**BOSNA I HERCEGOVINA**

**FEDERACIJA BiH**

**UNSKO-SANSKI KANTON**

**JU GIMNAZIJA „BIHAĆ“ U BIHAĆU**

**MATURSKI RAD IZ PREDMETA WEB TEHNOLOGIJE**

**WEB APLIKACIJA ZA MATRIČNI RAČUN KROZ JAVASCRIPT**

**Mentor: Učenik:**

**Mušić Muharem, MA Muslić Dino IV D**

**Bihać, maj 2022.**

# **UVOD**

Kako su matrice u matematici poprilično česte odlučio sam obraditi ovu temu da vježbanje matričnih zadataka učinim što zanimljivijim i interaktivnijim. S obzirom da računanje matrica može biti dugotrajan i spor proces ova aplikacija pomaže korisniku da u kratkom roku izračuna zbir, razliku ili proizvod dvije matrice ili da provjeri tačnost svoga zadataka kojeg je već uradio na papiru.

Kako novije generacije sve više koriste Internet i razne aplikacije smatram da bi mnogim učenicima učenje i ponavljanje gradiva postalo mnogo zanimljivije kada bi to vršili putem aplikacije u kojoj sa samo nekoliko klikova miša mogu da dođu do raznoraznih informacija i odgovora.

Svoju aplikaciju za matrični račun sam podijelio u 2 dijela.

Prvi dio aplikacije služi samo kao teoretska osnova za korisnika da nauči osnovne pojmove vezane za matrice, te formule koje će mu koristiti za vršenje matematičkih operacija sa matricama.

Drugi dio aplikacije jeste sam kalkulator pomoću kojeg korisnik unosi elemente i dimenzije matrica zajedno sa računskom operacijom (sabiranje, oduzimanje ili množenje). Pored računskih operacija korisnik može da: vrši zamjenu matrica, transponuje matrice i da nađe determinantu matrice.

Korisnik aplikaciju može da koristi kako bi naučio nešto novo ali i da ponovi i bolje savlada već naučeno gradivo.

# **MATRICE**

Matrica je pravougla tablica realnih brojeva s *m* redova i *n* kolona, omeđenih uglatim zagradama. Svaki broj koji se nalazi u matrici zove se *element* matrice. Pripadajući elementi matrice ne odvajaju se zarezom, već samo praznim prostorom.

A = B = C =

Matrica koja ima *m* redova i *n* kolona naziva se m × n matrica ili matrica reda (m, n).

Ako je *m* = *n* matricu nazivamo *kvadratna* matrica *n* – tog reda.

Pozicija u matrici svakog elementa matrice jednoznačno je određena pripadnošću elementa određenom redu i određenoj koloni matrice, te se označava notacijom s dvostrukim indeksima *aij*, što označava da se element nalazi na presjeku *i* – tog reda i *j* – te kolone

*a*11 = 1 *a*12 = -3

*a21* = 8 *a*22 = 4

*a31* = 2 *a*32 = -1

A =

Neki osnovni pojmovi o matricama:

* Kažemo da je *A* realna matrica ako je *Aij* ∈ *R* za sve elemente *Aij*.
* Kažemo da je *A* kompleksna matrica ako je *Aij* ∈ *C* za sve elemente *Aij.*
* Retčana matrica je matrica koja ima jedan redak.
* Stupčana matrica je matrica koja ima jedan stupac.
* Nul – matrica je matrica čiji su svi elementi jednaki nula.

## **SABIRANJE I ODUZIMANJE MATRICA**

Dvije matrice su prikladne za sabiranje ili oduzimanje ako je zadovoljen zahtjev da su matrice istog reda, te ih tek onda možemo sabirati (oduzimati). Sabiranje (oduzimanje) se definira kao zbir (razlika) odgovarajućih elemenata svakog para matrice.

Neka su *A, B* ∈ *M*nm(*F).* Zbir matrica *A* i *B* je matrica

*A* + *B* =

Kraći zapis:

*A*ij + [*B*ij] = [*A*ij + *B*ij]

Sabiranje i oduzimanje matrica nije definirano za matrice različitih dimenzija.

Ako su matrice *A* i *B* istog tipa tada je matrica *C* = *A* + *B* (*C* = *A* *– B)*  istog tipa i vrijedi:

*cij = aij* + *bij (cij = aij* - *bij)*

Sabiranje matrica je *komutativno* i *asocijativno:*

A = + B = = C =

A = + B = = C =

## **MNOŽENJE MATRICA**

Množenje matrica je specifična operacija koja čini osnovu matrične algebre. Redovi i kolone matrica mogu se promatrati kao vektori reda i vektori kolone odgovarajućih dimenzija; drugim riječima, bilo koja matrica se može tumačiti kao zbir vektora reda ili vektora kolone.

Množenje matrica *A* i *B* definira se samo ukoliko matrica *A* ima toliko kolona koliko matrica *B* ima redova.

Neka je matrica *A* tipa *m* × *k*, a matrica *B* tipa *k* × *n*, tada je matrica *C* = A • *B* tipa *m × n* i vrijedi:

*cij* = *ai*1*b*1*j + ai*2*b*2*j +… aikbkj*

Množenje matrica nije komutativno:

*AB ≠ BA*

A = • B = = C =

A = • B = = C =

Kažemo da je par matrica (*A, B*) *ulančan* ako *A* ima onoliko kolona koliko *B* ima redova.

Množenje matrica ima sljedeća svojstva:

* asocijativnost
* homogenost
* distributivnost

## **TRANSPONIRANA MATRICA**

Transponirana matrica je matrica u kojoj su redovi i kolone zamijenjeni.

Transponirana matrica matrice *A*  je matrica *AT*koja je definirana sa [*AT*]*ij* =  *Aij*.

Dakle, ako je matrica *A* tipa *m* × *n* tada je *AT* tipa *n* × *m:*

T

=

Matrica za koju je AT = *A* nazivamo *simetrična matrica*.

Neka su *A* i *B* matrice nad poljem *F.* Tada vrijedi:

* (*A*T)T = *A*
* *(A* + *B*)T = *A*T + *B*T
* (λ*A*)T = λ*A*T, λ ∈ *F*
* (*AB*)T = BT AT

Neka su *A* i *B* matrice nad poljem *C.* Tada vrijedi:

* (*A*\*)\* = *A*
* *(A* + *B*)\* = *A*\* + *B*\*
* (λ*A*)\* = λ*A*\*, λ ∈ *C*
* (*AB*)\* = B\* A\*

## **DETERMINANTA MATRICE**

Determinanta *A* je funkcija definirana na skupu svih kvadratnih matrica, a poprima vrijednost skupa skalara.

Determinanta matrice definira se induktivno, tj. determinanta matrice *n* – tog reda definira se pomoću determinante matrice (n – 1) – og reda.

Determinanta prvog reda: Determinanta matrice *A* = [*a*] je broj *a*.

Determinanta drugog reda: Determinantom matrice A =

zovemo broj *A* = *a11a22* – *a12a21.*

Determinanta trećeg reda: Determinanta matrice *A* =

je broj *A* = *a*11  – *a*21  + *a*31 .

Determinanta četvrtog reda: Determinanta matrice *A* =

je broj det *A* = *a*11det*A*11 – *a*21det*A*21 + … + (–1)n + 1 *a*n1det*A*n1 = k + 1 *a*k1det*A*k1.

Svojstva determinante

* Matrica *A* i transponirana matrica *A*T imaju jednake determinante

det *A* = det AT .

* Ako je *A* = (*a*ij) trokutasta matrica n – tog reda, onda je

det *A* = *a*11*a*22 … *anm* .

* Ako dvije kolone determinante zamijene mjesta, determinanta mijenja predznak.
* Ako matrica *A* ima dvije jednake kolone, onda je det *A* = 0.
* Determinanta se množi skalarom tako da se samo jedna kolona pomnoži tim skalarom, tj. zajednički faktor svih elemenata nekog stupca može se izvući ispred determinante.
* Determinanta ne mijenja vrijednost ako nekoj koloni determinante dodamo linearnu kombinaciju preostalih kolona.

# **IZRADA WEB APLIKACIJE**

****

Slika 1. Početna stranica

Prikaz početne stranice na kojoj se nalaze osnovne informacije o matricama zajedno sa navigacijskom trakom (Slika 1.).

Navigacijska traka (Slika 2.) nam služi kako bismo odabrali željenu oblast vezanu za matrice i kako bi ta oblast bila prikazana na ekranu.



Slika 2. Navigacijska traka

HTML kod navigacijske trake (Slika 2.)

<div class="nav">

     <h2>Navigacija</h2>

        <ul>

            <li><a href="#" data-naziv="matrice"><i class="fas fa-calculator-simple">M</i>Matrice</a></li>

            <li><a href="#" data-naziv="sabiranje"><i class="fas fa-plus"></i>Sabiranje</a></li>

            <li><a href="#" data-naziv="oduzimanje"><i class="fas fa-minus"></i>Oduzimanje</a></li>

            <li><a href="#" data-naziv="mnozenje"><i class="fas fa-times"></i>Množenje</a></li>

            <li><a href="#" data-naziv="determinanta"><i class="fas fa-wave-square"></i>Determinanta</a></li>

            <li><a href="#" data-naziv="transponacija"><i class="fas fa-exchange-alt"></i>Transponacija</a></li>

            <li><a href="#" data-naziv="kalkulator"><i class="fas fa-calculator"></i>Kalkulator</a></li>

        </ul>

    </div>

CSS kod navigacijske trake (Slika 2.)

.kontejner .nav{

  width: 200px;

  height: 100%;

  background: #191c38;

  padding: 30px 0px;

  position: fixed;

}

.kontejner .nav h2{

  color: #fff;

  text-transform: uppercase;

  text-align: center;

  margin-bottom: 30px;

}

.kontejner .nav ul li{

  padding: 15px;

  border-bottom: 1px solid #bdb8d7;

  border-bottom: 1px solid rgba(0,0,0,0.05);

  border-top: 1px solid rgba(255,255,255,0.05);

}

.kontejner .nav ul li a{

  color: #bdb8d7;

  display: block;

}

.kontejner .nav ul li a .fas{

  width: 25px;

}

.kontejner .nav ul li:hover{

  background-color: #252a52;

}

.kontejner .nav .social\_media{

  position: absolute;

  bottom: 0;

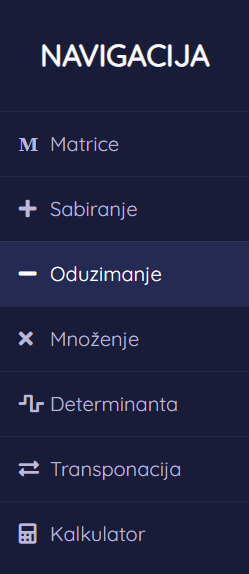
  left: 50%;

  transform: translateX(-50%);

  display: flex;

}

Kada hover – ujemo iznad željene oblasti ta oblast će promijeniti svoju boju (Slika 3.), a klikom na oblast ćemo dobiti ispit te oblasti na početnoj stranici (Slika 4.).

****

Slika 3.

Hover efekt navigacijske trake (Slika 3.) urađen je putem CSS – a:

.kontejner .nav ul li:hover{

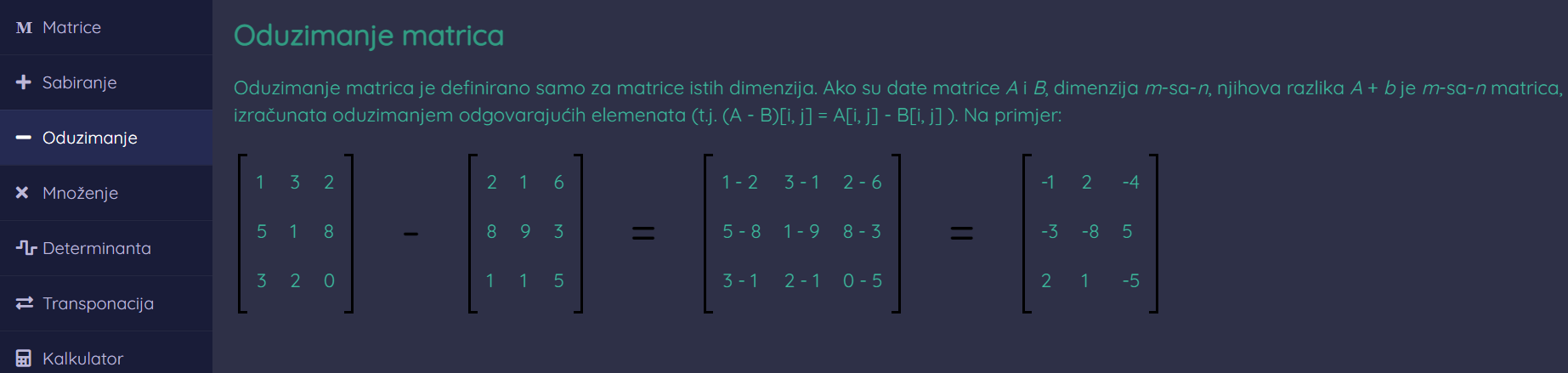
  background-color: #252a52;

}

.kontejner .nav ul li:hover a{

  color: #fff;

}



Slika 4. Nova oblast (oduzimanje)

JavaScript kod pomoću kojeg biramo željenu oblast:

const dodatneInfo = document.querySelectorAll('.dodatneInfo')

dodatneInfo.forEach(e => {

    e.style.display = 'none';

});

const matriceInfo = document.querySelector('.matriceInfo');

const tagovi = document.querySelectorAll('a');

tagovi.forEach(e => {

    e.addEventListener('click', () => {

        switch(e.dataset.naziv) {

            case 'matrice':

               dodatneInfo.forEach(e2 => {

                    if(e2.id == 'matrice') {

                        e2.style.display = 'block'

                        e2.style.opacity = '0';

                        e2.style.animation = 'animacija1 1s ease-in 1';

                        e2.style.animationFillMode =  'forwards';

                    }

                    else e2.style.display = 'none';

               });

                break;

            case 'sabiranje':

               dodatneInfo.forEach(e2 => {

                    if(e2.id == 'sabiranje') {

                        e2.style.display = 'block'

                        e2.style.opacity = '0';

                        e2.style.animation = 'animacija1 1s ease-in 1';

                        e2.style.animationFillMode =  'forwards';

                    }

                    else e2.style.display = 'none';

               });

                break;

            case 'oduzimanje':

            dodatneInfo.forEach(e2 => {

                if(e2.id == 'oduzimanje') {

                    e2.style.display = 'block'

                    e2.style.opacity = '0';

                    e2.style.animation = 'animacija1 1s ease-in 1';

                    e2.style.animationFillMode =  'forwards';

                }

                else e2.style.display = 'none';

            });

                break;

            case 'mnozenje':

            dodatneInfo.forEach(e2 => {

                if(e2.id == 'mnozenje') {

                    e2.style.display = 'block'

                    e2.style.opacity = '0';

                    e2.style.animation = 'animacija1 1s ease-in 1';

                    e2.style.animationFillMode =  'forwards';

                }

                else e2.style.display = 'none';

            });

                break;

            case 'determinanta':

                dodatneInfo.forEach(e2 => {

                     if(e2.id == 'determinanta') {

                        e2.style.display = 'block'

                        e2.style.opacity = '0';

                        e2.style.animation = 'animacija1 1s ease-in 1';

                        e2.style.animationFillMode =  'forwards';

                     }

                     else e2.style.display = 'none';

                });

                break;

            case 'transponacija':

                dodatneInfo.forEach(e2 => {

                     if(e2.id == 'transponacija') {

                        e2.style.display = 'block'

                        e2.style.opacity = '0';

                        e2.style.animation = 'animacija1 1s ease-in 1';

                        e2.style.animationFillMode =  'forwards';

                     }

                     else e2.style.display = 'none';

                });

                break;

            case 'inverza':

                dodatneInfo.forEach(e2 => {

                     if(e2.id == 'inverza') {

                        e2.style.display = 'block'

                        e2.style.opacity = '0';

                        e2.style.animation = 'animacija1 1s ease-in 1';

                        e2.style.animationFillMode =  'forwards';

                     }

                     else e2.style.display = 'none';

                });

                break;

            case 'kalkulator':

                window.close('glavna\_stranica.html');

                window.open('../Kalkulator/kalkulator.html');

                break;

        }

    });

});

const matrice = document.getElementById('matrice');

matrice.style.display = 'block';

matrice.style.opacity = '1';

U gornjem desnom uglu može se vidjeti animacija brojeva koji se okreću beskonačno za 360° (Slika 5.).

****

Slika 5 Animacija brojeva

Animacija je urađena isključivo putem CSS – a:

@keyframes animacijaBrojevi {

  from { transform: rotate(0deg); }

  to { transform: rotate(360deg); }

}

Početna stranica također ima svoju vlastitu ikonu zajedno sa naslovom „Matrice | početna stranica“ (Slika 6.).



Slika 6.

HTML kod (Slika 6.)

<link rel="shortcut icon" href="../ico/mat.ico" type="image/x-icon">

<title>Matrice | početna stranica</title>

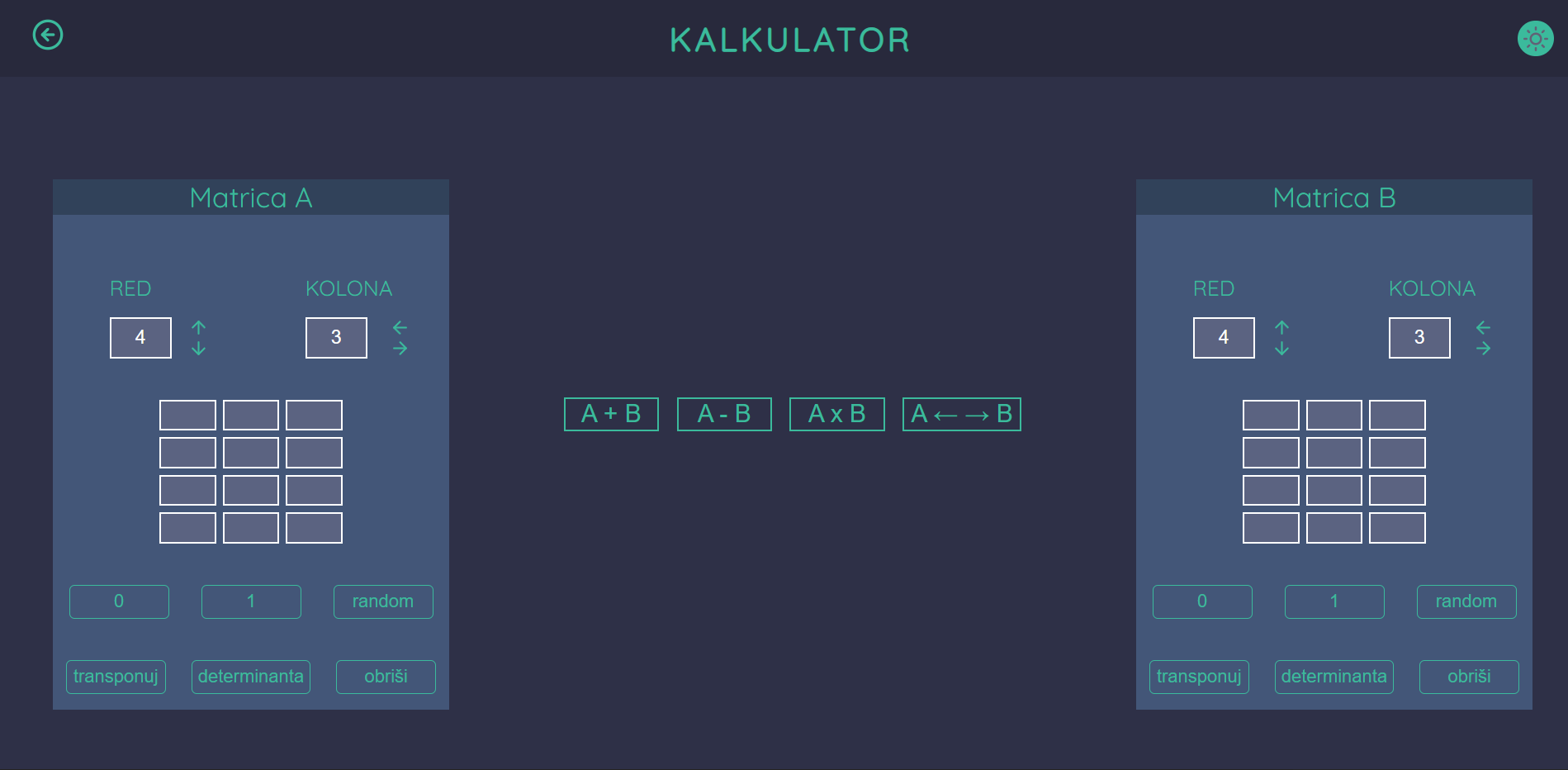
HTML kod pomoću kojeg smo povezali HTML dokument zajedno sa CSS dokumentom kako bi dobili izgled stranice (Slika 1.)

<link rel="stylesheet" href="glavna\_stranica.css">

Klikom na dugme „Kalkulator“ na navigacijskoj traci (Slika 7.) otvara se novi prozor (Slika 8.).



Slika 7. Kalkulator dugme



Slika 8. Novootvoreni prozor

U header – u novootvorenog prozora (Slika 8.) imamo 2 dugmeta. Prvo dugme jeste strelica nazad (Slika 9.) koja nas vraća na početnu stranicu kada korisnik klikne na nju.



Slika 9. Strelica nazad

Strelica je u projekat dodana kao SVG slika putem HTML – a:

<div id="nazad"><svg xmlns="http://www.w3.org/2000/svg" width="32" height="32" viewBox="0 0 24 24" fill="none" stroke="currentColor" stroke-width="2" stroke-linecap="round" stroke-linejoin="round" class="feather feather-arrow-left-circle"><circle cx="12" cy="12" r="10"></circle><polyline points="12 8 8 12 12 16"></polyline><line x1="16" y1="12" x2="8" y2="12"></line></svg></div>

Funkcionalnost strelice je odrađena putem JaveScript – a:

const nazad = document.getElementById('nazad');

nazad.onclick = () => {

    window.close('kalkulator.html');

    window.open('../GlavnaStranica/glavna\_stranica.html');

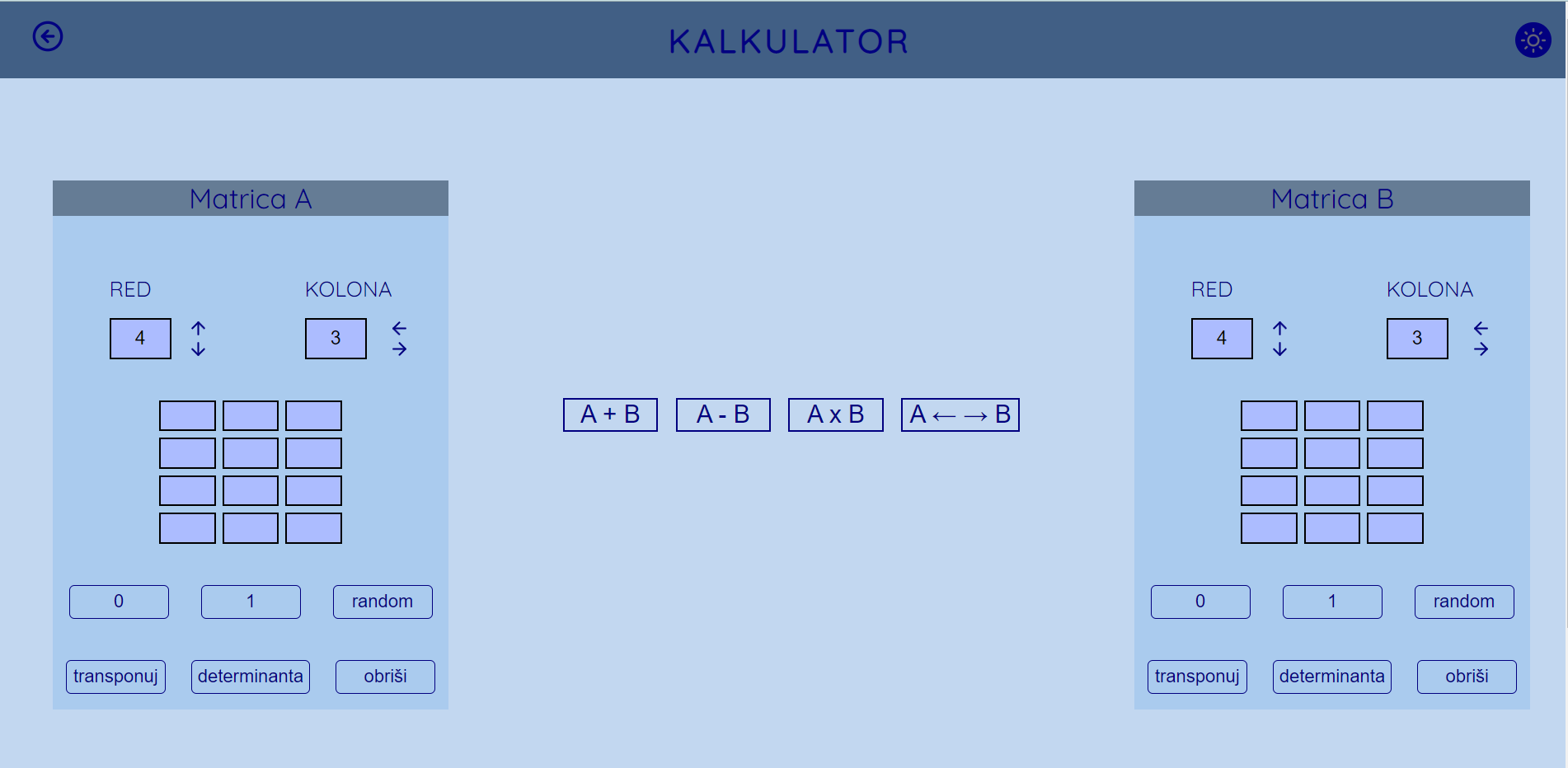
}

Drugo dugme koje se nalazi u header – u je dugme pomoću kojeg mijenjamo boju stranice (Slika 10.)



Slika 10.

Nakon što kliknemo na dugme stranica izgleda ovako:



Slika 11. Stranica sa izmijenjenim bojama

JavaScript kod za mijenjanje boje stranice:

sunce\_mjesec.addEventListener('click', () => {

    let INPUT = document.querySelectorAll('input');

    br++;

    if(br % 2 !== 0) {

        r.style.setProperty('--text', '#020080');

        body.style.setProperty('background-color', '#c2d7f0');

        header.style.setProperty('--header-background-color', '#0a2c57b3')

        sunce\_mjesec.style.setProperty('--text-hover', 'white');

        sunce\_mjesec.style.setProperty('--text', '#020080');

        INPUT.forEach(e => {

            e.classList.remove('input-matrice');

            e.classList.add('input-matrice-svijetli');

        });

    }else {

        r.style.setProperty('--text', '#3bba9c');

        body.style.setProperty('background-color', '#2e3047');

        header.style.setProperty('--header-background-color', '#252738b3');

        sunce\_mjesec.style.setProperty('--text-hover', 'white');

        sunce\_mjesec.style.setProperty('--text', '#3BBA9C');

        INPUT.forEach(e => {

            e.classList.remove('input-matrice-svijetli');

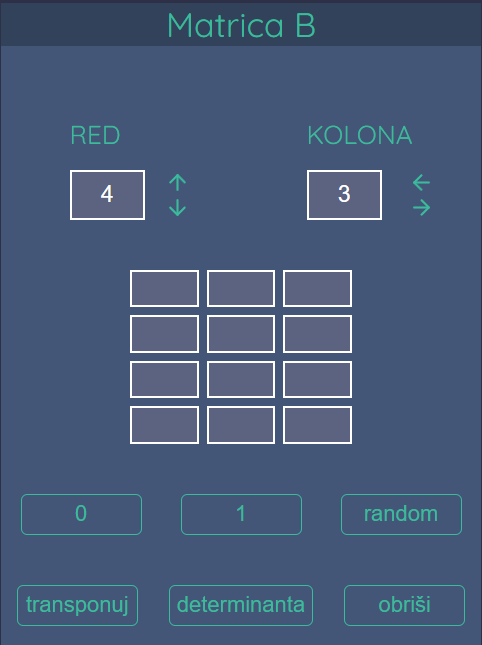
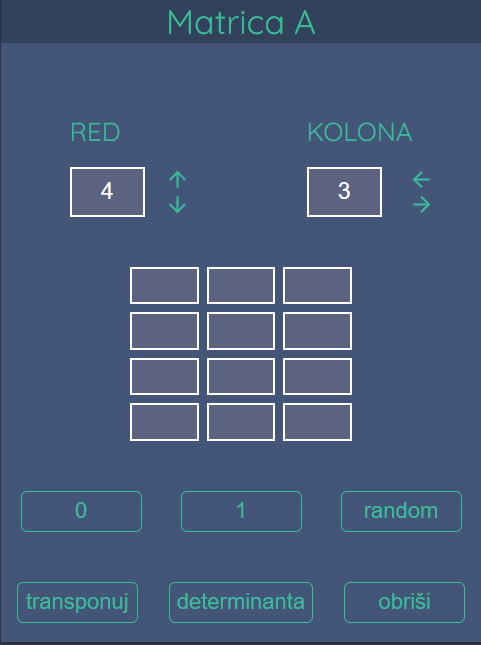
            e.classList.add('input-matrice');

        });

    }

});

Glavni element stranice jesu dvije matrice (Slika 12)



Slika 12. Matrica A i Matrica B

Obe matrice imaju identične funkcionalnosti.

HTML kod strukture matrica:

<div class="naslov-odabir" id="naslov-odabir1">

<center><div class="naslovMatrice" id="naslovMatrice1">Matrica A</div></center>

<div class="flex">

<div class="grid">

<div>RED</div>

<div class="red-kolona">

 <input readonly id="duzina1" class="duzina" type="number">

   <div class="kontejner-strelice">

 <svg id="strelica-gore1" xmlns="http://www.w3.org/2000/svg" width="24" height="24" viewBox="0 0 24 24" fill="none" stroke="currentColor" stroke-width="2" stroke-linecap="round" stroke-linejoin="round" class="strelica"><line x1="12" y1="19" x2="12" y2="5"></line><polyline points="5 12 12 5 19 12"></polyline></svg>

<svg id="strelica-dole1" xmlns="http://www.w3.org/2000/svg" width="24" height="24" viewBox="0 0 24 24" fill="none" stroke="currentColor" stroke-width="2" stroke-linecap="round" stroke-linejoin="round" class="strelica"><line x1="12" y1="5" x2="12" y2="19"></line><polyline points="19 12 12 19 5 12"></polyline></svg>

</div>

</div>

</div>

<div class="grid">

<div>KOLONA</div>

 <div class="red-kolona">

<input readonly id="sirina1" class="sirina" type="number">

<div class="kontejner-strelice">

<svg id="strelica-lijevo1" xmlns="http://www.w3.org/2000/svg" width="24" height="24" viewBox="0 0 24 24" fill="none" stroke="currentColor" stroke-width="2" stroke-linecap="round" stroke-linejoin="round" class="strelica"><line x1="19" y1="12" x2="5" y2="12"></line><polyline points="12 19 5 12 12 5"></polyline></svg><svg id="strelica-desno1" xmlns="http://www.w3.org/2000/svg" width="24" height="24" viewBox="0 0 24 24" fill="none" stroke="currentColor" stroke-width="2" stroke-linecap="round" stroke-linejoin="round" class="strelica"><line x1="5" y1="12" x2="19" y2="12"></line><polyline points="12 5 19 12 12 19"></polyline></svg>

</div>

</div>

 </div>

</div>

CSS kod za izgled i strukturu matrica:

.matrica {

    width: fit-content;

    display: grid;

    place-items: center;

    grid-gap: 0.4em;

}

#matrica1 {

    grid-template-columns: repeat(var(--kolona1), 1fr);

    grid-template-rows: repeat(var(--red1), 1fr);

}

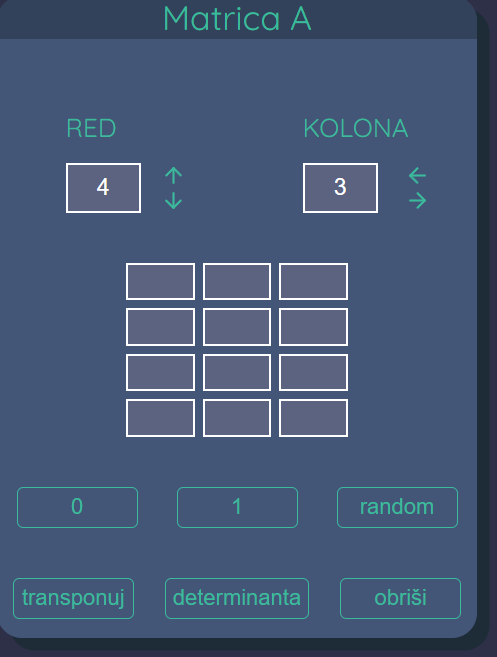
#matrica2 {

    grid-template-columns: repeat(var(--kolona2), 1fr);

    grid-template-rows: repeat(var(--red2), 1fr);

}

Ukoliko hover – ujemo na jednu od dvije matrice, matrica će dobiti zaobljene ivice i sijenu (Slika 13.)



Slika 13. Hover efekt na matrici

CSS kod hover efekta (Slika 13.)

@keyframes animacija-naslov-odabir1 {

    from {

        border-radius: 0;

        box-shadow: none;

    }

    to {

        border-radius: 20px;

        box-shadow: var(--box-shadow);

    }

}

@keyframes animacija-naslov-odabir2 {

    from {

        border-radius: 20px;

        box-shadow: var(--box-shadow);

    }

    to {

        border-radius: 0;

        box-shadow: none;

    }

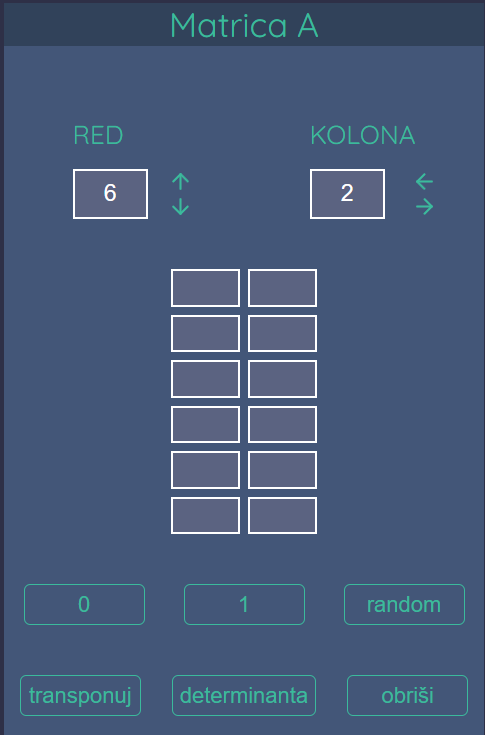
}

Matrice imaju input forme „RED“ i „KOLONA“ pored kojih se nalaze strelice(Slika 14.). Nakon što kliknemo na strelice promijenit će se broj redova ili kolona u zavisnosti koju smo strelicu kliknuli



Slika 14.

Matrica nakon što smo promijenili redove i kolone (Slika 15.):



Slika 15. Izmijenjena matrica

JavaScript kod pomoću kojeg mijenjamo dimenzije matrica:

const strelicaGore1 = document.querySelector('#strelica-gore1');

strelicaGore1.addEventListener('click', () => {

    if(duzinaMatrice1 < 8) {

        duzinaMatrice1++;

        duzinaMatricaInput.forEach((e, index) => {

            if(index === 0) {

                e.value++;

            }

        })

        mjestoZaMatricu1.style.gridTemplateColumns = `repeat(${sirinaMatrice1}, 1fr)`;

        mjestoZaMatricu1.style.gridTemplateRows = `repeat(${duzinaMatrice1}, 1fr)`;

        napraviMatricu1(duzinaMatrice1, sirinaMatrice1);

    }

});

const strelicaDole1 = document.querySelector('#strelica-dole1');

strelicaDole1.addEventListener('click', () => {

    if(duzinaMatrice1 > 1) {

        duzinaMatrice1--;

        duzinaMatricaInput.forEach((e, index) => {

            if(index === 0) {

                e.value--;

            }

        })

        mjestoZaMatricu1.style.gridTemplateColumns = `repeat(${sirinaMatrice1}, 1fr)`;

        mjestoZaMatricu1.style.gridTemplateRows = `repeat(${duzinaMatrice1}, 1fr)`;

        napraviMatricu1(duzinaMatrice1, sirinaMatrice1);

    }

})

const strelicaLijevo1 = document.querySelector('#strelica-lijevo1');

strelicaLijevo1.addEventListener('click', () => {

    if(sirinaMatrice1 > 1) {

        sirinaMatrice1--;

        sirinaMatricaInput.forEach((e, index) => {

            if(index === 0) {

                e.value--;

            }

        });

        mjestoZaMatricu1.style.gridTemplateColumns = `repeat(${sirinaMatrice1}, 1fr)`;

        mjestoZaMatricu1.style.gridTemplateRows = `repeat(${duzinaMatrice1}, 1fr)`;

        napraviMatricu1(duzinaMatrice1, sirinaMatrice1);

    }

});

const strelicaDesno1 = document.querySelector('#strelica-desno1')

strelicaDesno1.addEventListener('click', () => {

    if(sirinaMatrice1 < 8) {

        sirinaMatrice1++;

        sirinaMatricaInput.forEach((e, index) => {

            if(index === 0) {

                e.value++;

            }

        });

        mjestoZaMatricu1.style.gridTemplateColumns = `repeat(${sirinaMatrice1}, 1fr)`;

        mjestoZaMatricu1.style.gridTemplateRows = `repeat(${duzinaMatrice1}, 1fr)`;

        napraviMatricu1(duzinaMatrice1, sirinaMatrice1);

    }

});

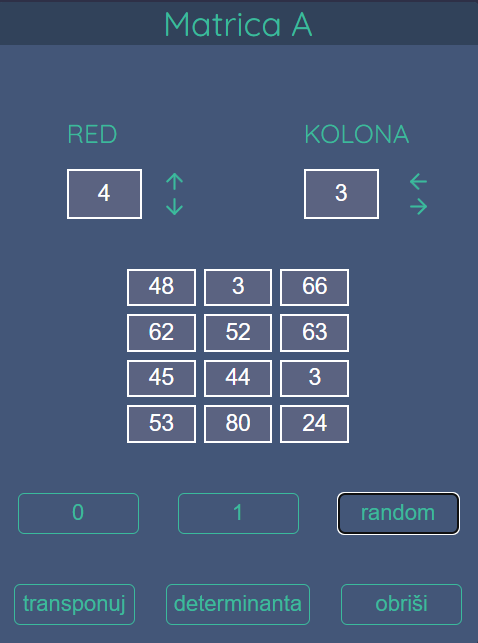
Bitno je napomenuti da je donja granica vrijednosti redova i kolona 1, a gornja 8.

Klikom na dugme „0“, „1“ ili „random“ (Slika 16.) cijela matrica će biti popunjena nulama, jedinicama ili nasumično generisanim brojevima u zavisnosti koje dugme smo kliknuli.



Slika 16.

Matrica nakon što smo kliknuli jedno dugme „random“ (Slika 17.):



Slika 17. Matrica sa randomiziranim poljima

JavaScript kod za popunjavanje polja matrica putem 3 dugmeta (Slika 16.):

Nule:

function sveNula1(duzina, sirina) {

    mjestoZaMatricu1.innerHTML = '';

    duzina = duzinaMatrice1;

    sirina = sirinaMatrice1;

    input1 = new Array(duzina);

    for(let i = 0; i < input1.length; i++) {

        input1[i] = new Array(sirina);

    }

    for(let i = 0; i < duzina; i++) {

        for(let j = 0; j < sirina; j++) {

            input1[i][j] = document.createElement('input');

            input1[i][j].type = 'number';

            if(br % 2 !== 0)

            input1[i][j].className = 'input-matrice-svijetli';

            else

            input1[i][j].className = 'input-matrice';

            input1[i][j].value = 0;

            mjestoZaMatricu1.appendChild(input1[i][j]);

        }

    }

}

Jedinice:

function sveJedan1(duzina, sirina) {

    mjestoZaMatricu1.innerHTML = '';

    duzina = duzinaMatrice1;

    sirina = sirinaMatrice1;

    input1 = new Array(duzina);

    for(let i = 0; i < input1.length; i++) {

        input1[i] = new Array(sirina);

    }

    for(let i = 0; i < duzina; i++) {

        for(let j = 0; j < sirina; j++) {

            input1[i][j] = document.createElement('input');

            input1[i][j].type = 'number';

            if(br % 2 !== 0)

            input1[i][j].className = 'input-matrice-svijetli';

            else

            input1[i][j].className = 'input-matrice';

            input1[i][j].value = 1;

            mjestoZaMatricu1.appendChild(input1[i][j]);

        }

    }

}

Random:

function random1(duzina, sirina) {

    mjestoZaMatricu1.innerHTML = '';

    duzina = duzinaMatrice1;

    sirina = sirinaMatrice1;

    input1 = new Array(duzina);

    for(let i = 0; i < input1.length; i++) {

        input1[i] = new Array(sirina);

    }

    for(let i = 0; i < duzina; i++) {

        for(let j = 0; j < sirina; j++) {

            input1[i][j] = document.createElement('input');

            input1[i][j].type = 'number';

            if(br % 2 !== 0)

            input1[i][j].className = 'input-matrice-svijetli';

            else

            input1[i][j].className = 'input-matrice';

            input1[i][j].value = parseInt(Math.random() \* 100 + 0);

            mjestoZaMatricu1.appendChild(input1[i][j]);

        }

    }

}

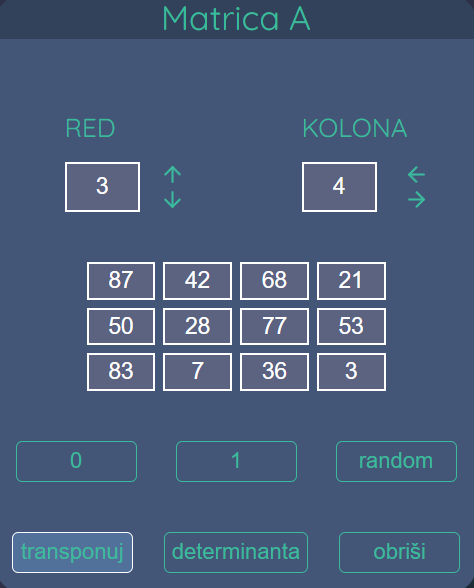
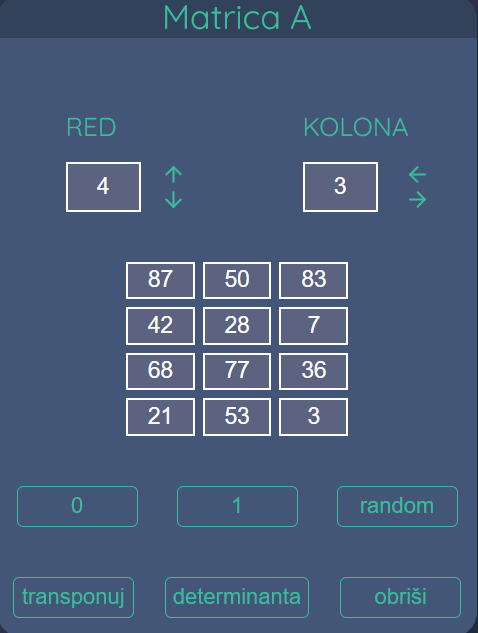
Pored navedena 3 dugmeta imamo još 3 dodatna dugmeta: „transponuj“, „determinanta“ i „obriši“(Slika 18.).



Slika 18.

Prvo dugme koje vidimo jeste dugme „transponuj“.

Klikom da dugme „transponuj“ transponovat ćemo unesenu matricu tj. zamijenit će se redovi i kolone matrice. Primjer rada dugmeta možemo vidjeti na slici „Slika 19“.



Slika 19. Transponacija matrice

JavaScript kod pomoću kojeg se vrši transponacija viđena na slici „Slika 19.“:

transponujBtn1.addEventListener('click', () => {

    [duzinaMatrice1, sirinaMatrice1] = [sirinaMatrice1, duzinaMatrice1];

    MATRICA1 = new Array(duzinaMatrice1)

    for(let i = 0; i < duzinaMatrice1; i++) {

        MATRICA1[i] = new Array(sirinaMatrice1);

    }

    for(let i = 0; i < duzinaMatrice1; i++) {

        for(let j = 0; j < sirinaMatrice1; j++) {

            MATRICA1[i][j] = input1[j][i].value;

        }

    }

    while (mjestoZaMatricu1.lastElementChild) {

        mjestoZaMatricu1.removeChild(mjestoZaMatricu1.lastElementChild);

      }

    duzina1.value = duzinaMatrice1;

    sirina1.value = sirinaMatrice1;

    input1Kopija = new Array(duzinaMatrice1)

    for(let i = 0; i < duzinaMatrice1; i++) {

        input1Kopija[i] = new Array(sirinaMatrice1);

    }

    for(let i = 0; i < duzinaMatrice1; i++) {

        for(let j = 0; j < sirinaMatrice1; j++) {

            input1Kopija[i][j] = document.createElement('input');

            input1Kopija[i][j].type = 'number';

            input1Kopija[i][j].value = MATRICA1[i][j];

            if(br % 2 !== 0)

            input1Kopija[i][j].className = 'input-matrice-svijetli';

            else

            input1Kopija[i][j].className = 'input-matrice';

            mjestoZaMatricu1.appendChild(input1Kopija[i][j]);

        }

    }

    input1 = input1Kopija;

    mjestoZaMatricu1.style.gridTemplateColumns = `repeat(${sirinaMatrice1}, 1fr)`;

    mjestoZaMatricu1.style.gridTemplateRows = `repeat(${duzinaMatrice1}, 1fr)`;

});

Drugo dugme jeste dugme „determinanta“ (Slika 19.). Klikom na to dugme ispisat će nam se determinanta unesene matrice (Slika 20.):



Slika 20. Determinanta

JavaScript kod za ispis determinante matrice:

const determinant = m =>

  m.length == 1 ?

  m[0][0] :

  m.length == 2 ?

  m[0][0]\*m[1][1]-m[0][1]\*m[1][0] :

  m[0].reduce((r,e,i) =>

    r+(-1)\*\*(i+2)\*e\*determinant(m.slice(1).map(c =>

      c.filter((\_,j) => i != j))),0)

determinanta1Btn.addEventListener('click', () => {

    detIspis1.innerHTML = '';

    if(duzinaMatrice1 == sirinaMatrice1) {

        matrica1 = new Array(duzinaMatrice1)

        for(let i = 0; i < duzinaMatrice1; i++) {

            matrica1[i] = new Array(sirinaMatrice1);

        }

        for(let i = 0; i < duzinaMatrice1; i++) {

            for(let j = 0; j < sirinaMatrice1; j++) {

                matrica1[i][j] = parseInt(input1[i][j].value);

            }

        }

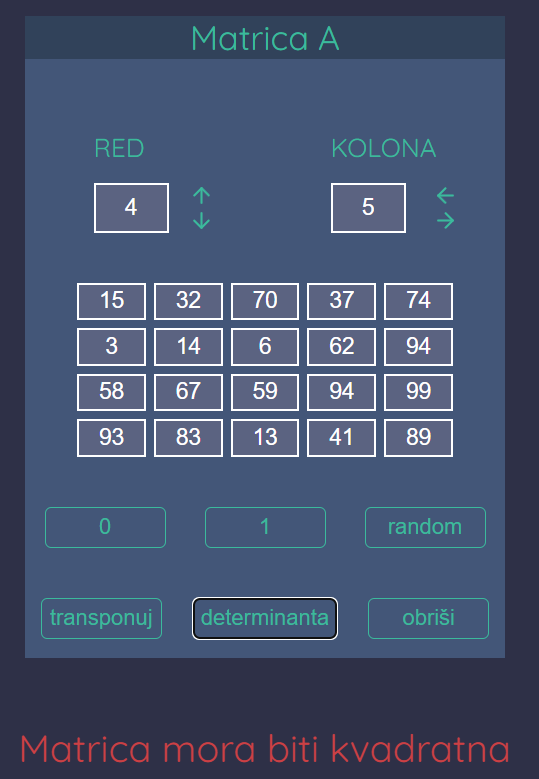
        Determinanta1 = determinant(matrica1);

        detIspis1.innerHTML = `Determinanta: ${Determinanta1}`;

        detIspis1.style.animation = 'animacija-ispis1 0.5s ease-in';

        detIspis1.style.animationFillMode =  'forwards';

Ukoliko unesena matrica nije kvadratna klikom na dugme „determinanta“ ispisat će se error „Matrica mora biti kvadratna“ (Slika 21.).



Slika 21. Error

JavaScript kod koji nam javlja error sa slike „Slika 21.“:

detIspis1.innerHTML = 'Matrica mora biti kvadratna';

        detIspis1.style.animation = 'animacija-ispis1 0.5s ease-in';

        detIspis1.style.animationFillMode =  'forwards';

        let interval = setInterval(() => {

            detIspis1.style.animation = 'animacija-ispis2 0.5s ease-out';

            detIspis1.style.animationFillMode =  'forwards';

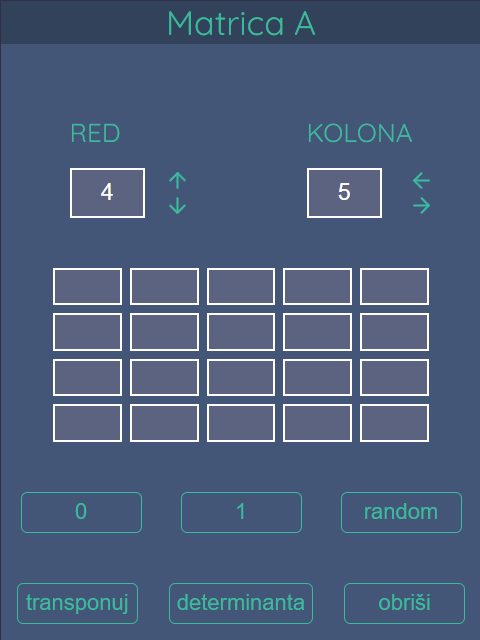
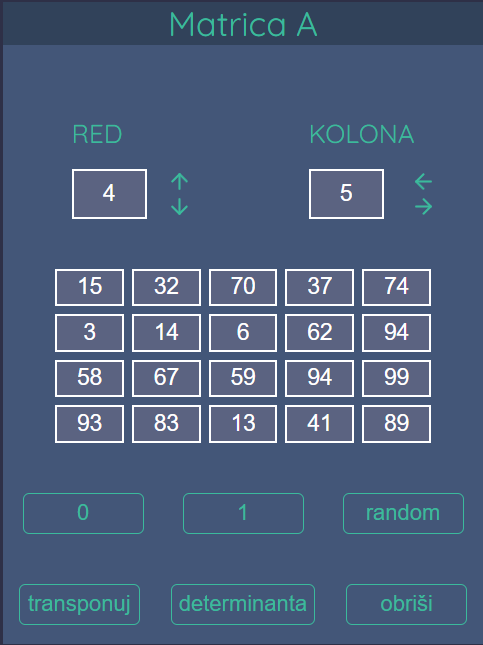
        }, 2500);

        setTimeout(() => {

            clearInterval(interval);

        }, 2500);

Treće dugme je dugme „obriši“ (Slika 19.) koje briše elemente matrice (Slika 22.).



Slika 22. Brisanje elemenata matrice

HTML kod navedenih 6 dugmića:

<div class="dugmici">

   <button class="btn" id="sve-01">0</button>

   <button class="btn" id="sve-11">1</button>

   <button class="btn" id="random1">random</button>

 </div>

 <br><br>

 <div class="dugmici">

   <button class="btn" id="transponuj1">transponuj</button>

   <button class="btn" id="determinanta1">determinanta</button>

   <button class="btn" id="obrisi1">obriši</button>

</div>

CSS kod navedenih 6 dugmića:

.dugmici {

    width: 100%;

    display: flex;

    justify-content: space-around;

}

.btn {

    font-size: 1.1em;

    min-width: 5.5em;

    color: var(--text);

    background-color: transparent;

    border: 1px solid var(--text);

    border-radius: 5px;

    padding: 5px;

    margin: 0 2%;

}

.btn:hover {

    cursor: pointer;

    background-color: rgba(116, 175, 233, 0.3);

    color: var(--text);

    border: 1px solid white;

}

U centru same aplikacije nalaze se 4 dugmeta koja imaju sljedeće uloge: zbir 2 matrice, razlika 2 matrice, proizvod 2 matrice i zamjena 2 matrice (Slika 23.).



Slika 23.

Hover – ovanjem iznad svakog od dugmića dobit ćemo inverz boja (Slika 24.).



Slika 24.

HTML kod dugmica sa slike „Slika 24“:

<div class="operacije">

button class="op-btn" id="plus">A + B</button>

 <button class="op-btn" id="minus">A - B</button>

<button class="op-btn" id="pomnozeno">A x B</button>

 <button class="op-btn" id="zamjena">A &larr; &rarr; B

</button>

 </div>

CSS:

.operacije {

    width: 100%;

    height: 30vh;

    display: flex;

    place-items: center;

    justify-content: space-around;

}

.op-btn {

    cursor: pointer;

    width: 20%;

    height: 15%;

    color: var(--text);

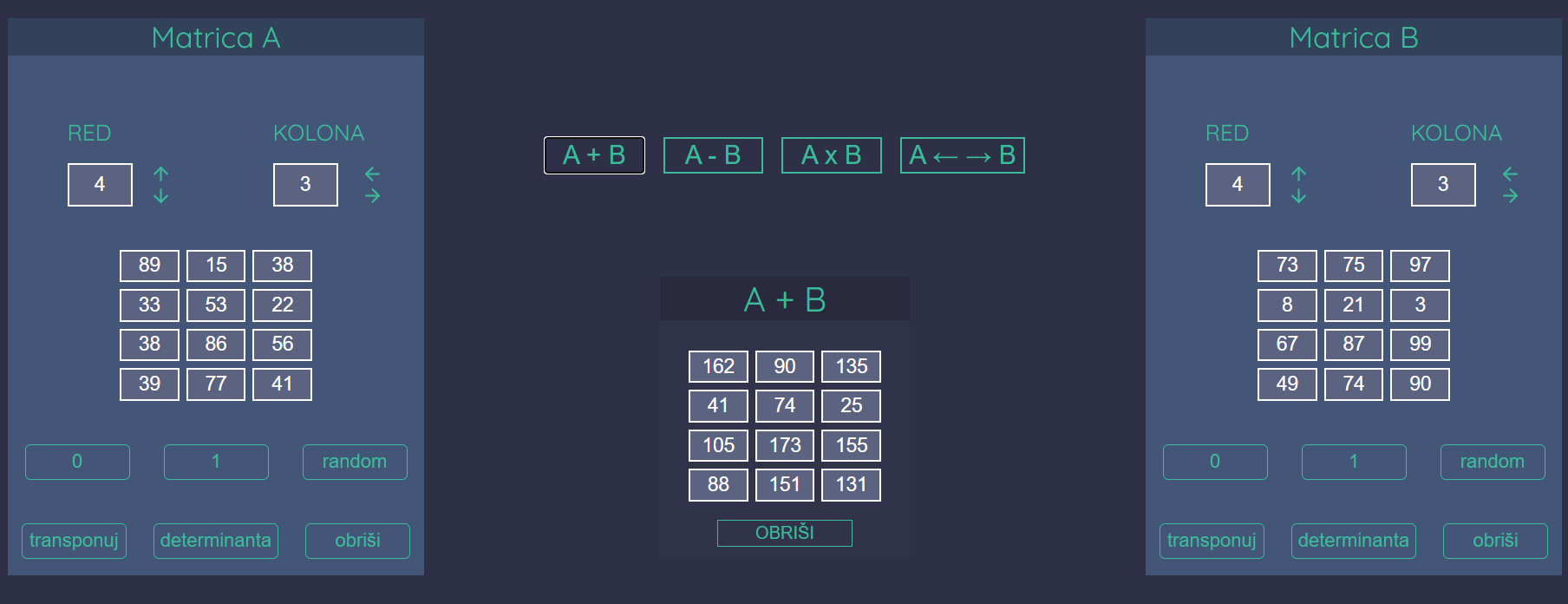
    background-color: transparent;

    font-size: 1.5em;

    border: 2px solid var(--text);

}

Klikom na dugme „A + B“ ispod dugmadi na slici „Slika 23.“ stvara se nova matrica „A + B“ koja daje zbir dvije unesene matrice (Slika 25.).



Slika 25. Zbir dvije matrice

JavaScript kod zbira matrica:

saberiBtn.addEventListener('click', () => {

    if(duzinaMatrice1 == duzinaMatrice2 && sirinaMatrice1 == sirinaMatrice2) {

        rez.innerHTML = '';

        rezultati.innerHTML = '';

        errori.innerHTML = '';

        detIspis2.innerHTML = '';

        rez.style.opacity = '1';

        rezultati.style.opacity = '1';

        naslovRez = document.createElement('div');

        naslovRez.innerHTML = '';

        naslovRez.innerHTML = 'A + B';

        naslovRez.className = 'naslov-rez';

        rez.appendChild(naslovRez);

        zbirMatrica = new Array(duzinaMatrice1);

        for(let i = 0; i < duzinaMatrice1; i++) {

            zbirMatrica[i] = new Array(sirinaMatrice1);

        }

        for(let i = 0; i < duzinaMatrice1; i++) {

            for(let j = 0; j < sirinaMatrice1; j++) {

                zbirMatrica[i][j] = parseInt(input1[i][j].value) + parseInt(input2[i][j].value);

            }

        }

        rezultat = new Array(duzinaMatrice1);

        for(let i = 0; i < duzinaMatrice1; i++) {

            rezultat[i] = new Array(sirinaMatrice1);

        }

        for(let i = 0; i < duzinaMatrice1; i++) {

            for(let j = 0; j < sirinaMatrice1; j++) {

                rezultat[i][j] = document.createElement('input');

                rezultat[i][j].type = 'number';

                rezultat[i][j].value = zbirMatrica[i][j];

                if(br % 2 !== 0)

                rezultat[i][j].className = 'input-matrice-svijetli';

                else

                rezultat[i][j].className = 'input-matrice';

                rezultati.appendChild(rezultat[i][j]);

            }

        }

        rezultati.style.gridTemplateColumns = `repeat(${sirinaMatrice1}, 1fr)`;

        rezultati.style.gridTemplateRows = `repeat(${duzinaMatrice1}, 1fr)`;

        rez.appendChild(naslovRez);

        rez.appendChild(rezultati);

        rez.appendChild(obrisiRezBtn);

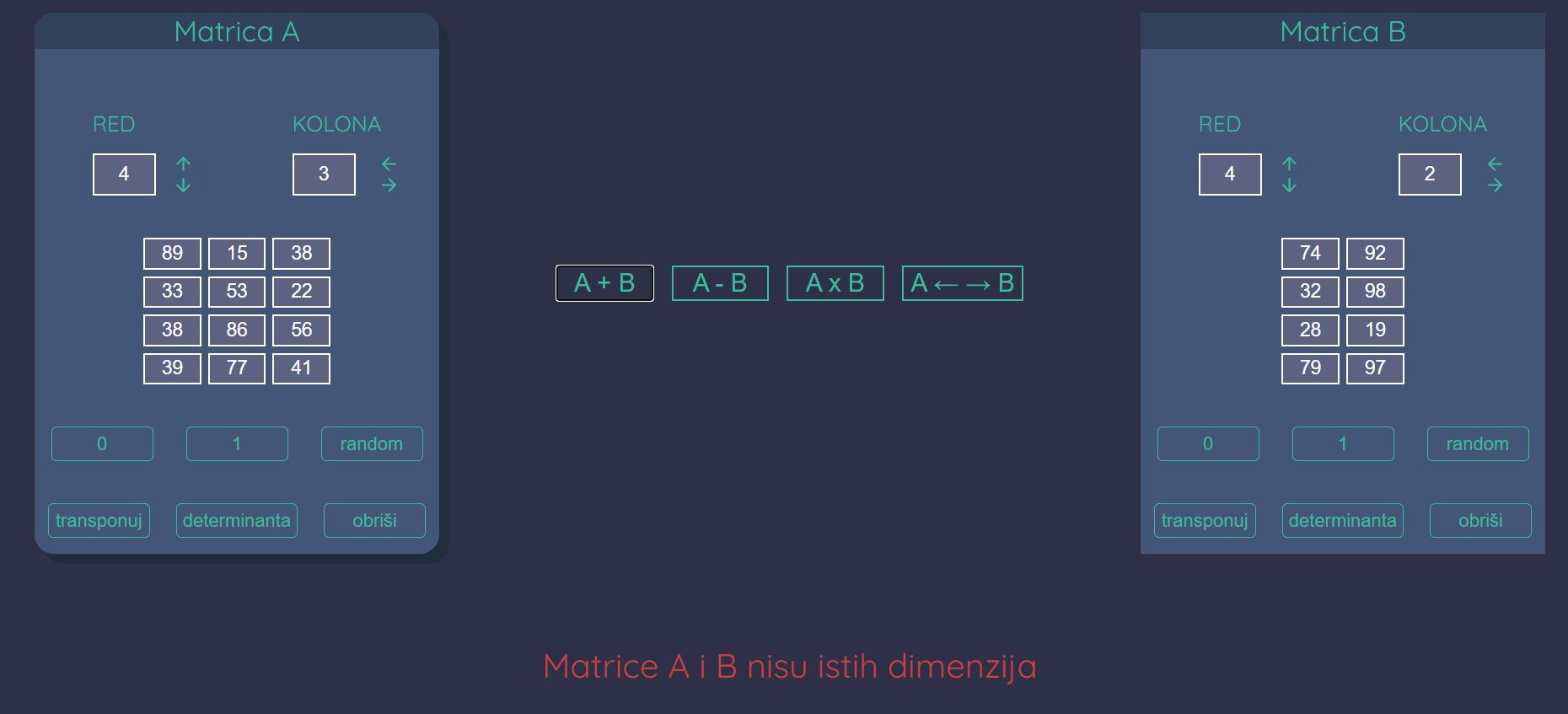
        obrisiRezBtn.addEventListener('click', () => {

            rez.style.opacity = '0';

            rez.innerHTML = '';

        });

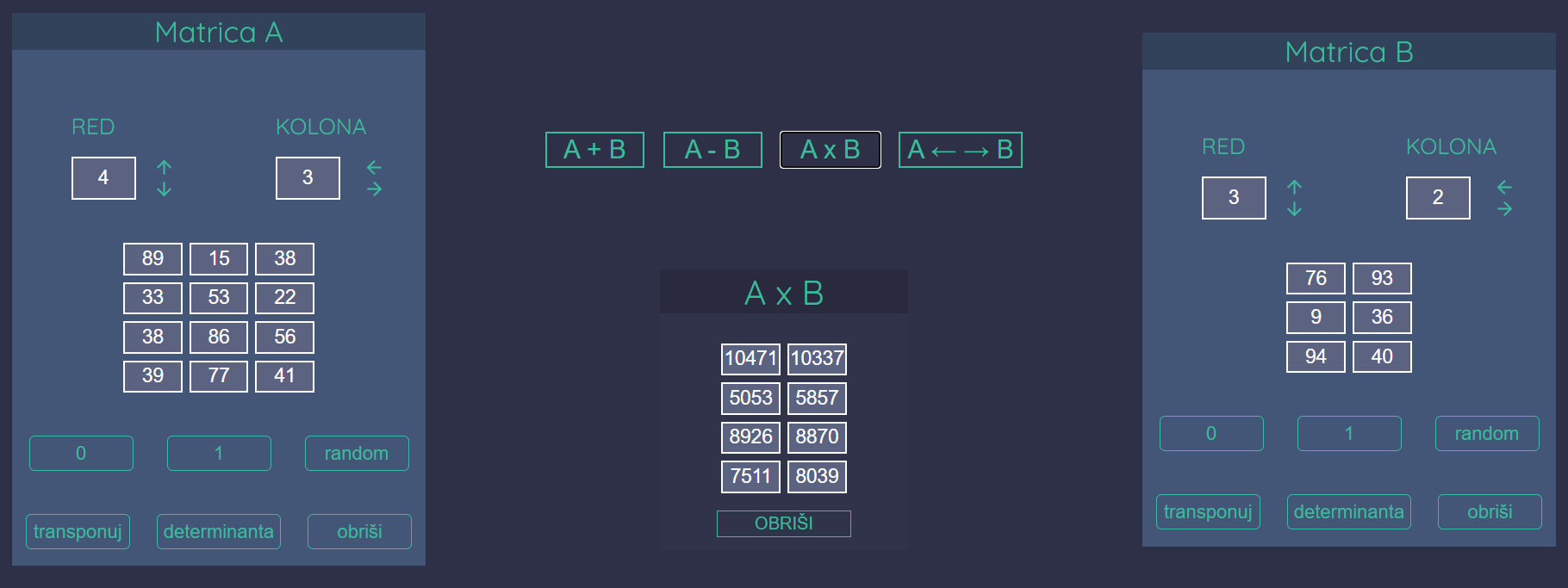
Ukoliko matrice nisu istih dimenzija javlja se error „Matrice A i B nisu istih dimenzija“ (Slika 26).



Slika 26. Error

Isto vrijedi i za dugme „A – B“.

Klikom na dugme „A x B“ dobivamo proizvod dvaju matrica (Slika 27.).



Slika 27. Proizvod dvije matrice

JavaScript kod koji nam daje proizvod dvije matrice:

pomnoziBtn.addEventListener('click', () => {

    if(sirinaMatrice1 == duzinaMatrice2) {

        rez.innerHTML = '';

        rezultati.innerHTML = '';

        detIspis1.innerHTML = '';

        detIspis2.innerHTML = '';

        rez.style.opacity = '1';

        rezultati.style.opacity = '1';

        naslovRez = document.createElement('div');

        naslovRez.innerHTML = '';

        naslovRez.innerHTML = 'A x B';

        naslovRez.className = 'naslov-rez';

        rez.appendChild(naslovRez);

        proizvodMatrica = new Array(duzinaMatrice1);

        for(let i = 0; i < duzinaMatrice1; i++) {

            proizvodMatrica[i] = new Array(sirinaMatrice2);

            for(let j = 0; j < sirinaMatrice2; j++) {

                proizvodMatrica[i][j] = 0;

                for(let k = 0; k < sirinaMatrice1; k++) {

                    proizvodMatrica[i][j] += parseInt(input1[i][k].value) \* parseInt(input2[k][j].value);

                }

            }

        }

        rezultat = new Array(duzinaMatrice1);

        for(let i = 0; i < duzinaMatrice1; i++) {

            rezultat[i] = new Array(sirinaMatrice2);

        }

        for(let i = 0; i < duzinaMatrice1; i++) {

            for(let j = 0; j < sirinaMatrice2; j++) {

                rezultat[i][j] = document.createElement('input');

                rezultat[i][j].type = 'number';

                rezultat[i][j].value = proizvodMatrica[i][j];

                if(br % 2 !== 0)

                rezultat[i][j].className = 'input-matrice-svijetli';

                else

                rezultat[i][j].className = 'input-matrice';

                rezultati.appendChild(rezultat[i][j]);

            }

        }

        rezultati.style.gridTemplateColumns = `repeat(${sirinaMatrice2}, 1fr)`;

        rezultati.style.gridTemplateRows = `repeat(${duzinaMatrice1}, 1fr)`;

        rez.appendChild(naslovRez);

        rez.appendChild(rezultati);

        rez.appendChild(obrisiRezBtn);

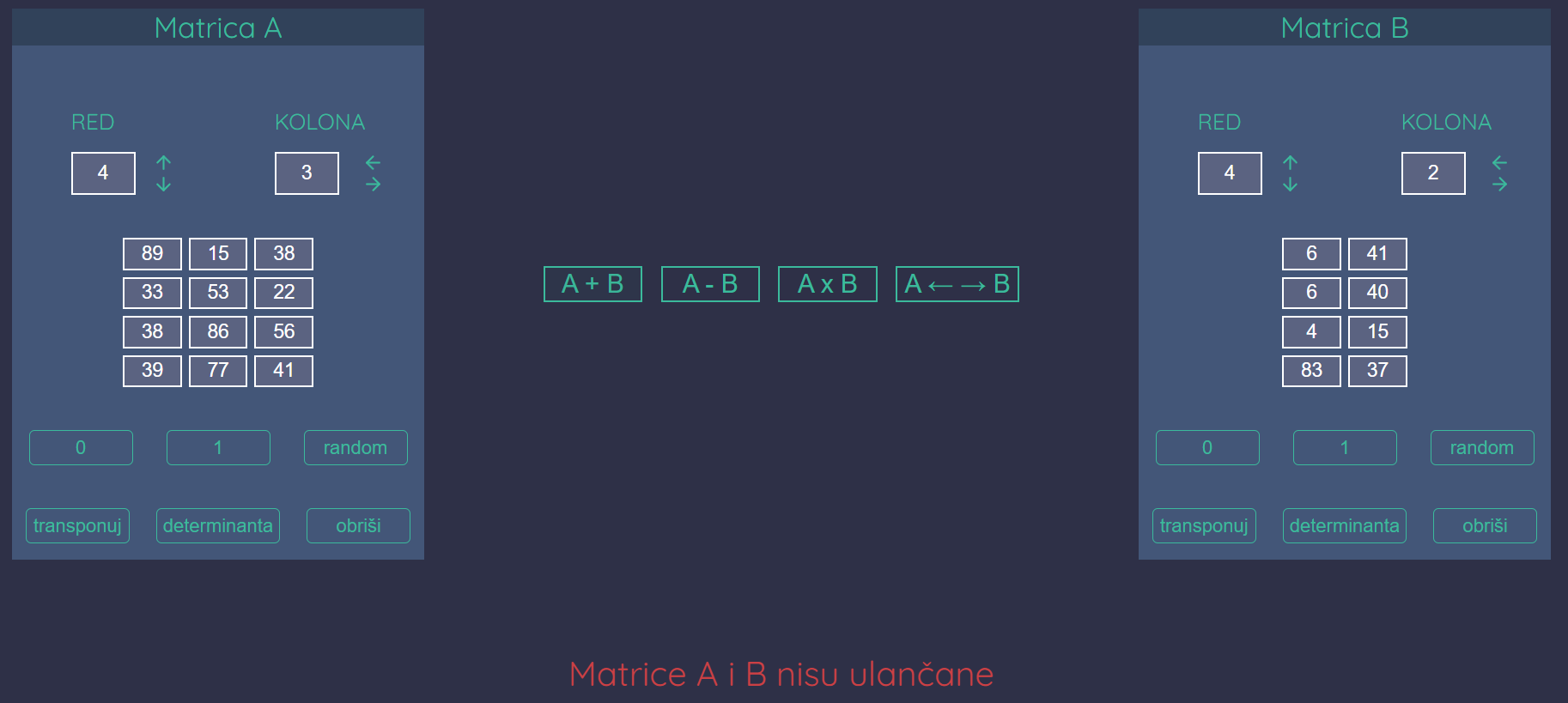
        obrisiRezBtn.addEventListener('click', () => {

            rez.style.opacity = '0';

            rez.innerHTML = '';

        });

Ukoliko matrice nisu ulančane prikazat će se error „Matrice A i B nisu ulančane“ (Slika 28.).



Slika 28. Error

JavaScript kod koji nam javlja error sa slike „Slika 28.“:

while(rez.firstChild) {

            rez.removeChild(rez.firstChild);

        }

        window.scrollTo(0, document.body.scrollHeight);

        errori.innerHTML = 'Matrice A i B nisu ulančane';

        errori.className = 'error';

        errori.style.animation = 'animacija-ispis1 0.5s ease-in';

        errori.style.animationFillMode =  'forwards';

        let interval = setInterval(() => {

            errori.style.animation = 'animacija-ispis2 0.5s ease-out';

            errori.style.animationFillMode =  'forwards';

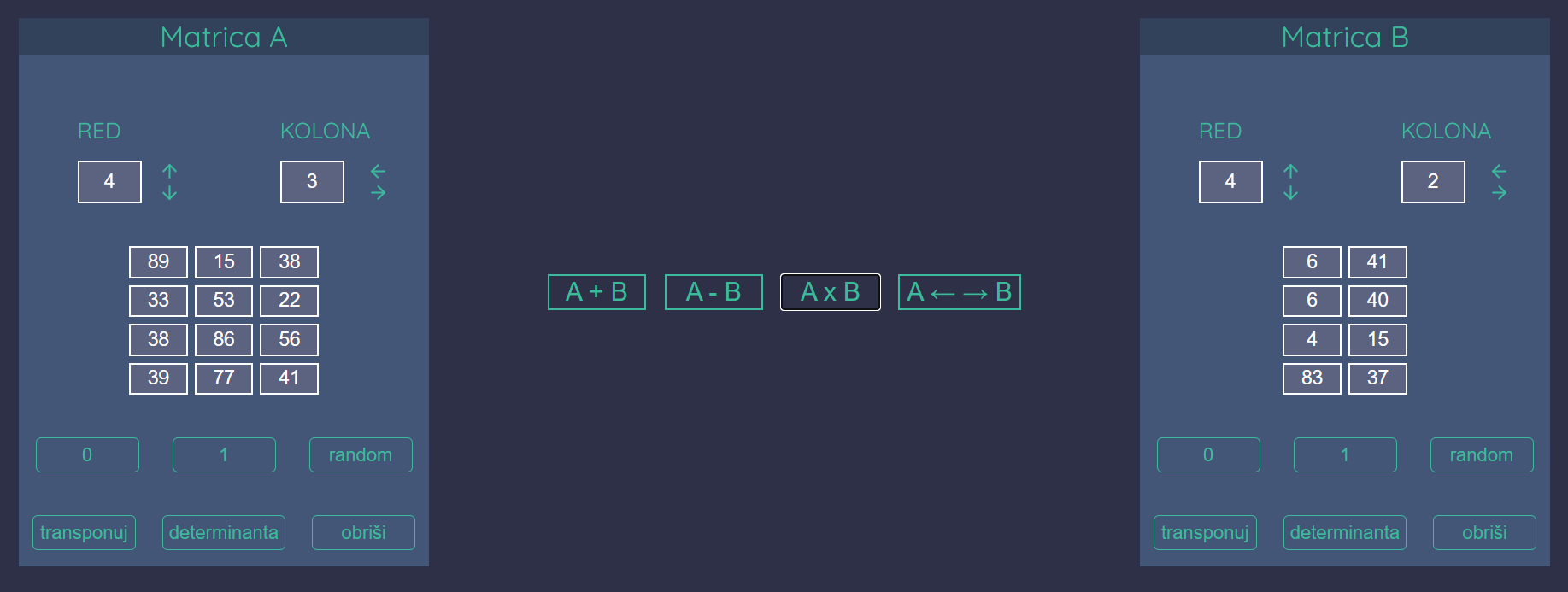
        }, 2500);

        setTimeout(() => {

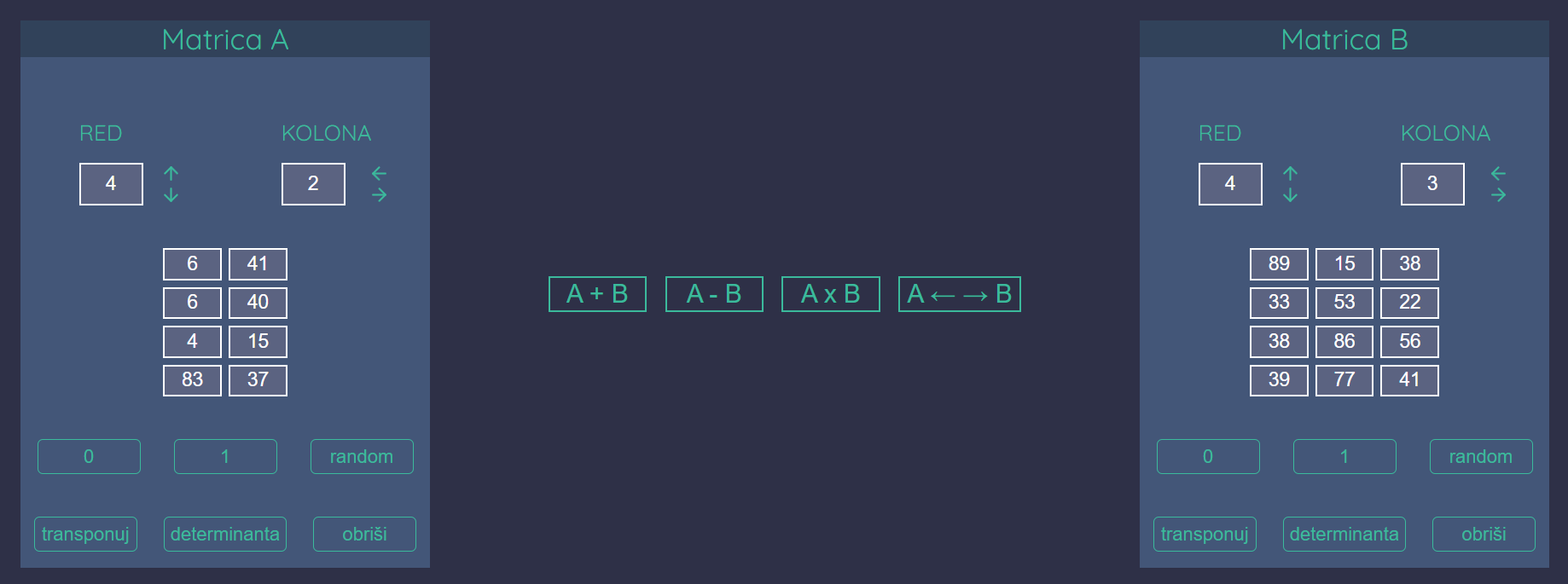
            clearInterval(interval);

        }, 2500);

Posljednje dugme je dugme „A 🡨 🡪 B“ koje zamjenjuje dimenzije i elemente matrica A i B (Slika 29. Slika 30.).



Slika 29.



Slika 30.

JavaScript kod pomoću kojeg zamjenjujemo matrice:

zamjeniBtn.addEventListener('click', () => {

    niz1 = new Array(duzinaMatrice1);

    for(let i = 0; i < duzinaMatrice1; i++) {

        niz1[i] = new Array(sirinaMatrice1);

    }

    for(let i = 0; i < duzinaMatrice1; i++) {

        for(let j = 0; j < sirinaMatrice1; j++) {

            niz1[i][j] = parseInt(input1[i][j].value);

        }

    }

    niz2 = new Array(duzinaMatrice2);

    for(let i = 0; i < duzinaMatrice2; i++) {

        niz2[i] = new Array(sirinaMatrice2);

    }

    for(let i = 0; i < duzinaMatrice2; i++) {

        for(let j = 0; j < sirinaMatrice2; j++) {

            niz2[i][j] = parseInt(input2[i][j].value);

        }

    }

    [duzinaMatrice1, duzinaMatrice2] = [duzinaMatrice2, duzinaMatrice1];

    [sirinaMatrice1, sirinaMatrice2] = [sirinaMatrice2, sirinaMatrice1];

    [duzina1.value, duzina2.value] = [duzina2.value, duzina1.value];

    [sirina1.value, sirina2.value] = [sirina2.value, sirina1.value];

    napraviMatricu1(duzinaMatrice2, sirinaMatrice2);

    napraviMatricu2(duzinaMatrice1, sirinaMatrice1);

    for(let i = 0; i < duzinaMatrice1; i++) {

        for(let j = 0; j < sirinaMatrice1; j++) {

            input1[i][j].value = niz2[i][j];

        }

    }

    for(let i = 0; i < duzinaMatrice2; i++) {

        for(let j = 0; j < sirinaMatrice2; j++) {

            input2[i][j].value = niz1[i][j];

        }

    }

    while(rez.firstChild) {

        rez.removeChild(rez.firstChild);

    }

    mjestoZaMatricu1.style.gridTemplateColumns = `repeat(${sirinaMatrice1}, 1fr)`;

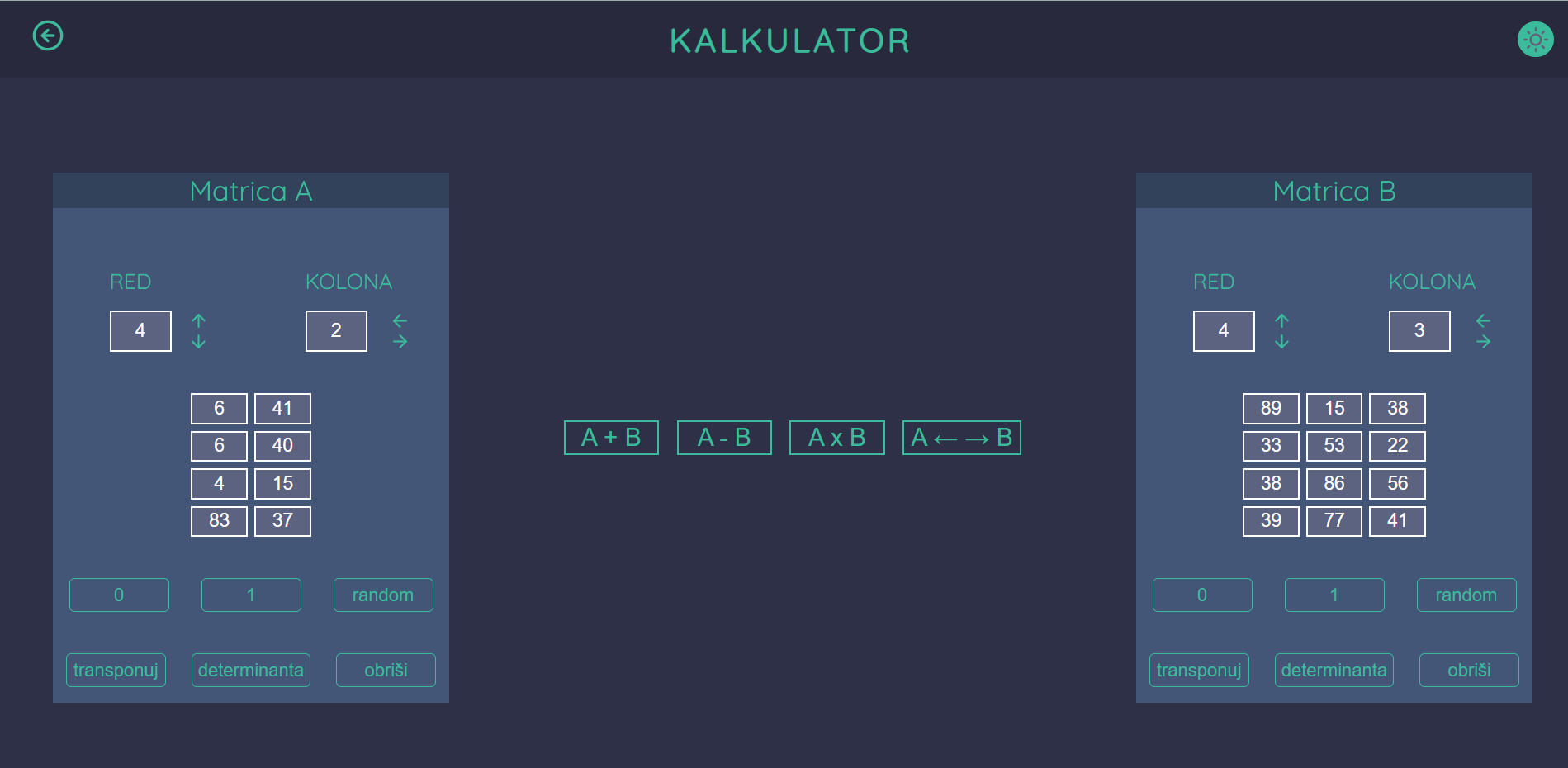
    mjestoZaMatricu1.style.gridTemplateRows = `repeat(${duzinaMatrice1}, 1fr)`;

    mjestoZaMatricu2.style.gridTemplateColumns = `repeat(${sirinaMatrice2}, 1fr)`;

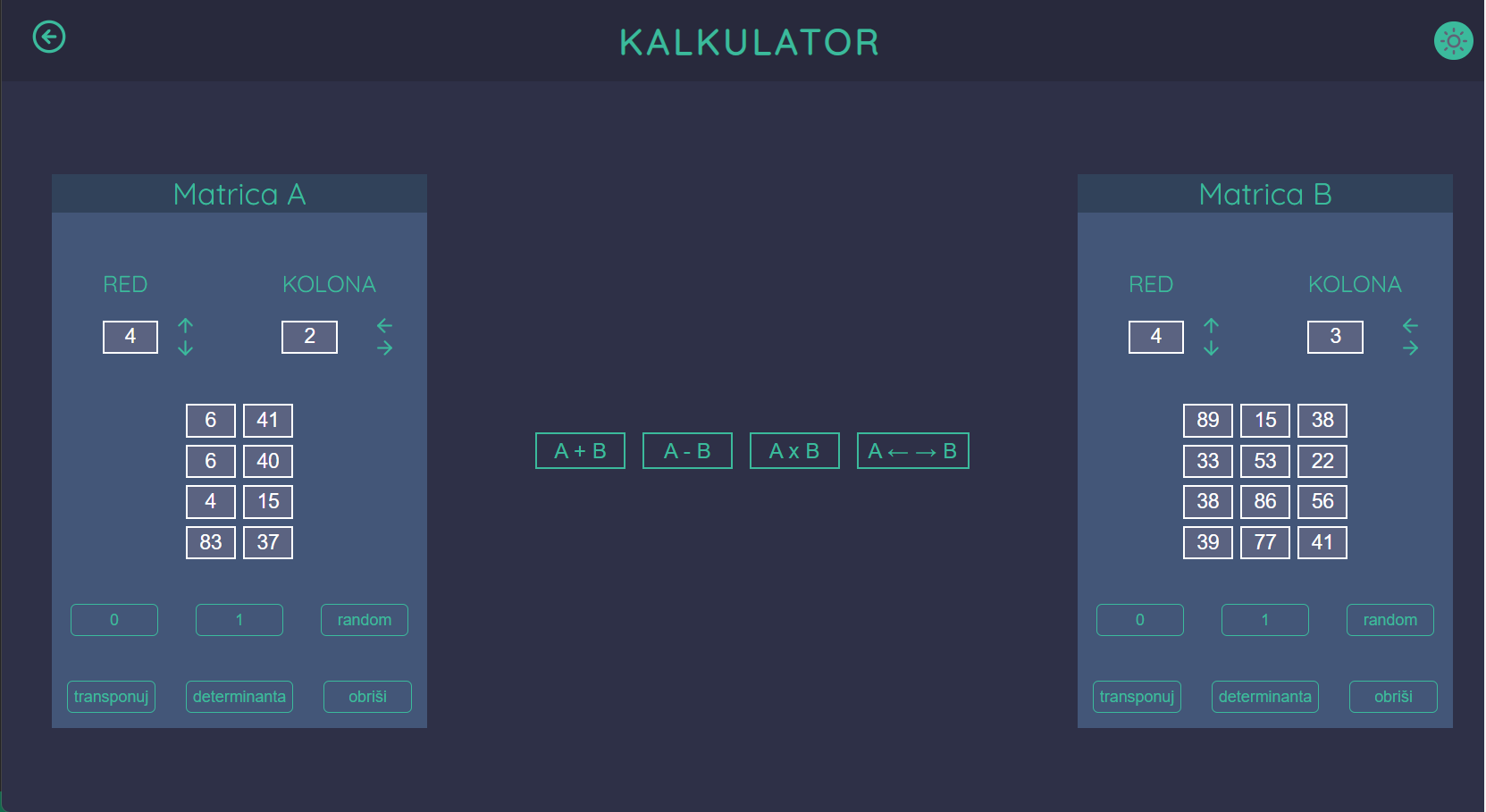
    mjestoZaMatricu2.style.gridTemplateRows = `repeat(${duzinaMatrice2}, 1fr)`;

});

Na kraju kako bi stranica bila responzivnija putem media querry – ja napravljeno je da kad se prozor suzi, dugmići i font postaju manji ili se mijenja raspored stranice kako se elementi stranice ne bi „sudarali“ (Slika 31., Slika 32., Slika 33., Slika 34. ).

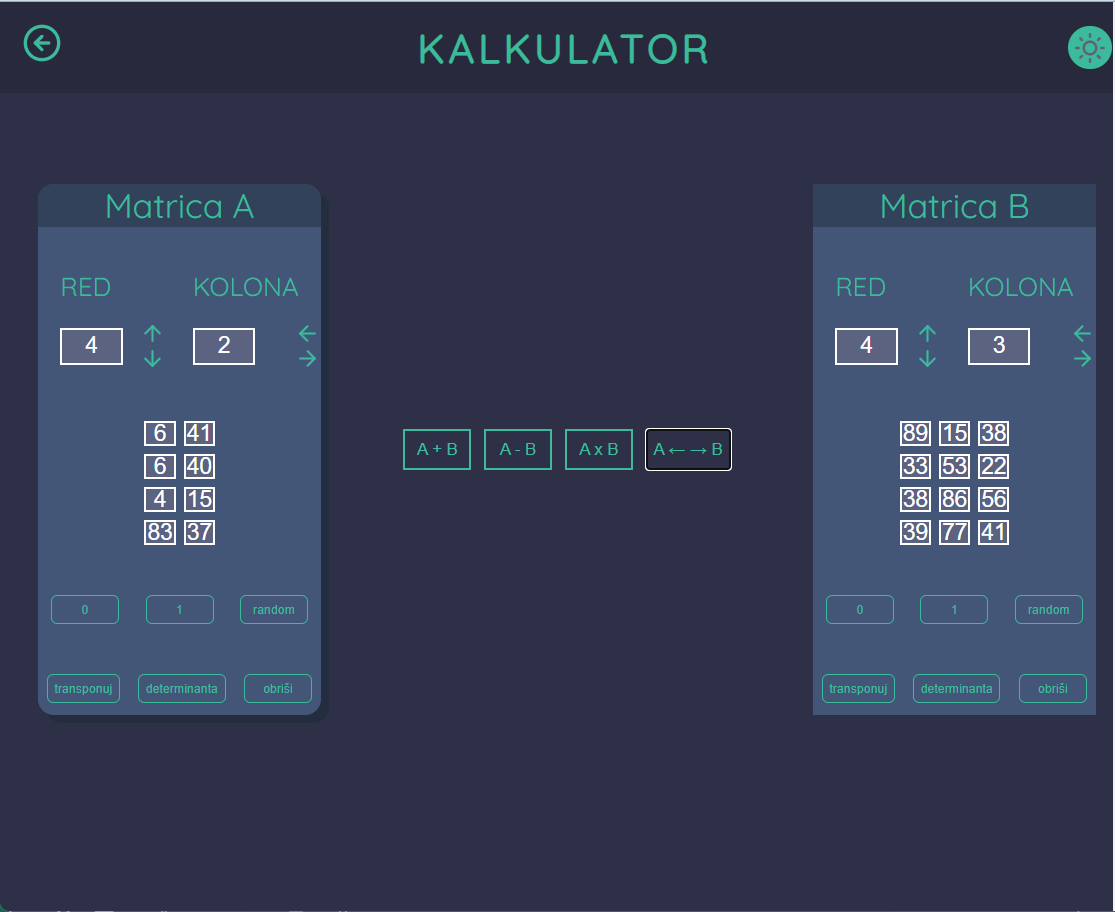


Slika 31.

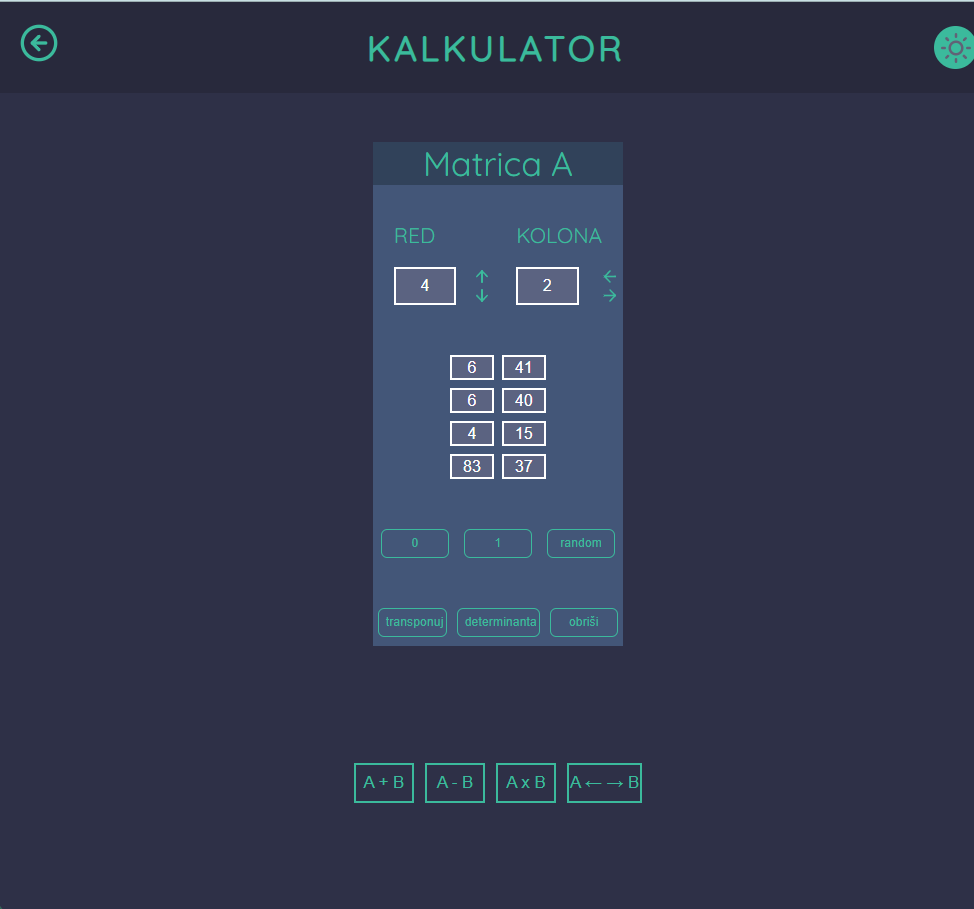


Slika 32.

.



Slika 33.



Slika 34.

Ovaj dio koda urađen je isključivo putem CSS – a:

@media only screen and (max-width: 1375px) {

    .dugmici {

        font-size: small;

    }

    .operacije {

        font-size: small;

    }

    .input-matrice {

        width: 45px;

        height: 30px;

    }

    .input-matrice-svijetli {

        width: 45px;

        height: 30px;

    }

    .duzina, .sirina {

        width: 50px;

        height: 30px;

    }

}

@media only screen and (max-width: 1170px) {

    .dugmici {

        font-size: x-small;

    }

    .operacije {

        font-size: x-small;

    }

    .input-matrice {

        width: 30px;

        height: 25px;

    }

    .input-matrice-svijetli {

        width: 30px;

        height: 25px;

    }

    .duzina, .sirina {

        width: 50px;

        height: 30px;

    }

}

@media only screen and (max-width: 950px) {

    .dugmici {

        font-size: xx-small;

    }

    .operacije {

        font-size: xx-small;

    }

    .input-matrice {

        width: 25px;

        height: 20px;

    }

    .input-matrice-svijetli {

        width: 25px;

        height: 20px;

    }

    .duzina, .sirina {

        width: 50px;

        height: 30px;

    }

}

@media only screen and (max-width: 800px) {

    #header {

        font-size: 28px;

    }

    .input-matrice {

        width: 35px;

        height: 20px;

        font-size: small;

    }

    .input-matrice-svijetli {

        width: 35px;

        height: 20px;

        font-size: small;

    }

    .flex {

        font-size: small;

    }

    .strelica {

        width: 15px;

        height: 15px;

    }

    #mjesto-za-matrice {

        flex-direction: column;

    }

    .duzina, .sirina {

        width: 50px;

        height: 30px;

    }

}

# **ZAKLJUČAK**

Kao što sam već spomenuo cilj ove aplikacije jeste bio da učenje i vježbu matričnih zadataka učini što interaktivnijim. Za razliku od raznih zbirki i knjiga, koje učeniku mogu izgledati neprivlačne i dosadne, web aplikacije pružaju jedan sasvim novi i drugačiji vid učenja koji je za savremenog učenika mnogo više zanimljiv.

Velika prednost ove aplikacije i njoj sličnim jeste da korisnik može da sebi zada neograničen broj zadataka i da u vrlo kratkom roku provjeri njihovu tačnost što inače ne bi bilo moguće.

Zbog brzine i efikasnosti aplikacija za rješavanje matematičkih problema smatram da će te aplikacije u budućnosti imati mnogo veću primjenu i da će učenicima biti mnogo lakše naučiti i savladati određeno gradivo, te da će im učenje gradiva biti lakše i zabavnije bez nepotrebnog stresa i frustracija.

# **LITERATURA**

* Dakić B., Elezović N. (2014). Matematika 3 udžbenik i zbirka zadataka za 3. razred prirodoslovno – matematičke gimnazije, Zagreb: Element
* Matrice na adresi:

<https://mapmf.pmfst.unist.hr/~skresic/LAMR/Folije/Matrice.pdf>

* Matrice na adresi:

<https://shkolyariki.ru/hr/klassnomu/chto-takoe-transpoziciya-matricy-kak-vyglyadit-transponirovannaya-matrica-ee.html>

* Nastavni materijal sa redovne nastave

**SADRŽAJ**

[**UVOD** 1](#_Toc103636739)

[**1.** **MATRICE** 2](#_Toc103636740)

[**1.1.** **SABIRANJE I ODUZIMANJE MATRICA** 3](#_Toc103636741)

[**1.2.** **MNOŽENJE MATRICA** 4](#_Toc103636742)

[**1.3.** **TRANSPONIRANA MATRICA** 5](#_Toc103636743)

[**1.4.** **DETERMINANTA MATRICE** 6](#_Toc103636744)

[**2.** **IZRADA WEB APLIKACIJE** 7](#_Toc103636745)

[**ZAKLJUČAK** 40](#_Toc103636746)

[**LITERATURA** 41](#_Toc103636747)

Datum predaje rada: \_\_\_\_\_\_\_\_\_

Komisija:

Predsjednik: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ispitivač: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Član: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Mišljenje o radu:

Datum odbrane rada: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ocjena o radu: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (\_\_\_\_)