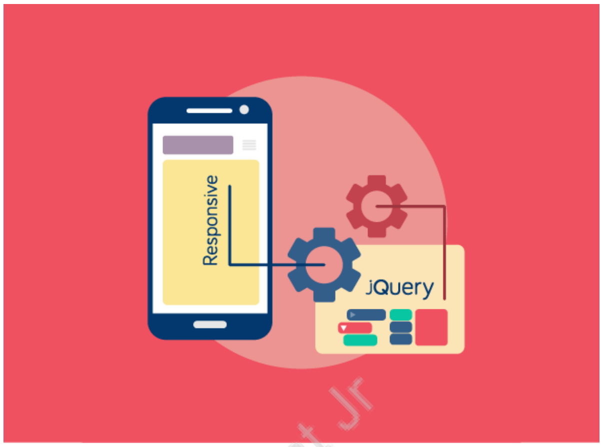
# PRO-C177

# jQuery AJAX METHOD



## What is our GOAL for this MODULE?

We learned about the jQuery ajax() method and used the jQuery ajax() method call to update the score result of the Mad Libs story.

## What did we ACHIEVE in the class TODAY?

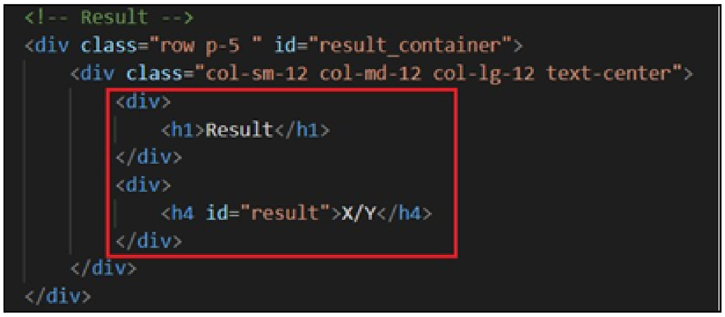
- Understood the use of AJAX techniques in web applications.
- Learned about the jQuery ajax() method.
- Showed the score result of the Mad Libs story using jQuery ajax() method call.

## Which CONCEPTS/CODING BLOCKS did we cover today?

- jQuery val(), text(),attr(),html(),append(),remove() methods.
- jQuery event listeners click and keyup.
- jQuery ajax() method.
- Flask Python's Framework to create APIs.

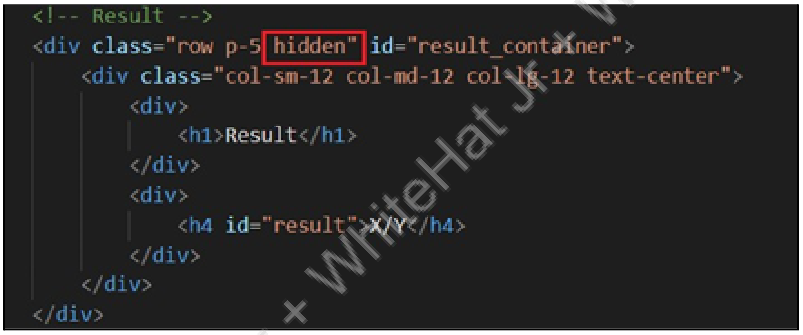
## How did we DO the activities?

1. Add the **<div>** tags after the Word Bank div in index.html to show the result.



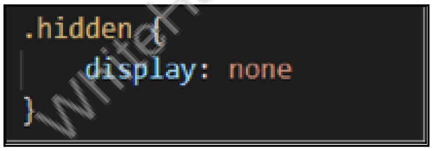
```
<!-- Result -->
<div class="row p-5 " id="result_container">
    <div class="col-sm-12 col-md-12 col-lg-12 text-center">
        <div>
            <h1>Result</h1>
        </div>
        <div>
            <h4 id="result">X/Y</h4>
        </div>
    </div>
</div>
```

2. Change the style to hidden for the **result div** until the submit button is clicked:
    - index.html



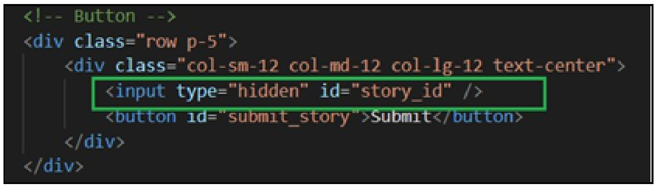
```
<!-- Result -->
<div class="row p-5 hidden" id="result_container">
    <div class="col-sm-12 col-md-12 col-lg-12 text-center">
        <div>
            <h1>Result</h1>
        </div>
        <div>
            <h4 id="result">X/Y</h4>
        </div>
    </div>
</div>
```

    - index.css



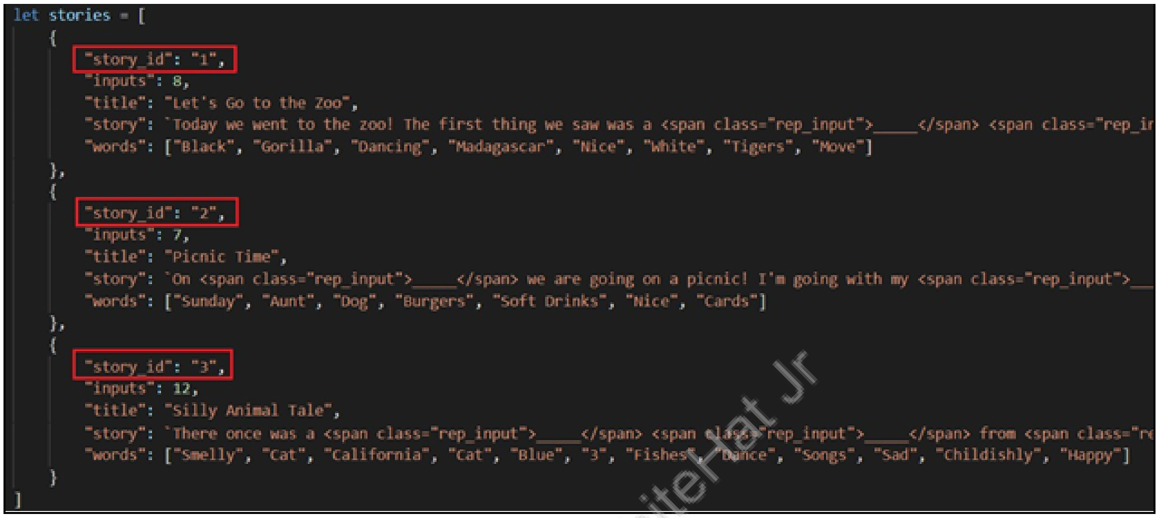
```
.hidden {
    display: none
}
```

3. To identify which story is submitted:
    - Add the **#story_id** <div> in the index.html file, which will be hidden initially.



```
<!-- Button -->
<div class="row p-5">
    <div class="col-sm-12 col-md-12 col-lg-12 text-center">
        <input type="hidden" id="story_id" />
        <button id="submit_story">Submit</button>
    </div>
</div>
```

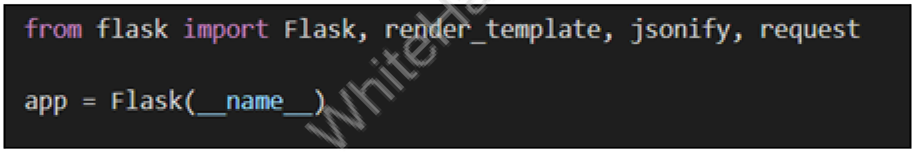
- Add the “**story_id**” in the stories array (in index.js).



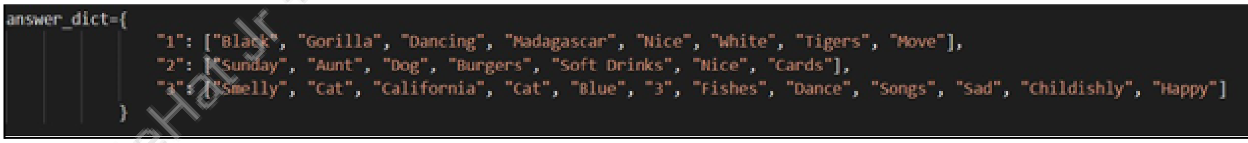
```
let stories = [
    {
        "story_id": "1",
        "inputs": 8,
        "title": "Let's Go to the Zoo",
        "story": `Today we went to the zoo! The first thing we saw was a <span class="rep_input">____</span> <span class="rep_ir
        "words": ["Black", "Gorilla", "Dancing", "Madagascar", "Nice", "White", "Tigers", "Move"]
    },
    {
        "story_id": "2",
        "inputs": 7,
        "title": "Picnic Time",
        "story": `On <span class="rep_input">____</span> we are going on a picnic! I'm going with my <span class="rep_input">__
        "words": ["Sunday", "Aunt", "Dog", "Burgers", "Soft Drinks", "Nice", "Cards"]
    },
    {
        "story_id": "3",
        "inputs": 12,
        "title": "Silly Animal Tale",
        "story": `There once was a <span class="rep_input">____</span> <span class="rep_input">____</span> from <span class="re
        "words": ["Smelly", "Cat", "California", "Cat", "Blue", "3", "Fishes", "Dance", "Songs", "Sad", "Childishly", "Happy"]
    }
]
```

4. Write a flask POST API to get the result:

    - Store answers in the variable **answer_dict** in app.py and also also initialize flask.

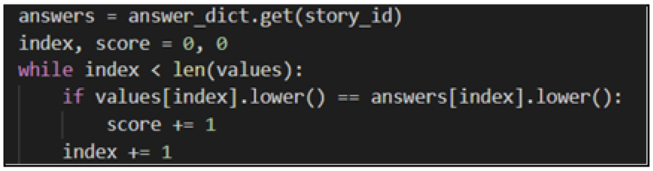


```
from flask import Flask, render_template, jsonify, request

app = Flask(__name__)
```

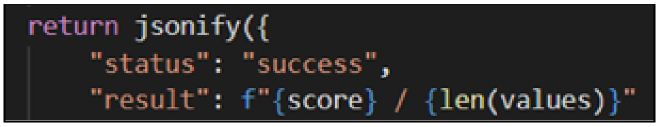


```
answer_dict={
            "1": ["Black", "Gorilla", "Dancing", "Madagascar", "Nice", "White", "Tigers", "Move"],
            "2": ["Sunday", "Aunt", "Dog", "Burgers", "Soft Drinks", "Nice", "Cards"],
            "3": ["Smelly", "Cat", "California", "Cat", "Blue", "3", "Fishes", "Dance", "Songs", "Sad", "Childishly", "Happy"]
        }
```

    - Add logic if the user's answer matches with the expected answer, we are increasing the score by 1 and return it to the user.

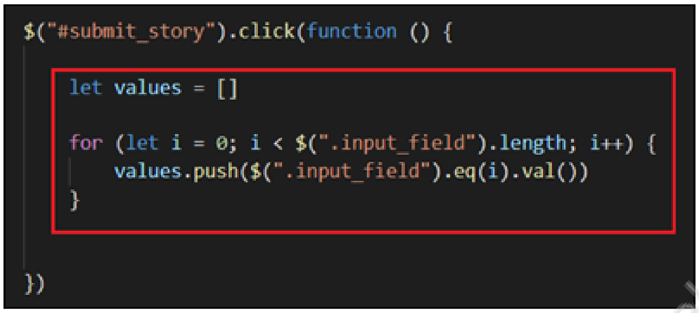


```
answers = answer_dict.get(story_id)
index, score = 0, 0
while index < len(values):
    if values[index].lower() == answers[index].lower():
        score += 1
    index += 1
```



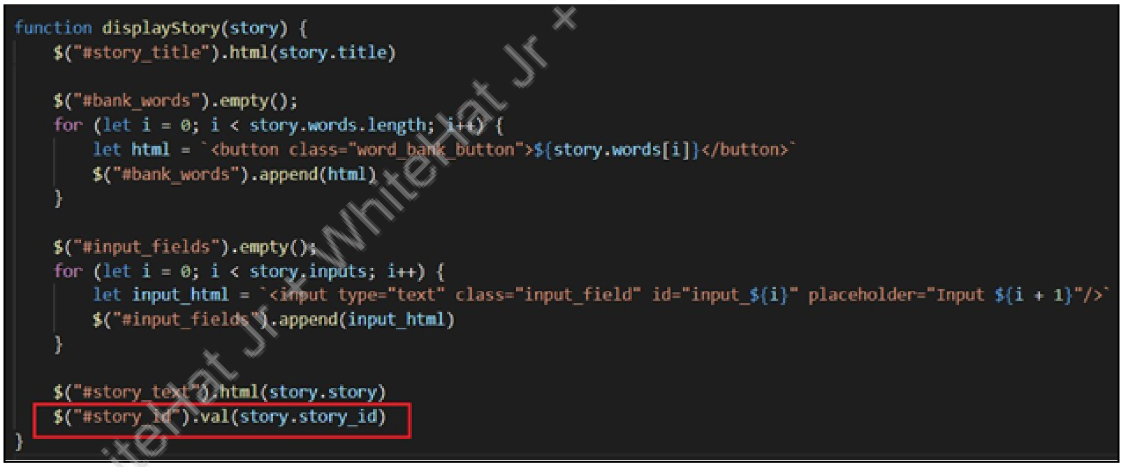
```
return jsonify({
    "status": "success",
    "result": f"{score} / {len(values)}"
```

5. To **Submit** the story:
    - Select the submit button element using #id.
    - Add a click event in the index.js file.
    - Loop through all the #input_filed filled by the user.
    - Push all the inputs to an array 'values'.



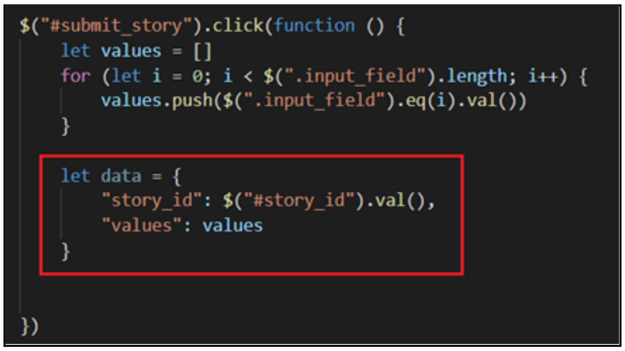
```
$("#submit_story").click(function () {

    let values = []

    for (let i = 0; i < $(".input_field").length; i++) {
        values.push($(".input_field").eq(i).val())
    }

})
```

    - Update **displayStory()** function with **#story_id** in the HTML.



```
function displayStory(story) {
    $("#story_title").html(story.title)

    $("#bank_words").empty();
    for (let i = 0; i < story.words.length; i++) {
        let html = `<button class="word_bank_button">${story.words[i]}</button>`
        $("#bank_words").append(html)
    }

    $("#input_fields").empty();
    for (let i = 0; i < story.inputs; i++) {
        let input_html = `<input type="text" class="input_field" id="input_${i}" placeholder="Input ${i + 1}"/>`
        $("#input_fields").append(input_html)
    }

    $("#story_text").html(story.story)
    $("#story_id").val(story.story_id)
}
```

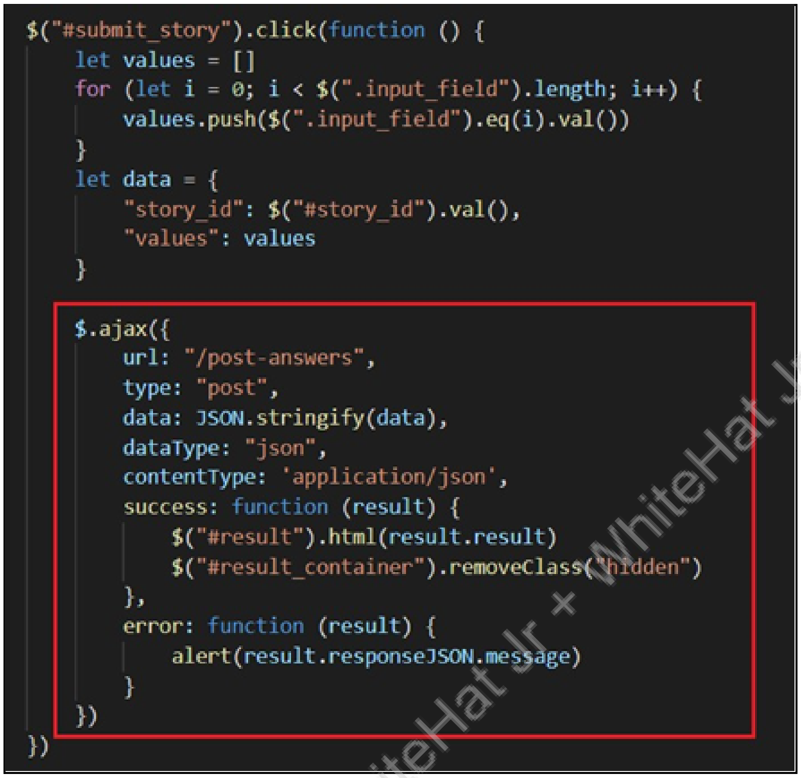
    - Store the **#story_id** and **values** of all the input_field in a JSON object variable, **data.**



```
$("#submit_story").click(function () {
    let values = []
    for (let i = 0; i < $(".input_field").length; i++) {
        values.push($(".input_field").eq(i).val())
    }

    let data = {
        "story_id": $("#story_id").val(),
        "values": values
    }

})
```

© 2020 The content of this email is confidential and intended for the recipient specified in the message only. It is strictly forbidden to share any part of this message with any third party without a written consent of the sender. If you received this message by mistake, please reply to this message and follow with its deletion, so that we can ensure such a mistake does not occur in the future.

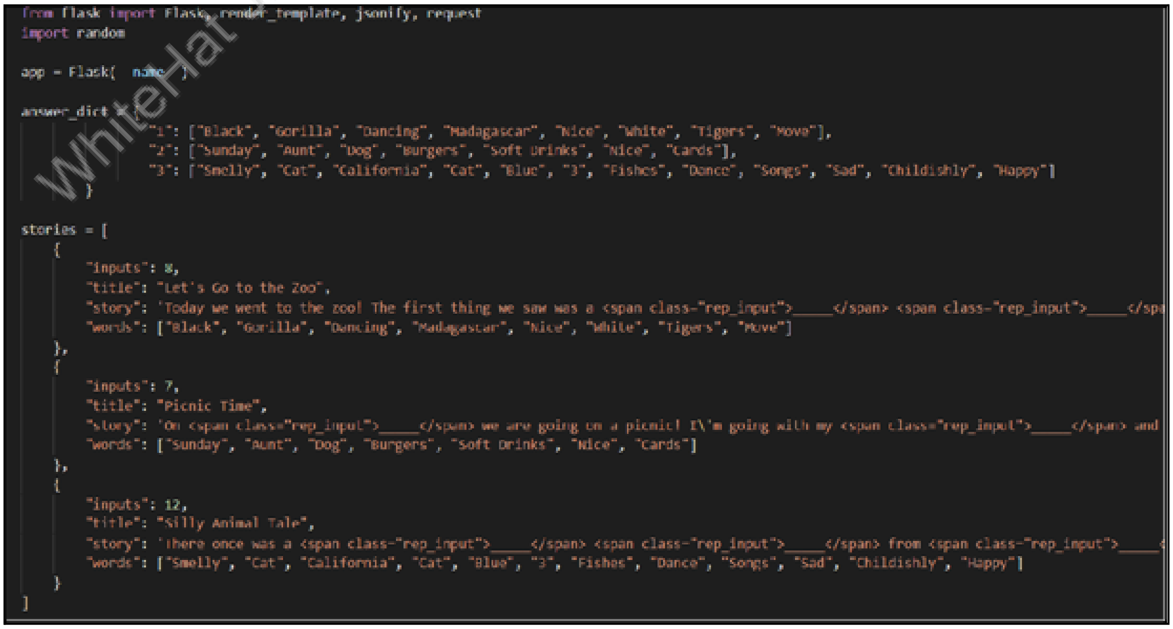
- Create an AJAX request with all the parameters.



```
$("#submit_story").click(function () {
    let values = []
    for (let i = 0; i < $(".input_field").length; i++) {
        values.push($(".input_field").eq(i).val())
    }
    let data = {
        "story_id": $("#story_id").val(),
        "values": values
    }

    $.ajax({
        url: "/post-answers",
        type: "post",
        data: JSON.stringify(data),
        dataType: "json",
        contentType: 'application/json',
        success: function (result) {
            $("#result").html(result.result)
            $("#result_container").removeClass("hidden")
        },
        error: function (result) {
            alert(result.responseJSON.message)
        }
    })
})
```

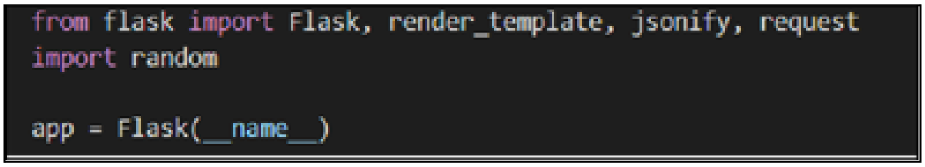
6. Write a flask GETAPI to get the stories:

- Move stories' variables from "index.js" to "app.py".

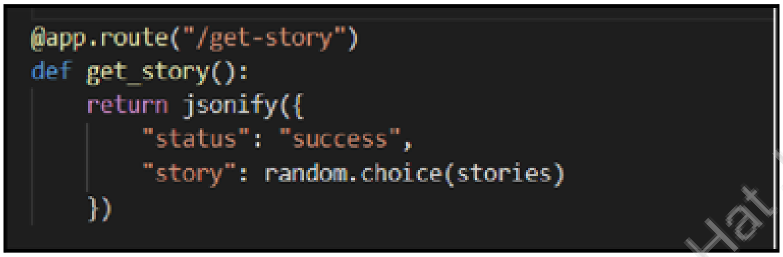


```
from flask import Flask, render_template, jsonify, request
import random

app = Flask(__name__)

answer_dict = {
            "1": ["Black", "Gorilla", "Dancing", "Madagascar", "Nice", "White", "Tigers", "Move"],
            "2": ["Sunday", "Aunt", "Dog", "Burgers", "Soft Drinks", "Nice", "Cards"],
            "3": ["Smelly", "Cat", "California", "Cat", "Blue", "3", "Fishes", "Dance", "Songs", "Sad", "Childishly", "Happy"]
    }

stories = [
    {
        "inputs": 8,
        "title": "Let's Go to the Zoo",
        "story": 'Today we went to the zoo! The first thing we saw was a <span class="rep_input">____</span> <span class="rep_input">____</spa
        "words": ["Black", "Gorilla", "Dancing", "Madagascar", "Nice", "White", "Tigers", "Move"]
    },
    {
        "inputs": 7,
        "title": "Picnic Time",
        "story": 'On <span class="rep_input">____</span> we are going on a picnic! I\'m going with my <span class="rep_input">____</span> and
        "words": ["Sunday", "Aunt", "Dog", "Burgers", "Soft Drinks", "Nice", "Cards"]
    },
    {
        "inputs": 12,
        "title": "Silly Animal Tale",
        "story": 'There once was a <span class="rep_input">____</span> <span class="rep_input">____</span> from <span class="rep_input">____
        "words": ["Smelly", "Cat", "California", "Cat", "Blue", "3", "Fishes", "Dance", "Songs", "Sad", "Childishly", "Happy"]
    }
]
```

- Import the random module to select a random story in “app.py”.

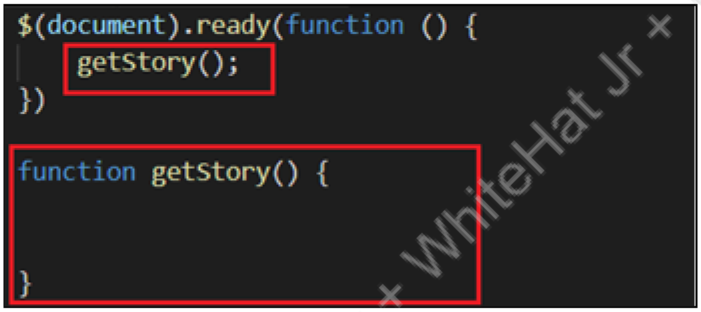


```
from flask import Flask, render_template, jsonify, request
import random

app = Flask(__name__)
```

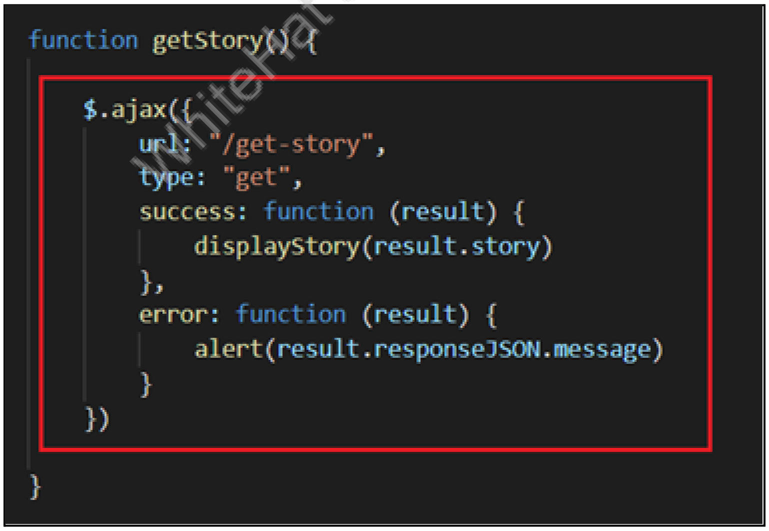


```
@app.route("/get-story")
def get_story():
    return jsonify({
        "status": "success",
        "story": random.choice(stories)
    })
```

7. Write a function getStory() to call the API using jQuery ajax() method.

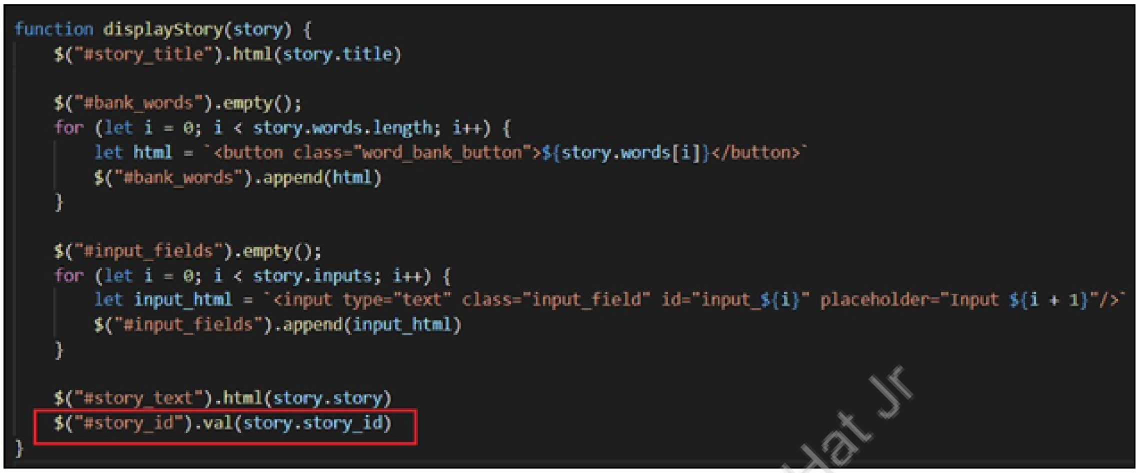


```
$(document).ready(function () {
    getStory();
})

function getStory() {


}
```



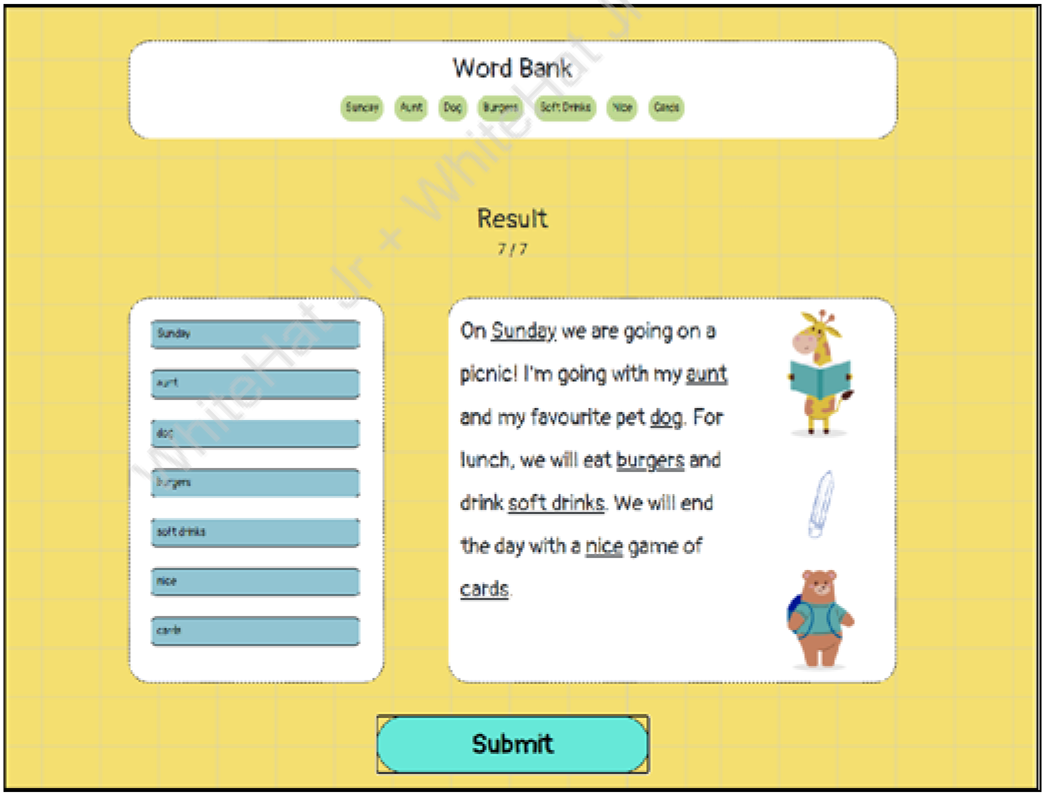
```
function getStory() {

    $.ajax({
        url: "/get-story",
        type: "get",
        success: function (result) {
            displayStory(result.story)
        },
        error: function (result) {
            alert(result.responseJSON.message)
        }
    })

}
```

8. Pass this story parameter in **displayStory()**.



```
function displayStory(story) {
    $("#story_title").html(story.title)

    $("#bank_words").empty();
    for (let i = 0; i < story.words.length; i++) {
        let html = `<button class="word_bank_button">${story.words[i]}</button>`
        $("#bank_words").append(html)
    }

    $("#input_fields").empty();
    for (let i = 0; i < story.inputs; i++) {
        let input_html = `<input type="text" class="input_field" id="input_${i}" placeholder="Input ${i + 1}"/>`
        $("#input_fields").append(input_html)
    }

    $("#story_text").html(story.story)
    $("#story_id").val(story.story_id)
}
```

9. Test the output by typing into different input fields and seeing how it is updating the blanks of the story in real time and displaying the result after clicking on submit.



Sunday
aunt
dog
burgers
soft drinks
nice
cards

On Sunday we are going on a picnic! I'm going with my aunt and my favourite pet dog. For lunch, we will eat burgers and drink soft drinks. We will end the day with a nice game of cards.

We learned to create a simple application to learn about the basics of jQuery and used jQuery ajax() method to get the result also to get a random story each time.

### What's NEXT?

In the next class, we will learn about rendering maps in a web application.

### EXTEND YOUR KNOWLEDGE:

- You can refer to the link below to explore more about AFrame
  [A-Frame]
- You can refer to the link below to explore more about jQuery
  [jQuery]