

Programare web

Laboratorul 08

8. Concepte avansate

Cuprins

| | |
|----------------------------------|---|
| 8. Concepte avansate..... | 1 |
| 8.1. Concepte și tehnologii..... | 1 |
| 8.2. Structura proiectului | 1 |
| 8.3. Resurse utile | 2 |
| 8.4. Teme..... | 2 |
| 8.4.1. Tema 1 | 2 |
| 8.4.2. Tema 2 | 3 |
| 8.4.3. Tema 3 | 3 |

8.1. Concepte și tehnologii

În acest laborator veți lucra cu concepte precum prototipuri/clase, Promises, Web Storage, Web Worker, IndexedDB. De asemenea, veți lucra în continuare cu sistemul de versionare git și cu aplicația Visual Studio Code.

8.2. Structura proiectului

Structura proiectului `proiect1-numeUtilizatorGitHub` de până la acest laborator (inclusiv):

- `[server_web]`
 - `lanseaza_server.bat` (Windows) / `lanseaza_server.sh` (Linux) - lab 06
 - `server_web.py` (Python) / `ServerWeb.java` (Java) - lab 06, actualizat lab 07
- `[continut]`
 - `[css]`
 - `stil.css` - lab 04
 - `[imagini]`
 - `logo.png` - lab 03
 - `[js]`
 - `cumparaturi.js` - **lab 08**
 - `script.js` - lab 05
 - `persoane.js` - lab 07
 - `worker.js` - **lab 08**
 - `[resurse]`
 - `persoane.dtd` și/sau `persoane.xsd` - lab 01
 - `persoane.xml` - lab 01
 - `utilizatori.json` - lab 07
 - `acasa.html` - lab 07
 - `cumparaturi.html` - **lab 08**
 - `desen.html` - lab 03 tema 3, actualizat lab 04, lab 05, lab 07

- `despre.html` - lab 02, actualizat lab 03, lab 04, lab 05, lab07
- `favicon.ico` - lab 06
- `index.html` - lab 03, actualizat lab 04, lab 05, lab 07, **lab08**
- `inregistreaza.html` - lab 03, actualizat lab 04, lab 05, lab07
- `invat.html` - lab 05, lab07
- `persoane.html` - lab07
- `verifică.html` - lab07
- `video.html` - lab 03, actualizat lab 04, lab 05, lab07

8.3. Resurse utile

Pentru detalii privind limbajul CSS puteți să citiți cursurile 4, 5 și 6, sau să consultați documentația următoare:

- Tutorial JavaScript - <http://www.w3schools.com/js/default.asp>
- JavaScript | MDN - <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
- Tutorial AJAX - https://www.w3schools.com/xml/ajax_intro.asp
- Using the Web Storage API - https://developer.mozilla.org/en-US/docs/Web/API/Web_Storage_API/Using_the_Web_Storage_API
- Using Web Workers - https://developer.mozilla.org/en-US/docs/Web/API/Web_Workers_API/Using_web_workers
- Using IndexedDB - https://developer.mozilla.org/en-US/docs/Web/API/IndexedDB_API/Using_IndexedDB

8.4. Teme

8.4.1. Tema 1

Modificați proiectul de pe GitHub astfel încât să existe posibilitatea creării unei liste de cumpărături.

Adăugarea produselor se va face din pagina `cumparaturi.html`, care va fi integrată în site-ul existent.

Pentru a realiza acest lucru, continuați proiectul de pe GitHub și efectuați următorii pași (după fiecare pas, dați `git add` și `commit`):

- Creați în directorul `continut` un fișier cu numele `cumparaturi.html` în care trebuie să aveți un formular cu minim două căsuțe text `nume` și `cantitate`, și un buton `Adaugă`,
- Creați în directorul `continut/js` un fișier cu numele `cumparaturi.js` care să conțină logica de adăugare a unui item la lista de cumpărături,
- Pentru fiecare item, creați un obiect de tipul `Produs` (utilizați funcții constructor¹ sau clase²),
- Unde se justifică folosiți `promises`³ și `arrow functions`⁴,
- De fiecare dată când se apasă butonul `Adaugă`, utilizați Web Storage API - `localStorage`⁵ pentru a salva în browser produsul respectiv din lista de cumpărături (fiecare produs va conține și o proprietate `id` care reprezintă numărul de ordine al produsului).

¹ JavaScript Object Constructors - https://www.w3schools.com/js/js_object_constructors.asp

² JavaScript Classes - https://www.w3schools.com/js/js_classes.asp

³ Promises | MDN - https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Promise

⁴ Arrow function expressions | MDN

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Functions/Arrow_functions

⁵ Using the Web Storage API | MDN-

https://developer.mozilla.org/en-US/docs/Web/API/Web_Storage_API/Using_the_Web_Storage_API

8.4.2. Tema 2

Extindeți proiectul prin adăugarea următorului comportament paginii Cumpărături, astfel încât să arate similar cu figura:

Cumpărături

Nume Produs Cantitate

| Nr. | Nume Produs | Cantitate |
|-----|-------------|------------|
| 1. | Făină | 10 Kg |
| 2. | Ouă | 20 buc. |
| 3. | Drojdie | 10 plicuri |

- Creăți un Web Worker⁶ (fișierul `continut/js/worker.js`) care este notificat de script-ul principal de fiecare dată când s-a apăsut butonul Adaugă.
- De fiecare dată când worker-ul este notificat, acesta va afișa un mesaj la consola browserului și va notifica scriptul care l-a creat (`cumparaturi.js`), care la rândul său va adăuga o linie în tabelul cu lista de cumpărături.

8.4.3. Tema 3

Extindeți proiectul și permiteți utilizatorului să aleagă modalitatea de salvare a listei de cumpărături și, anume, dacă să fie utilizat `localStorage` din Web Storage API sau să fie utilizat `IndexedDB`⁷.

Creăți câte o clasă pentru fiecare dintre cele două modalități de stocare, și ambele clase trebuie să moștenească aceeași clasă de bază prin intermediul `class extends` sau `prototype`.

⁶ Using Web Workers - https://developer.mozilla.org/en-US/docs/Web/API/Web_Workers_API/Using_web_workers

⁷ Using IndexedDB - https://developer.mozilla.org/en-US/docs/Web/API/IndexedDB_API/Using_IndexedDB