

## Report Lab n.7

### Introduction to jQuery:

jQuery is a fast, lightweight, and cross-platform JavaScript library that simplifies the process of traversing HTML documents, handling events, creating animations, and executing Ajax requests.

jQuery works by selecting elements from the DOM (Document Object Model) using CSS-like selectors and then manipulating those elements using a variety of built-in methods. It also provides a range of utility functions for common tasks such as string manipulation, object iteration, and array manipulation.

One of the key advantages of jQuery is its cross-browser compatibility. It automatically detects and handles browser-specific quirks and inconsistencies, making it easier to write code that works consistently across all major browsers.

### Task:

Please use the following jQuery features to make your main website clearer when it comes to linking to content.

1. Selector manipulation.
2. Event handling (on mouse over/out, focus/blur etc.) - for example (a description of what you get when clicking the link)
3. Hide/Fade some text/menu options elements
4. Slide effect,
5. Animation effects

In order to start programming with jQuery library it is necessary to import it:

```
<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
```

After importing I changed the ajax functions that connect all the project page to the main one using jQuery functionality, and I saw that it is so much easier and shorter to do it with jQuery:

```
ajaxconnect:function ajaxConnect(setting) {  
    $.ajax({  
        url: setting.contentsource[1],  
        cache: !this.bustajaxcache,  
        success: function(response) {  
            $('#'+ setting.id).html(response);  
            featuredcontentslider.buildpaginate(setting);  
        }  
    })  
}
```

```

    });
  },

  ajaxpopulate:function ajaxPopulate(page_request, setting) {
    if (page_request.readyState == 4 && (page_request.status==200 ||
window.location.href.indexOf("http")==-1)) {
      $('#'+ setting.id).html(page_request.responseText);
      featuredcontentslider.buildpaginate(setting);
    }
  }
}

```

The ajaxConnect function takes a setting object as a parameter that specifies the URL of the content source and other options such as whether to cache the response. It uses the \$.ajax function from the jQuery library to make an AJAX request to the specified URL. If the request is successful, the response is inserted into the HTML element with the ID specified in the setting object, and the buildpaginate function is called to build pagination links for the content slider. The ajaxPopulate function is called when the AJAX request initiated by ajaxConnect is completed. It checks if the response status is 200 (OK) and inserts the response into the HTML element with the ID specified in the setting object. It then calls the buildpaginate function to build pagination links for the content slider.

After this I played with jQuery functionality:

```

$(document).ready(function(){
  $("#1").hover(function(event){
    $("#description1").fadeIn();
  }, function(){
    $("#description1").fadeOut();
  });
  $("#2").hover(function(event){
    $("#description2").fadeIn();
  }, function(){
    $("#description2").fadeOut();
  });
  $("#3").hover(function(event){
    $("#description3").fadeIn();
  }, function(){
    $("#description3").fadeOut();
  });
  $("#4").hover(function(event){
    $("#description4").fadeIn();
  }, function(){
    $("#description4").fadeOut();
  });
});

```

```

}, function(){
    $("#description4").fadeOut();
});
});

```

This code is using jQuery to create hover effects on four div elements previously created with description of each project inside and IDs #1, #2, #3, and #4.

When the mouse enters one of these elements, the fadeIn() method is called on the corresponding element with ID #description1, #description2, #description3, or #description4. This causes the description element to fade in gradually, becoming more and more visible over time.

When the mouse leaves the element, the fadeOut() method is called on the corresponding description element. This causes the description element to fade out gradually, becoming less and less visible over time.

Result(mouse is over Project n.1)

Advanced internet programming projects

HTML table



PROJECT N.1 VINCENZO\_MURRO

First name	First name: <input type="text"/>
Last name	Last name: <input type="text"/>
Age	17 ▾
Sex	<input type="radio"/> Male <input type="radio"/> Female
Hobby	<input type="checkbox"/> Soccer <input type="checkbox"/> Basket <input type="checkbox"/> Hockey <input type="checkbox"/> Running <input type="checkbox"/> Others
OK	Cancel

Project n.1

Project n.2

Project n.3

Project n.4

<

>

```
$(document).ready(function(){
  $("#1").click(function(event){
    $("#description1").toggleClass("myClass"),
    setTimeout(()=>$("#description1").toggleClass("backupdescription"), 1000));
  $("#2").click(function(event){
    $("#description2").toggleClass("myClass"),
    setTimeout(()=>$("#description2").toggleClass("backupdescription"), 1000));
  $("#3").click(function(event){
    $("#description3").toggleClass("myClass"),
    setTimeout(()=>$("#description3").toggleClass("backupdescription"), 1000));
  $("#4").click(function(event){
    $("#description4").toggleClass("myClass"),
    setTimeout(()=>$("#description4").toggleClass("backupdescription"), 1000));
});
```

## CSS:

```
.myClass {
  background-color: yellow;
  color: red;
  font-size: 24px;
  padding: 20px;
  transition: all 0.7s ease-out;
}
.backupdescription{
  background-color: #fff;
  transition: all 0.7s ease-out;
  padding: 10px;
  font-size: 14px;
  color: #000;
}
```

Each element with an ID of 1 through 4 is selected using the jQuery `$( )` function, and a `click()` event is attached to each element. When the element is clicked, a function is executed.

Within each click event function, the `toggleClass()` method is used to add or remove the class `myClass` from the corresponding description div.

After toggling the `myClass` class, the `setTimeout()` method is used to wait for 1 second before toggling the `backupdescription` class. This class is used to restore the original CSS properties of the description div.

The `setTimeout()` method takes two arguments: a function to execute after the specified time has elapsed, and the time in milliseconds to wait before executing the function.

