

PROIECT

CUPRINS

1	Proiect JavaScript.....	2
1.1	Obiective.....	2
1.2	Structura de fisiere si directoare.....	2
1.3	Modificarea dinamica a paginii.....	2
1.4	Sortarea produselor.....	2
1.5	Adaugarea produselor in cosul de cumparaturi.....	3
1.6	Stergerea elementelor din cosul de cumparaturi.....	3
1.7	Actualizarea totalurilor.....	3
1.8	Evaluare.....	3
1.9	Trimiterea arhivei.....	3

1 PROIECT JAVASCRIPT

1.1 Obiective

- Modificarea dinamica a paginii
- Sortarea datelor in functie de un criteriu selectat in pagina
- Completarea unui cos de cumparaturi
- Calcularea totalului produselor din cosul de cumparaturi

1.2 Structura de fisiere si directoare

Arhiva contine urmatoarele fisiere si directoare:

index.html	pagina care trebuie modificata dinamic din JavaScript
css	directorul care contine stilurile de afisare
img	directorul cu imaginile produselor
src	directorul care contine fisierele JavaScript
src/data.js	fisierul cu datele initiale

1.3 Modificarea dinamica a paginii

Obiectul `data`, declarat in fisierul `data.js`, contine descrierea produselor ce trebuie completate in pagina. Pentru aceasta se va folosi proprietatea `items`. Continutul fisierului `data.js` nu trebuie modificat.

```
const data = {
  items: [
    {
      name: "Supa",
      price: 10.45,
      image: "img/supa.jpg"
    },
    ...
    {
      name: "Tort de fructe",
      price: 32.55,
      image: "img/tort.jpg"
    }
  ]
};
```

Pentru fiecare element din `items` se va clona continutul elementului `template` care are `id` cu valoarea `product-template`.

Se va completa continutul elementelor cu clasele: `name` respectiv `price` cu valorile proprietatilor cu acelasi nume ale elementelor din proprietatea `items`.

```
<template id="product-template">
  <li>
    <div class="produs">
      <div class="name"></div>
      <img class="image" src="" alt="product-image">
      <div class="price"></div>
      <div class="add" data-index="">Adauga in cos</div>
    </div>
  </li>
</template>
```

De asemenea, se va seta valoarea atributului `src` pentru elementul cu clasa `image`. Elementele clonate se vor adauga in pagina in elementul care are `id` cu valoarea `data`. Pentru optimizare se recomanda folosirea unui `DocumentFragment`¹.

1.4 Sortarea produselor

Se va implementa sortarea produselor dupa nume sau dupa pret atat in ordine crescatoare cat si in ordine descrescatoare. Criteriul de sortare este specificat de valoarea elementului care are `id` cu valoarea `sortare`. Sortarea initiala a produselor este in ordine crescatoare a pretului. Se va folosi evenimentul `change` pentru elementul cu `id sortare` prezentat anterior. Dupa sortare se actualizeaza atributul `data-index` cu indexul curent al produsului.

¹ <https://developer.mozilla.org/en-US/docs/Web/API/DocumentFragment>

1.5 Adaugarea produselor in cosul de cumparaturi

Adaugarea produselor in cosul de cumparaturi se face pe evenimentul `click` la elementul cu clasa `add`.

Daca mai exista un produs cu acelasi nume in cos, se va incrementa cantitatea acestuia. Daca produsul nu mai exista in cosul de cumparaturi, cantitatea va fi 1.

Pentru adaugarea in cosul de cumparaturi se va clona continutul elementului template cu `id` `item-basket-template`. Se vor seta valorile pentru celulele cu clasele `price` si `total`. Se va seta sau actualiza valoarea pentru inputul cu clasa `cantitate`.

```
<template id="item-basket-template">
  <tr>
    <td class="name"></td>
    <td>
      <input class="cantitate" type="number" name="cantitate" value="1" min="1">
      <div class="remove" data-index="">Sterge</div>
    </td>
    <td class="price"></td>
    <td class="total"></td>
  </tr>
</template>
```

Elementul clonat se va adauga in pagina in elementul care are `id` cu valoarea `items`.

Dupa adaugare se actualizeaza atributul `data-index` cu indexul curent din cosul de cumparaturi.

1.6 Stergerea elementelor din cosul de cumparaturi

Atunci cand se face `click` pe elementul cu clasa `remove` se va sterge produsul din cosul de cumparaturi.

Pentru identificarea indexului elementului se poate folosi atributul `data-index`.

1.7 Actualizarea totalurilor

Totalurile se actualizeaza atunci cand se adauga sau sterg elemente din cosul de cumparaturi sau la modificarea cantitatilor produselor. Pentru modificarea cantitatilor produselor se foloseste un handler pe evenimentul `change`. Trebuie actualizate atat totalurile pentru produse cat si totalul cosului. Pentru setarea totalului cosului se va seta continutul elementului care are `id` cu valoarea `total`.

1.8 Evaluare

Solutia trimisa nu trebuie sa aiba erori la executie. O functionalitate se considera implementata atunci cand rezultatele sunt corecte. Pragul de promovare este de 75 puncte.

#	Criterii	Punctaj
1.	Adaugarea produselor in pagina	15p
2.	Implementarea sortarii pentru cele 4 criterii	15p
3.	Adaugarea produselor in cosul de cumparaturi	15p
4.	Stergerea produselor din cosul de cumparaturi	15p
5.	Actualizarea cantitatilor	15p
6.	Organizarea si formatarea codului	15p
7.	Trimiterea corecta a arhivei in formatul specificat	10p

1.9 Trimiterea arhivei

- Se arhiveaza directorul proiectului in format `zip`.
- Se trimite arhiva prin email fie ca atasament, fie ca link public la google drive
- Subiectul mesajului trebuie sa contina numele clasei.
- **Daca aveti neclaritati sau intampinati dificultati la implementarea functionalitatilor, trimiteti arhiva variantei la care ati ajuns in formatul prezentat anterior impreuna cu intrebarile aferente.**