jQuery

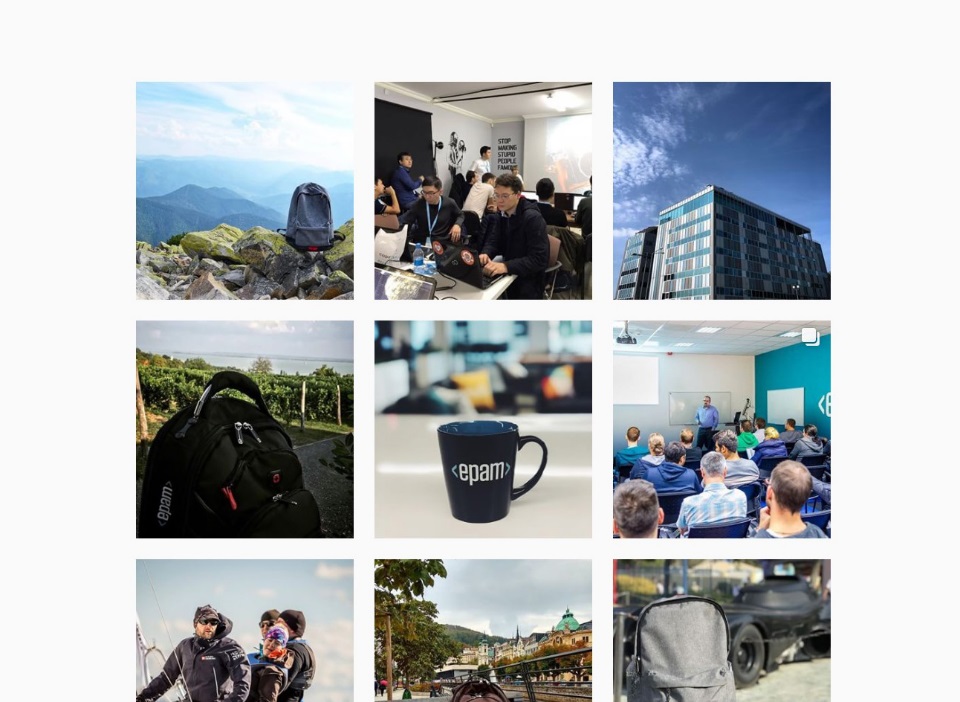
**DEADLINE:** 16/10/2018

## FOLDER STRUCTURE

|  |  |
| --- | --- |
| FL\_9\_7\_homework\_jquery/\*     homework/\*         src/\*             index.html\*  js/\*              jquery-3.3.1.min.js \*              app.js\*  css/\*              style.css\*  data/\*              media.json\*  img/\*              icons.png\*  picture/\*              \*.jpg\* | \*  – required |
|  |  |

## TASK

Create Instagram’s image gallery (simplified version – pic. design1).



Pic.1 - design1

**Requirements**:

* On 1st page load shows 12 items and button ‘View More’;
* After click ‘View More’ add next 6 items;
* Item click: open pop-up with content: info (image, author, location, description, ‘likes’ count, and mocked text ‘Add comment…’) and controls – arrows and close button(cross);
* Arrows click: show next/prev element;
* Cross click (or ‘Esc’): close pop-up;

## RESTRICTIONS

* jQuery source files shouldn't be touched at all;
* no need to use any of build systems(grunt, gulp, webpack).

## BEFORE SUBMIT

* test application;
* check files structure;
* check code structure(code style);

## SUBMIT

* The folder should be uploaded to github repository ‘**fl-9**’ into **master** branch

/\*

let earthLoader = document.getElementById('land');

let waterLoader = document.getElementById('water');

let simpleLoader = document.getElementById('simple');

let result = document.getElementById('result');

let trackBtn = document.getElementById('trackBtn');

trackBtn.addEventListener('click', findCoors);

let http = {

get: function (url) {

return new Promise(function (resolve, reject) {

let xhr = new XMLHttpRequest();

xhr.open('GET', url);

xhr.onload = function () {

if (this.status >= 200 && this.status < 300) {

let response = JSON.parse(xhr.responseText);

resolve(response);

} else {

reject({

status: this.status,

statusText: xhr.statusText

});

}

};

xhr.onerror = function () {

reject({

status: this.status,

statusText: xhr.statusText

});

};

xhr.send();

});

}

};

function hideElement (elem) {

elem.style.display = 'none';

}

function showElement (elem) {

elem.style.display = 'block';

}

function findCoors () {

hideElement(result);

hideElement(earthLoader);

hideElement(waterLoader);

showElement(simpleLoader);

let lat = document.getElementById('latitude').value;

let lon = document.getElementById('longtitude').value;

let url = `https://api.onwater.io/api/v1/results/${lat},${lon}`;

http.get(url)

.then(res => {

let text = document.getElementById('result');

text.innerHTML = res.water ? 'Water' : 'Land';

hideElement(simpleLoader);

showElement(text);

if (res.water) {

hideElement(earthLoader);

showElement(waterLoader);

} else {

hideElement(waterLoader);

showElement(earthLoader);

}

})

.catch(err => {

console.log(err);

}

);

}

\*/