

Universitatea Tehnică a Moldovei
Facultatea *Calculatoare, Informatică și Microelectronică*
Specialitatea *Tehnologii Informaționale*



Raport

la lucrarea de laborator nr. 3

Tema: “*JavaScript – crearea elementelor dinamice in pagina, verificare formulare*”

Disciplina: “Tehnologii web”

A efectuat:
A verificat:

Student grupa TI-231 FR
Asistent universitar

Apareci Aurica
Rusu Viorel

Chișinău 2025

Cuprins

1. Cadru teoretic.....	3
2. Repere teoretice	4
3. Sarcini practice.....	5
5. Concluzii.....	8
6. Webografie.....	9

1. Cadru teoretic

Tema lucrării: JavaScript - crearea elementelor dinamice in pagina, verificare formulare.

Scopul lucrării: Crearea elementelor dinamice in pagina

Sarcina practică:

- a) creati meniuri derulante utilizand JS,
- b) creati imagini defilante, afisati data/timp curent in timp real prin utilizarea
SetTimeout/ClearTimeout/SetInterval,
- c) verificati si validati formulare utilizand JS
- d) faceti alte implementari utile site-ului vostru utilizand JS .

2. Repere teoretice

JavaScript este un limbaj de programare interpretat, orientat pe evenimente, folosit în principal pentru dezvoltarea aplicațiilor web interactive. Creat inițial pentru a rula în browser, JavaScript permite manipularea Document Object Model (DOM), interacțiunea cu utilizatorul și modificarea dinamică a conținutului paginilor web.

JavaScript este considerat unul dintre cele trei piloni fundamentali ai dezvoltării web, alături de HTML și CSS.

1.2 Elemente dinamice în pagină

Prin "elemente dinamice" se înțelege acele componente HTML care sunt adăugate, eliminate sau modificate în timp real, în funcție de interacțiunea utilizatorului sau de alte condiții logice definite în codul JavaScript.

1.3 Meniuri derulante (dropdown)

Un **meniu derulant** este o componentă UI care permite utilizatorului să aleagă o valoare dintr-o listă de opțiuni ascunse inițial.

1.4 Afișarea datei și orei curente

JavaScript oferă clasa nativă `Date` pentru manipularea și afișarea timpului. Pentru afișarea în timp real se folosesc funcții precum:

`setInterval(func, ms)` – execută funcția *func* la fiecare *ms* milisecunde;

`setTimeout(func, ms)` – execută o singură dată funcția *func* după *ms* milisecunde;

`clearInterval()` și *`clearTimeout()`* – opresc execuțiile planificate anterior.

1.5 Verificarea și validarea formularelor

Validarea formularului reprezintă procesul prin care se verifică dacă datele introduse de utilizator respectă anumite condiții înainte de trimiterea acestora către server. JavaScript permite:

accesul la valorile introduse prin `form.input.value`;

afișarea mesajelor de eroare dinamice;

prevenirea trimiterii formularului cu `event.preventDefault()` dacă datele nu sunt valide.

1.6 Alte implementări utile cu JavaScript

- gestionarea interactivă a coșului de cumpărături;
- filtrarea sau sortarea elementelor într-o listă;
- animații și tranziții personalizate;
- manipularea `localStorage/sessionStorage` pentru salvarea datelor local.

3. Sarcini practice

În cadrul acestei lucrări, a fost dezvoltată *aplicația de tip eCommerce* realizată în cadrul laboratorului nr. 2, prin implementări de noi funcționalități.

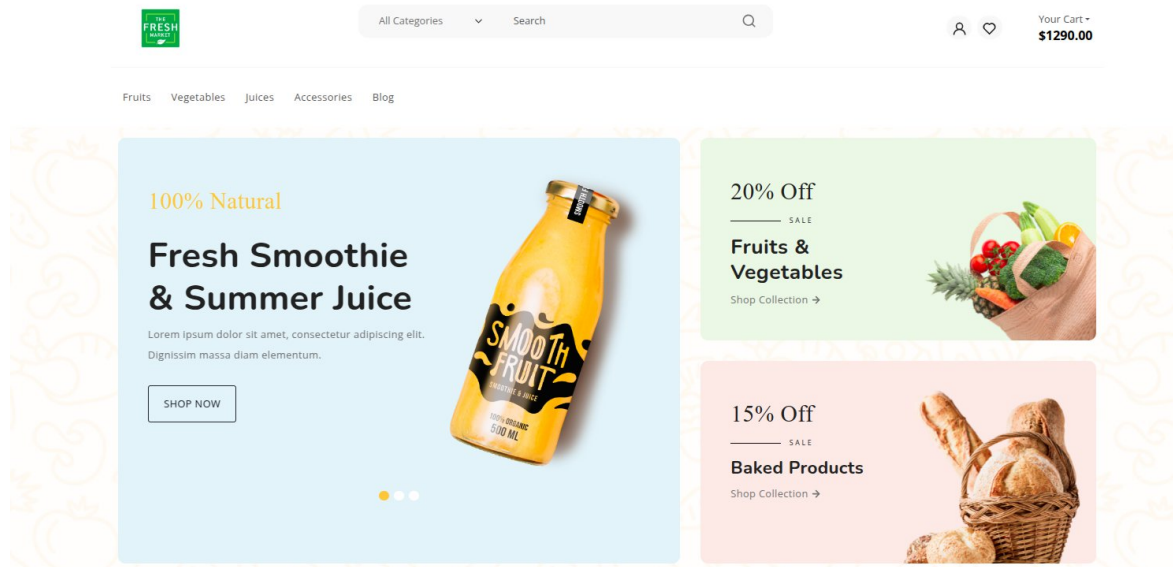
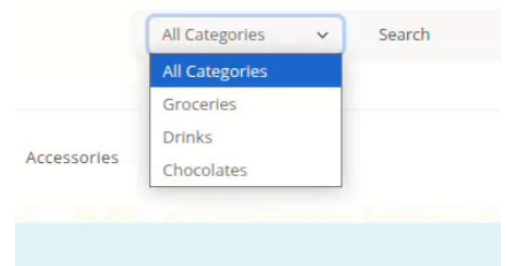


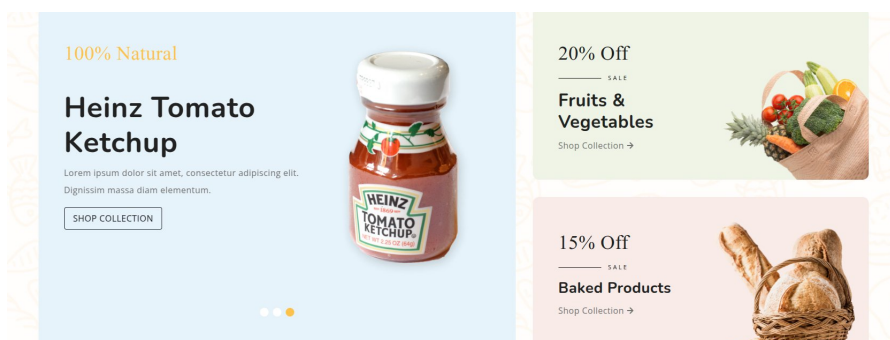
Figura 1 Interfața aplicației web

a) creați meniuri derulante utilizând JS,

```
<div class="cart text-end d-none d-lg-block dropdown">
  <button class="border-0 bg-transparent d-flex flex-column gap-2 lh-1" type="button" data-bs-toggle="offc
  <span class="fs-6 text-muted dropdown-toggle">Your Cart</span>
  <span class="cart-total fs-5 fw-bold">$1290.00</span>
</button>
</div>
</div>
```

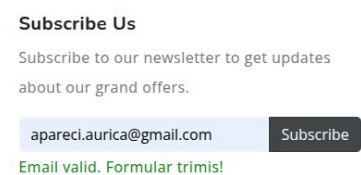
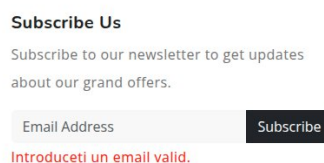
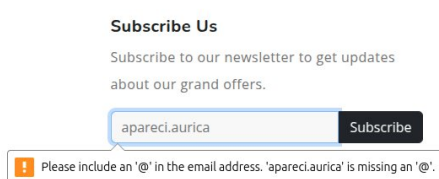


b) creați imagini defilante, afișați data/timp curent în timp real prin utilizarea SetTimeout/ClearTimeout/SetInterval.



```
var swiper = new Swiper(".main-swiper", {
  speed: 500,
  pagination: {
    el: ".swiper-pagination",
    clickable: true,
  },
  autoplay: {
    delay: 3000,
    disableOnInteraction: false
  },
  loop: true
});
```

c) verificați și validați formulare utilizând JS.



```

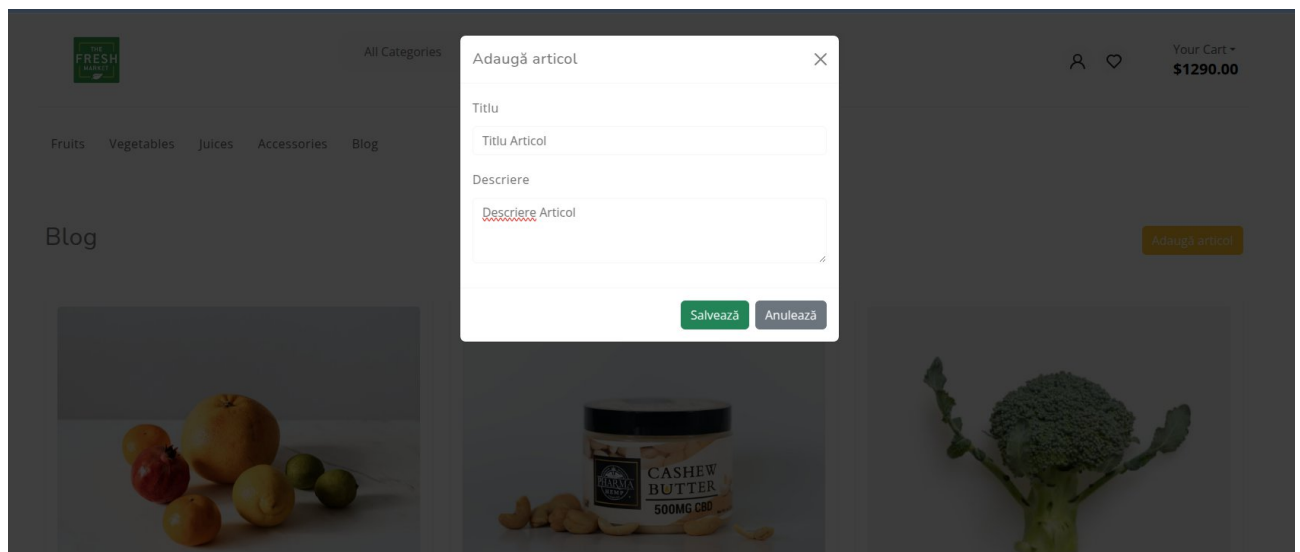
document.getElementById('subscribe-form').addEventListener('submit', function (e) {
    e.preventDefault();
    const emailInput = document.getElementById('email-input');
    const message = document.getElementById('form-message');
    const email = emailInput.value.trim();

    const emailPattern = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
    if (!emailPattern.test(email)) {
        message.textContent = "Introduceți un email valid.";
        message.style.color = 'red';
    } else {
        message.textContent = "Email valid. Formular trimis!";
        message.style.color = 'green';
    }
});

```

d) faceti alte implementari utile site-ului vostru utilizand JS . (adaugare articol nou)

În pagina blog.html a fost implementat un buton care, la apăsare, deschide o fereastră modală pentru introducerea unui nou articol. În interiorul acestei ferestre se află un formular simplu ce permite utilizatorului să completeze titlul și descrierea articolului. După trimiterea formularului, datele sunt preluate din câmpurile completate și utilizate pentru a genera dinamic un nou card de tip blog, care este afișat automat în secțiunea existentă de articole.



The screenshot displays a web application interface. At the top, there's a navigation bar with 'All Categories' and a 'Your Cart' summary showing a total of \$1290.00. Below the navigation, there's a 'Blog' section. A modal window titled 'Adaugă articol' is open, allowing users to add a new article. The modal contains two input fields: 'Titlu' (Title) and 'Descriere' (Description). Below these fields are two buttons: 'Salvează' (Save) and 'Anulează' (Cancel). The background shows a blog page with various food images and a sidebar with categories like Fruits, Vegetables, Juices, Accessories, and Blog. A 'Your Cart' summary shows a total of \$1290.00.

```

<div class="modal fade" id="addCardModal" tabindex="-1" aria-labelledby="addCardModalLabel"
aria-hidden="true">
<div class="modal-dialog">
<div class="modal-content">
<form id="card-form">
<div class="modal-header">
<h5 class="modal-title" id="addCardModalLabel">Adaugă articol</h5>
<button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
</div>
<div class="modal-body">
<div class="mb-3">
<label for="card-title" class="form-label">Titlu</label>
<input type="text" class="form-control" id="card-title" required>
</div>
<div class="mb-3">
<label for="card-description" class="form-label">Descriere</label>
<textarea class="form-control" id="card-description" rows="3" required></textarea>
</div>
</div>
<div class="modal-footer">
<button type="submit" class="btn btn-success">Salvează</button>
<button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Anulează</button>
</div>
</form>
</div>
</div>
</div>

```

```

<script>
const form = document.getElementById('card-form');
const modalEl = document.getElementById('addCardModal');
const cardsContainer = document.getElementById('blog-cards-container');

form.addEventListener('submit', function (e) {
  e.preventDefault();

  const title = document.getElementById('card-title').value.trim();
  const description = document.getElementById('card-description').value.trim();

  const newCardHTML = `
<div class="col-md-4">
  <article class="post-item card border-0 shadow-sm p-3">
    <div class="image-holder zoom-effect">
      <a href="#">
        
      </a>
    </div>
    <div class="card-body">
      <div class="post-meta d-flex text-uppercase gap-3 my-2 align-items-center">
        <div class="meta-date"><svg width="16" height="16"><use xlink:href="#calendar"></use></svg>${new Date().toLocaleDateString()}</div>
        <div class="meta-categories"><svg width="16" height="16"><use xlink:href="#category"></use></svg>custom</div>
      </div>
      <div class="post-header">
        <h3 class="post-title">
          <a href="#" class="text-decoration-none">${title}</a>
        </h3>
        <p>${description}</p>
      </div>
    </div>
  </article>
</div>
`;

  cardsContainer.insertAdjacentHTML('beforeend', newCardHTML);

  form.reset();

  const modalInstance = bootstrap.Modal.getOrCreateInstance(modalEl);
  modalInstance.hide();
});

```

5. Concluzii

Lucrarea de față a avut ca scop explorarea și aplicarea practică a limbajului JavaScript în contextul dezvoltării web moderne, cu accent pe crearea și manipularea elementelor dinamice într-o pagină HTML. Prin intermediul sarcinilor propuse, s-a demonstrat eficiența și flexibilitatea JavaScript-ului în îmbunătățirea interactivității și funcționalității unui site web.

Au fost realizate implementări esențiale precum: meniuri derulante care răspund la interacțiuni cu utilizatorul, un slider de imagini cu defilare automată și afișarea datei și orei curente în timp real folosind funcțiile `setTimeout`, `clearTimeout` și `setInterval`, validarea formularelor pentru prevenirea erorilor de introducere, precum și generarea dinamică a cardurilor de conținut prin intermediul unui formular plasat într-o fereastră modală.

Această sarcină nu a fost una complexă și nici nu a prezentat dificultăți semnificative, datorită experienței anterioare în lucrul cu JavaScript și dezvoltarea frontend. Cu toate acestea, ea a consolidat deprinderile legate de manipularea DOM-ului, gestionarea evenimentelor și crearea de funcționalități interactive care contribuie la o experiență mai bună pentru utilizator.

6. Webografie

1. <https://else.fcim.utm.md/course/view.php?id=2318>
2. <https://www.w3schools.com/>
3. <https://getbootstrap.com/docs/5.3/getting-started/introduction/>
4. <https://www.w3schools.com/js/DEFAULT.asp>
5. <https://chatgpt.com/>