UNIVERSITATEA DE STAT DIN MOLDOVA

FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ

DEPARTAMENTUL INFORMATICĂ

**Lupascu Iulii**

**Lucrul individual**

la disciplina „Java Script”

Coordonator: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ *Nume, prenume grad științific*

Chișinău, 2025

Cuprins

1-12: Lucrarea de laborator nr.1

12-25: Lucrarea de laborator nr.2a

25-44: Lucrarea de laborator nr.2b

44-49: Lucrare de laborator nr.3

49-54: Lucrare de laborator nr.4

Cuprins:

1. Structura HTML
2. Stilizare CSS
3. Funcționalitate JavaScript
   * Structura de date
   * Inițializare pagină
   * Schimbare categorie
   * Conversia valorilor
   * Dark Mode
   * Salvare în localStorage
4. Concluzie

1. HTML

<!DOCTYPE html>

<html lang="ro">

<head>

<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0"/>

<title>Convertor de unități</title>

<link rel="stylesheet" href="li.css" />

</head>

<body>

<div class="container">

<h1>Convertor de unități</h1>

<div class="category-buttons">

<button id="btnLength">Lungime</button>

<button id="btnWeight">Greutate</button>

<button id="btnTemp">Temperatură</button>

</div>

<input type="number" id="inputValue" placeholder="Introdu valoarea" />

<div class="unit-select">

<label for="fromUnit">Din:</label>

<select id="fromUnit"></select>

<label for="toUnit">În:</label>

<select id="toUnit"></select>

</div>

<button id="btnConvert">Converteste</button>

<button id="btnDarkMode">Dark Mode</button>

<div id="result">Rezultatul va apărea aici.</div>

</div>

<script src="li.js"></script>

</body>

</html>

**Explicație:**

* Se definește un formular cu butoane pentru alegerea categoriei (lungime, greutate, temperatură).
* Se introduce o valoare numerică.
* Se aleg două unități între care se face conversia.
* Se afișează rezultatul.
* Butonul „Dark Mode” permite schimbarea temei.

2. CSS

body {

font-family: Arial, sans-serif;

background-color: #f0f0f0;

color: #222;

transition: 0.3s;

}

body.dark {

background-color: #121212;

color: #f0f0f0;

}

.container {

max-width: 400px;

margin: 50px auto;

padding: 20px;

background: white;

border-radius: 10px;

box-shadow: 0 0 10px #aaa;

}

body.dark .container {

background: #1e1e1e;

}

input, select, button {

width: 100%;

padding: 10px;

margin-top: 10px;

font-size: 16px;

}

.category-buttons {

display: flex;

justify-content: space-between;

gap: 10px;

margin-bottom: 10px;

}

button {

cursor: pointer;

}

**Explicație:**

* Stiluri moderne pentru aspect curat.
* Clasa dark schimbă fundalul și textul pentru modul întunecat.
* Butoanele sunt aranjate în linie și responsive.

3. JS

1)Structura de date:

const units = {

length: ["meters", "kilometers", "millimeters"],

weight: ["grams", "kilograms", "pounds"],

temperature: ["celsius", "fahrenheit", "kelvin"]

};

let currentCategory = "length";

**Explicație:**  
Se definește un obiect care conține unitățile pentru fiecare categorie. Categoria curentă este inițial „length”.

2) Inițializare pagină:

window.onload = () => {

setCategory(currentCategory);

const saved = localStorage.getItem("lastResult");

if (saved) {

resultEl.innerText = saved;

}

if (localStorage.getItem("darkMode") === "true") {

document.body.classList.add("dark");

}

};

**Explicație:**

* La încărcarea paginii:
  + Se setează categoria implicită.
  + Se încarcă ultimul rezultat salvat.
  + Se activează tema dark dacă era salvată în localStorage.

3) Schimbare categorie:

function setCategory(category) {

currentCategory = category;

fromUnitEl.innerHTML = "";

toUnitEl.innerHTML = "";

units[category].forEach(unit => {

fromUnitEl.add(new Option(unit, unit));

toUnitEl.add(new Option(unit, unit));

});

fromUnitEl.selectedIndex = 0;

toUnitEl.selectedIndex = 1;

}

**Explicație:**  
Când utilizatorul apasă un buton de categorie, lista de unități se actualizează corespunzător.

4) **Conversia valorilor:**

function convert() {

const value = parseFloat(inputValueEl.value);

const from = fromUnitEl.value;

const to = toUnitEl.value;

if (isNaN(value)) {

resultEl.innerText = "Introdu o valoare validă.";

return;

}

let result = "";

switch (currentCategory) {

case "length":

result = convertLength(value, from, to);

break;

case "weight":

result = convertWeight(value, from, to);

break;

case "temperature":

result = convertTemperature(value, from, to);

break;

}

const output = `Rezultat: ${value} ${from} = ${result} ${to}`;

resultEl.innerText = output;

localStorage.setItem("lastResult", output);

}

**Explicație:**

* Se citește valoarea și se validează.
* Se apelează funcția de conversie corespunzătoare categoriei.
* Rezultatul este afișat și salvat.

5) Funcții de conversie:

function convertLength(value, from, to) {

const toMeters = { meters: 1, kilometers: 1000, millimeters: 0.001 };

return (value \* toMeters[from] / toMeters[to]).toFixed(4);

}

function convertWeight(value, from, to) {

const toGrams = { grams: 1, kilograms: 1000, pounds: 453.592 };

return (value \* toGrams[from] / toGrams[to]).toFixed(4);

}

function convertTemperature(value, from, to) {

let celsius;

if (from === "celsius") celsius = value;

else if (from === "fahrenheit") celsius = (value - 32) \* 5 / 9;

else if (from === "kelvin") celsius = value - 273.15;

let result;

if (to === "celsius") result = celsius;

else if (to === "fahrenheit") result = celsius \* 9 / 5 + 32;

else if (to === "kelvin") result = celsius + 273.15;

return result.toFixed(2);

}

**Explicație:**

* Lungime și greutate se convertesc prin raport la unitatea de bază (metri sau grame).
* Temperatura se transformă mai complex prin intermediul gradelor Celsius.

6) Dark Mode:

function toggleDarkMode() {

document.body.classList.toggle("dark");

const isDark = document.body.classList.contains("dark");

localStorage.setItem("darkMode", isDark);

}

**Explicație:**  
Adaugă sau elimină clasa .dark și salvează preferința în localStorage.

7) Evenimente:

btnLength.addEventListener("click", () => setCategory("length"));

btnWeight.addEventListener("click", () => setCategory("weight"));

btnTemp.addEventListener("click", () => setCategory("temperature"));

btnConvert.addEventListener("click", convert);

btnDarkMode.addEventListener("click", toggleDarkMode);

**Explicație:**  
La apăsarea fiecărui buton, se declanșează funcția asociată.

4. Concluzie

Acest proiect:

* Este un exemplu bun de aplicație interactivă în HTML/CSS/JavaScript.
* Salvează rezultatele și preferințele utilizatorului.
* Permite extensii viitoare (ex. adăugare de unități noi, mod light/dark avansat etc.).