**EXPERIMENT 4**

**AIM:** Design Web Pages Using The JQuery Library.

**THEORY:**

**JQuery Selectors:**

Query selectors allow you to select and manipulate HTML element(s).All selectors in jQuery start with the dollar sign and parentheses: **$()**.

The various types of selectors include:

1)element selector

2)id selector

3)class selector

**JQuery FadeIn:**

With jQuery you can fade an element in and out of visibility.

jQuery has the following fade methods:

fadeIn()

fadeOut()

fadeToggle()

fadeTo()

**Getters and Setters in JQuery:**

Three simple, but useful, jQuery methods for DOM manipulation are:

* **text() -** Sets or returns the text content of selected elements
* **html() -** Sets or returns the content of selected elements (including HTML markup)
* **val() -** Sets or returns the value of form fields

These functions allow us to **get** the value of the given element. However, if we wish to set the value of the given element, we can use the **same** functions with new value in quotes, inside the parenthesis. This allows us to **set** the value the elements

**JQuery Events:**

Event methods trigger or attach a function to an event handler for the selected elements.

The following are some of the jQuery methods used to handle events:

1. **$(document).ready() -**

The **$(document).ready()** method allows us to execute a function when the document is fully loaded. This event is already explained in the jQuery Syntax chapter.

1. **click() -**

The **click()** method attaches an event handler function to an HTML element. The function is executed when the user clicks on the HTML element.

**Example -**

$( "p" ).click(function() {

console.log( "<p> was clicked" );

});

1. **on() -**

The **on()** method attaches an event handler function to an HTML element, similar to the click() function but with much **more flexibility**. The function is executed depending on what we enter into the first parameter of the function. If the function has “click” as first parameter, then the function executes on the click of the element, much like the click() function.

**Example -**

$( "p" ).on( "click", function() {

console.log( "<p> was clicked" );

});

1. **focus() -**

The **focus()** event occurs when an element gets focus (when selected by a mouse click or by "tab-navigating" to it).

The **focus()** method triggers the focus event, or attaches a function to run when a focus event occurs.

**Example -**

$("input").focus(function(){

$("span").css("display", "inline").fadeOut(2000);

});

**JQuery FadeIn:**

jQuery offers several utility methods in the $ namespace. These methods are helpful for accomplishing routine programming tasks.

Some of these utility methods are given below:

1. **$.each() -**

The method **.each()** can be called on a selection to iterate over the elements contained in the selection. **.each()**, not $.each(), should be used for iterating over elements in a selection.

**Example -**

$.each({ ‘A’,’B’,’C’ }, function( k, v ) {

console.log( k + " : " + v );

});

1. **$.trim() -**

The method **.trim()** removes leading and trailing whitespace.

**Example -**

$.trim( " lots of extra whitespace " );

1. **$.inArray() -**

The method **.inArray()** returns a value's index in an array, or -1 if the value is not in the array.

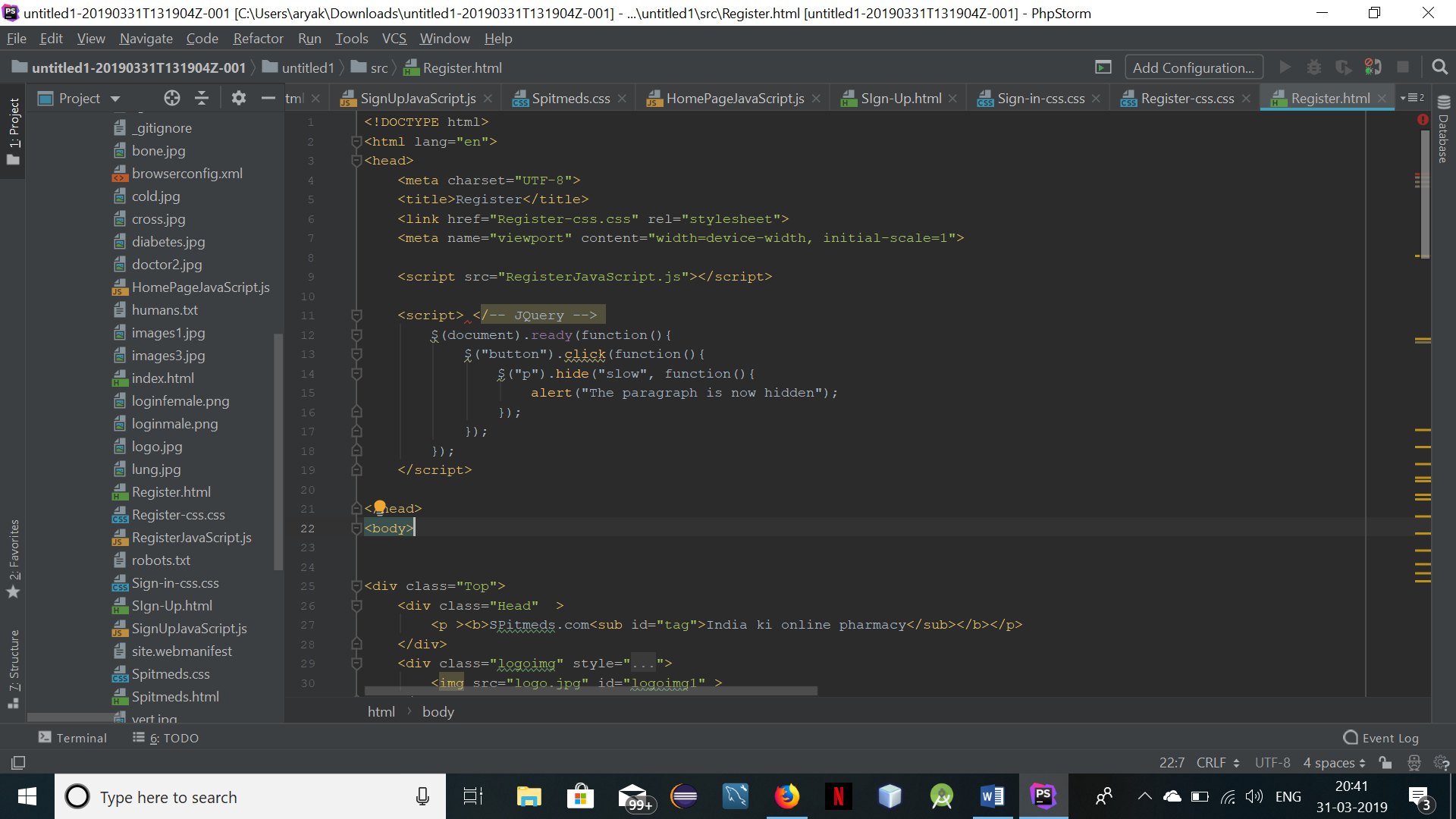
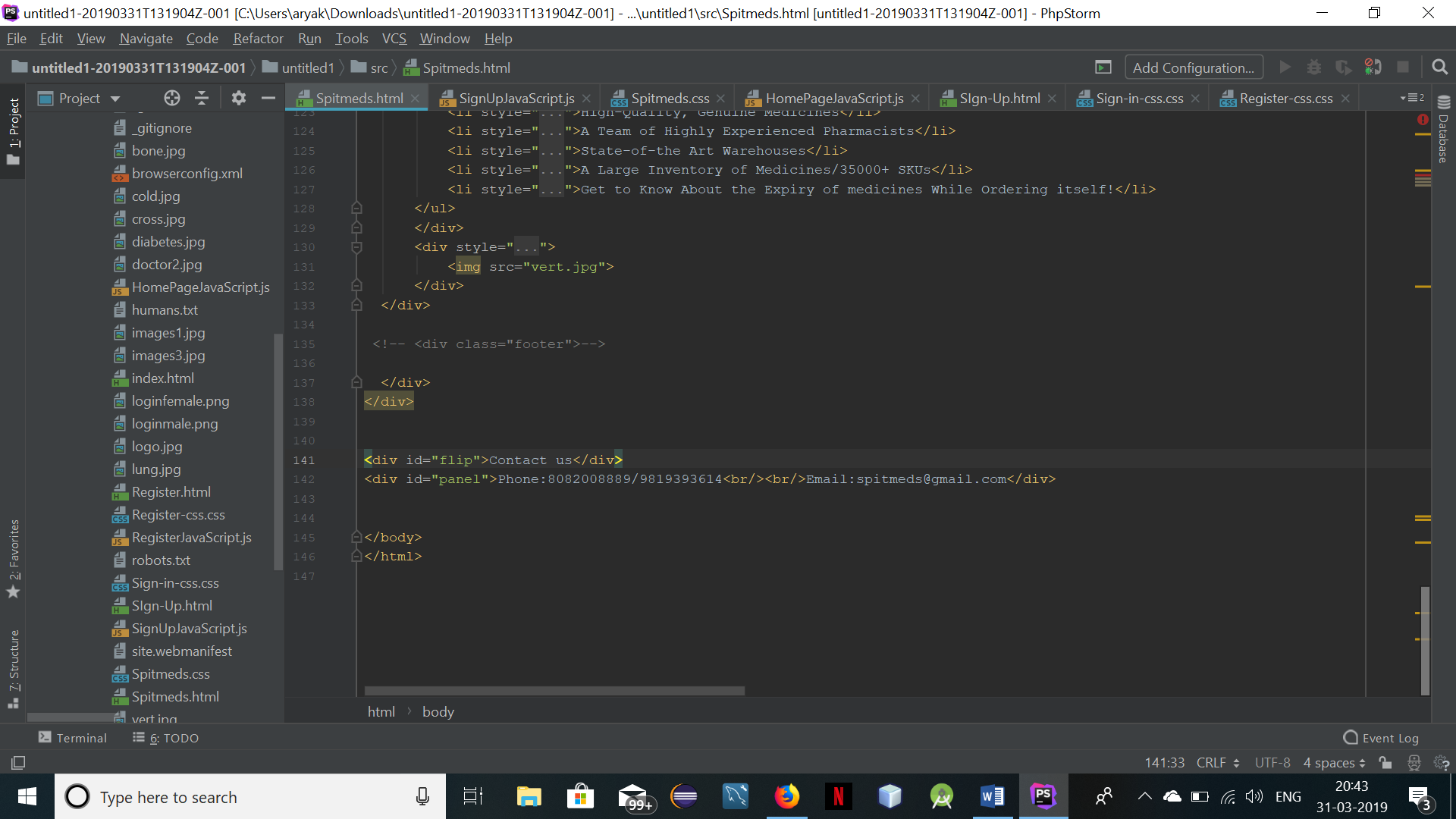
**Example -**

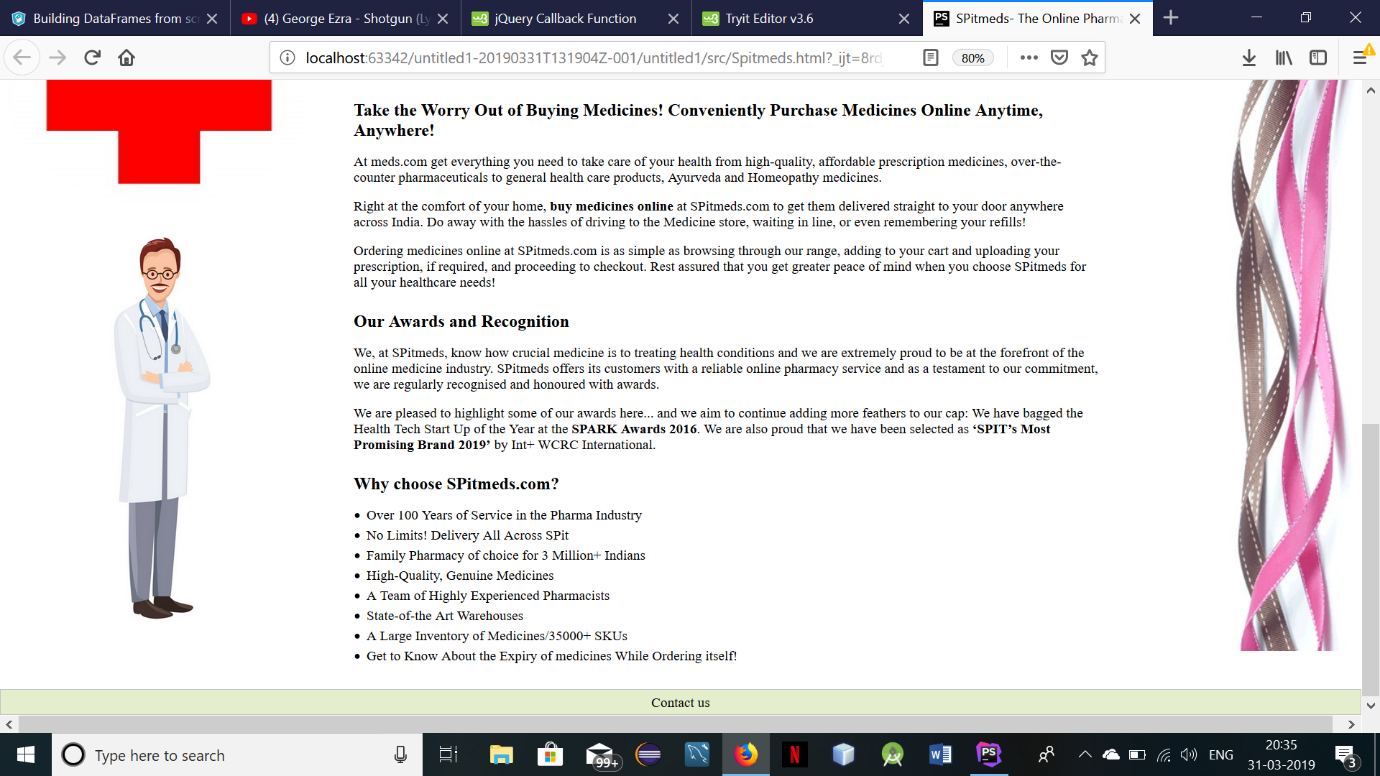
var myArray = [ 1, 2, 3, 5 ];

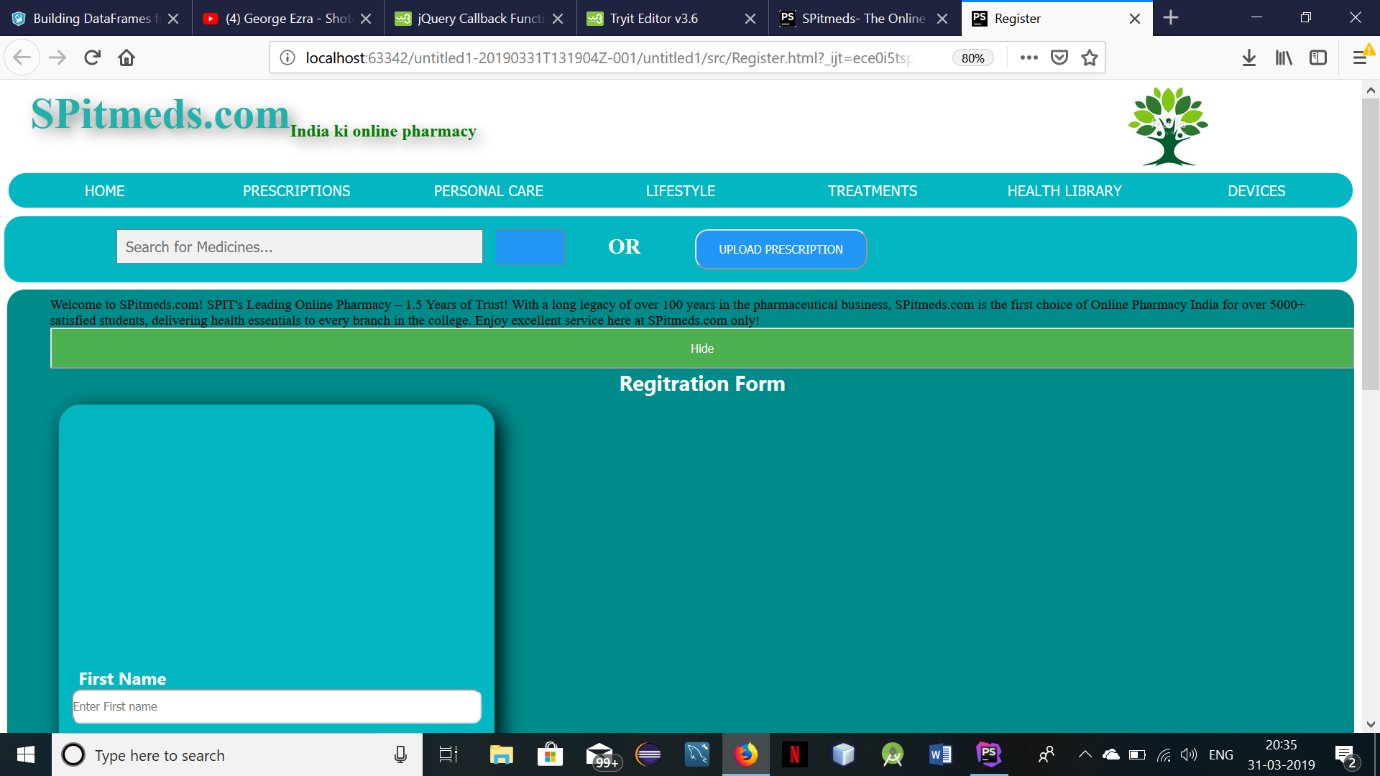
if ( $.inArray( 4, myArray ) !== -1 ) {

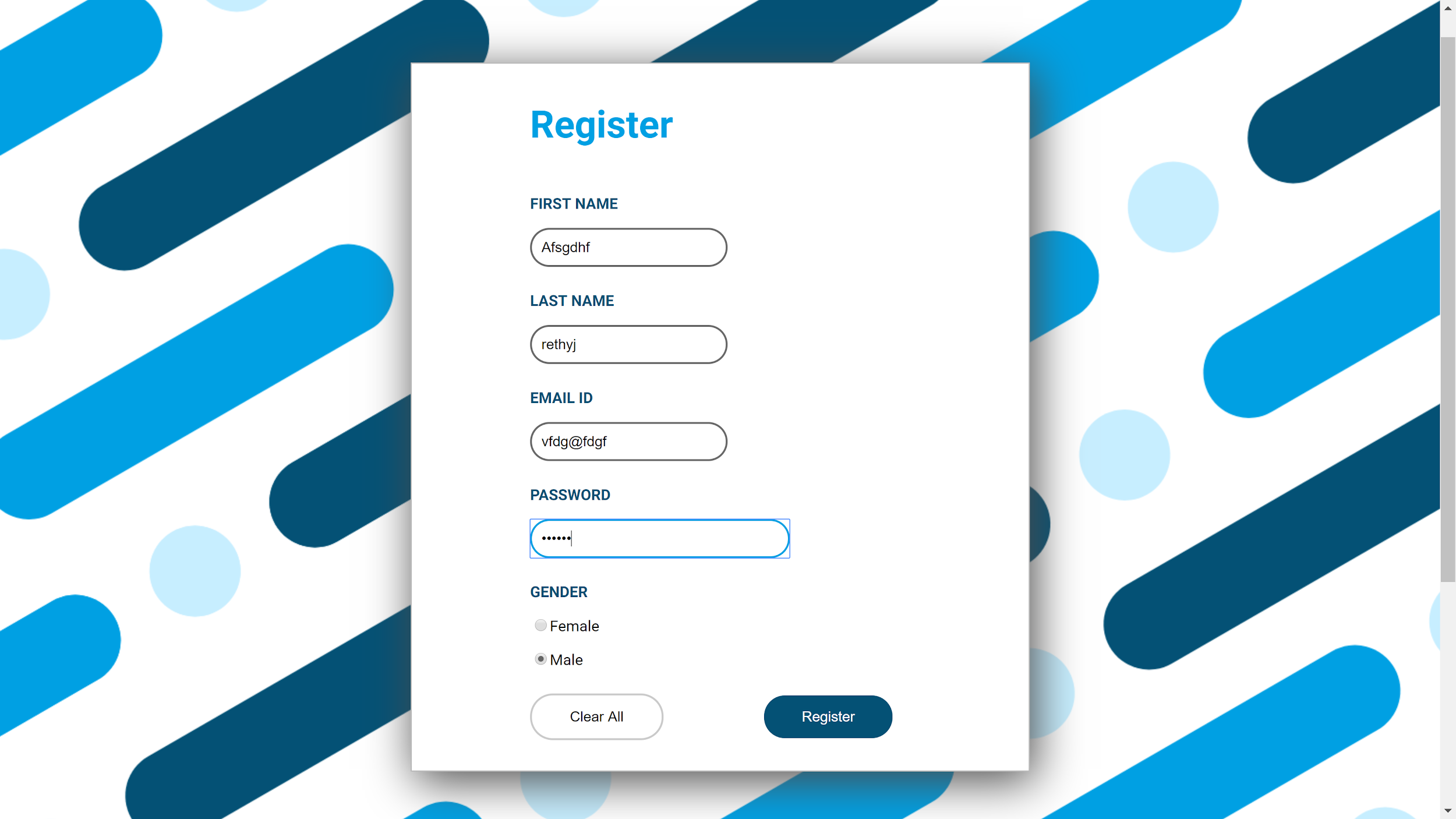
console.log( "found it!" );

}







****

**CONCLUSION:** From this experiment,we have implemented jquery in our web pages and thus we are able to add animations, like fade in and card flip, and added event handling to it.