**Repository**:

Index.html  
style.css  
data.js  
functions.js  
app.js – runs all the functions/ events (eventListener)

function attachCards(); is set to run as soon as the page loads. (*app.js*)

attachCards(evenimente);

function createCard(evenim){

    let card=document.createElement("div");

    card.classList.add("card");

    card.classList.add(`card-${evenim.id}`)

    card.innerHTML=`

                        <h3>${evenim.titlu}</h3>

                        <p>${evenim.data}</p>

                        <p>${evenim.descriere}Lorem ipsum dolor sit amet consectetur.</p>

                        <div class="actions-container">

                            <i class="favorite fa-solid fa-star"></i>

                            <i class="trash fa-solid fa-trash"></i>

                            <p class="edit">EDIT</p>

                            <p class="save hide">SAVE</p>

                        </div>

                  `;

    return card;

}

function attachCards(arr){

    let cardContainer = document.querySelector(".card-container");

    cardContainer.innerHTML="";

    arr.map(e=>createCard(e)).forEach(element => {

        cardContainer.append(element);

    });

}

**arr**.**map(e** => **createCard(e))**.**forEach(element** => **{**

**cardContainer**.**append(element)**;

**})**;

**// map()** method transforms each **element** in the array and returns a **new array** with the transformed elements.

* .map(...) goes through each element in arr.
* .map() transforms each event into a card.
* e => createCard(e) is an arrow function.

arr.map(e => createCard(e))

is equivalent to:

arr.map(function(e)){  
 return createCard(e);  
 });

A screenshot of a computer program

AI-generated content may be incorrect.

**// .forEach(e => { cardContainer.append(e); });**

****

is functionally equivalent to:  
A black screen with white text

AI-generated content may be incorrect.

- **forEach** is a method that takes a **callback function** and runs it once for each element in the array.

- the parameter e (or any name you choose) represents the **current element** in the array during each iteration.

- the body of the function tells JavaScript what to do with that element — in this case, append it to **cardContainer**.

A screenshot of a computer program

AI-generated content may be incorrect.

**// filter()**

**.filter()** method expects a *callback function* that returns a **Boolean value** (true or false) – for each element in the array.

Boolean decides:

* true --> keep the element in the new array.
* false --> exclude the element from the new array.

let evenimente = [

  { id: 1, titlu: "Event A" },

  { id: 2, titlu: "Event B" },

  { id: 3, titlu: "Event C" }

];

A screenshot of a computer

AI-generated content may be incorrect.