt_3: 2, ...}
Answer is: Wordle V2.vue:74
cause
ScoreBoard.vue:49
{answer: 'cause', Attempt_1:
▶ 5, Attempt_Number: 11, Attempt_
2: 3, Attempt_3: 2, ...}
```

As the table shows, no matter how many attempts we try, it will only store the top 5 scores!

## 4 Clean Board

If you want to challenge yourself again, you can press the clean board button

CSI 3140 Home

Info

Score Board has been reset

## Top 5 Scoreboard

Table

Attempt Number	Attempts
6	1
10	1
3	2
2	3
9	3

Clean Board

CSI 3140

localhost:5173/ScoreBoard

Console

```
2: 3}
Answer is: Wordle V2.vue:74
climb
Answer is: Wordle V2.vue:74
bland
Answer is: Wordle V2.vue:74
brief
Answer is: Wordle V2.vue:74
first
ScoreBoard.vue:49
{answer: 'first', Attempt_1:
▶ 5, Attempt_Number: 6, Attempt_
2: 3, Attempt_3: 2}
Answer is: Wordle V2.vue:74
brief
Answer is: Wordle V2.vue:74
delay
ScoreBoard.vue:49
{answer: 'delay', Attempt_1:
▶ 5, Attempt_Number: 8, Attempt_
2: 3, Attempt_3: 2, ...}
Answer is: Wordle V2.vue:74
debit
Answer is: Wordle V2.vue:74
candy
Answer is: Wordle V2.vue:74
eagle
ScoreBoard.vue:49
{answer: 'eagle', Attempt_1:
▶ 5, Attempt_Number: 10, Attempt_
2: 3, Attempt_3: 2, ...}
Answer is: Wordle V2.vue:74
cause
ScoreBoard.vue:49
{answer: 'cause', Attempt_1:
▶ 5, Attempt_Number: 11, Attempt_
2: 3, Attempt_3: 2, ...}
```

After a few seconds:

CSI 3140 Home      Portfolio      Wordle      Wordle\_V2      ScoreBoard

## Top 5 Scoreboard

Table

Attempt Number	Attempts
----------------	----------

Clean Board

Console >>      Default levels ▾

No Issues | 2 hidden

ScoreBoard.vue:49

The board is clean! Lets go back to the game:

