

WEB PROGRAMIRANJE 1

Uroš Bošković 140/22

Crystal Touch

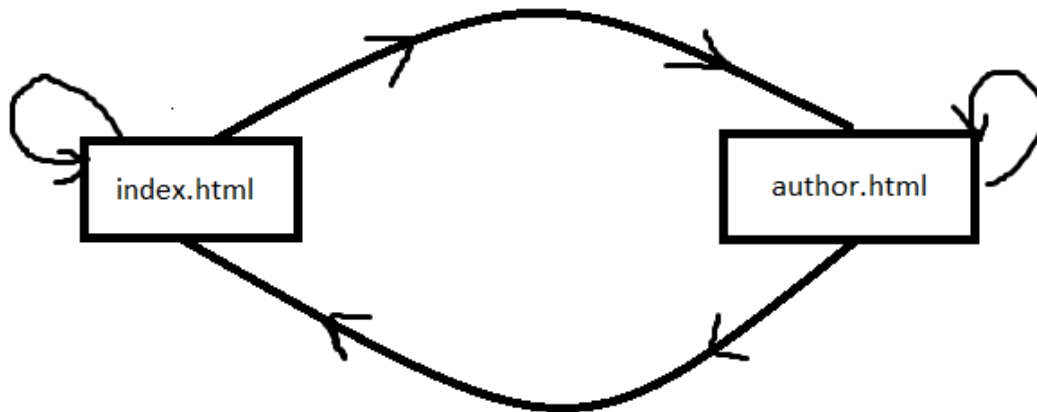
Korisceno: HTML, CSS, JavaScript, JQuery

Okruzenje: Visual Studio Code

Funkcionalnosti:

1. Dinamički ispisan carousel koji vrti automatski 3 slike smešten u header.
2. Dinamički ispisani divovi, slike i informacije proizvoda.
3. "Show more" button koji treba da doda nova 4 diva sponzora, kod koji medjutim meni na live serveru radi ali na hostovanom sajtu ne. Nakon pritiska on menja text u "Show less" i zajedno sa tim funkcionalnost a to je da izbriše novonastale blokove te se vraća sve u normalu.
4. Slider carousel koji drži 5 blokova dok je vidljiv samo jedan tj. jedna slika, na strelice ispod slike se vrti unapred ili unazad, a pritom kad se dodje do kraja (prve ili poslednje slike) on će da prikaže ponovo sliku sa početka ili sa kraja u zavisnosti dokle je došao tj. u kom je smeru išao.
5. Validacija forme, polja Ime i Prezime moraju početi sa velikim slovom a ostala su mala, Mail mora sadržati @ u sebi, Phone mora početi sa +3816 pa zatim neki od brojeva iz intervala 0 – 4 i narednih 6 ili 7 cifara mogu biti bilo koje, Genders mora biti selektovan jedan radio button, DDL mora biti selektovano bilo koje polje a da nije ono prvo početno.
6. JQuery animacija prilikom pritiska na link i odlaska na datu lokaciju na istoj web strani, primenjena u navigaciji, footeru i buttonu koji se nalazi na početku učitavanja stranice.

Šematski prikaz

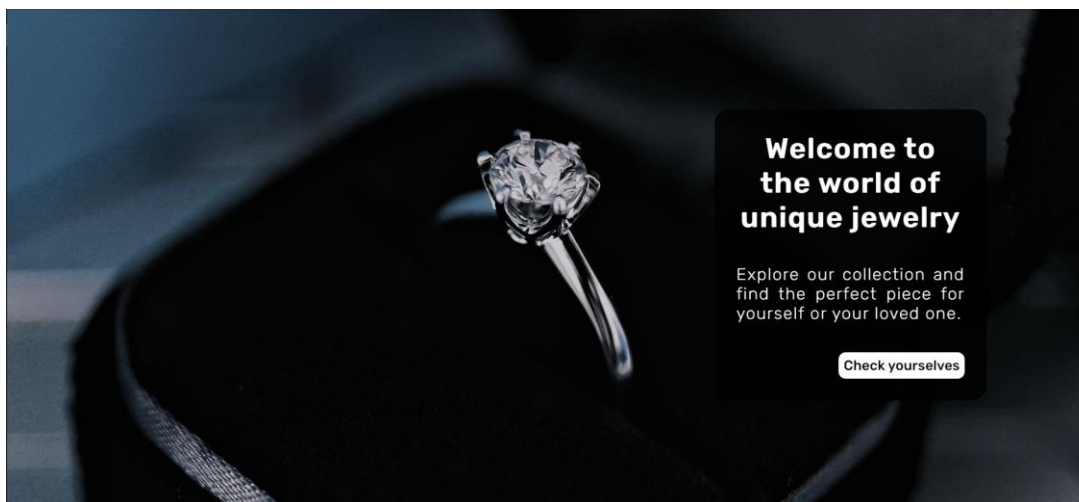
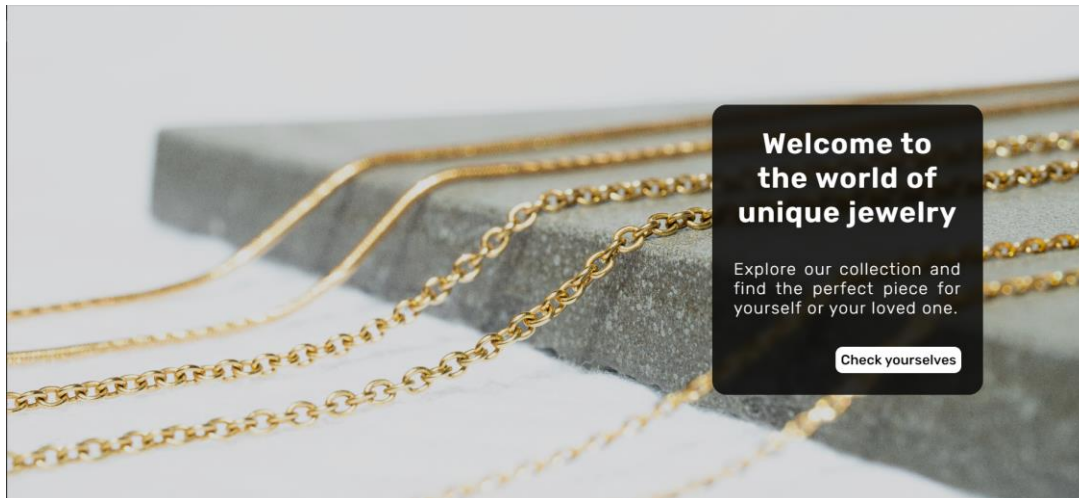


Mapa sajta

```
sitemap.xml
1  <?xml version="1.0" encoding="UTF-8"?>
2  <urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">
3      <url>
4          <loc>https://urosboskovic22.github.io/wp1/</loc>
5          <lastmod>2023-12-10</lastmod>
6          <changefreq>daily</changefreq>
7          <priority>1</priority>
8      </url>
9      <url>
10         <loc>https://urosboskovic22.github.io/wp1/author.html</loc>
11         <lastmod>2023-12-10</lastmod>
12         <changefreq>monthly</changefreq>
13         <priority>0.7</priority>
14     </url>
15 </urlset>
16
```

Funk. 1: Automatski carousel

Menjaju se 3 slike



JS:

```
1 // automatic slider
2 document.addEventListener('DOMContentLoaded', function() {
3     var header = document.querySelector('header');
4     var images = ['assets/img/lancic.jpg', 'assets/img/prsten8.jpg', 'assets/img/pozadina.jpg'];
5     let trntIndex = 0;
6
7     function nextImage() {
8         trntIndex = (trntIndex + 1) % images.length;
9         header.style.backgroundImage = `url('${images[trntIndex]}')`;
10    }
11
12    setInterval(() => {
13        header.classList.add('transition');
14        nextImage();
15
16        setTimeout(() => {
17            header.classList.remove('transition');
18        }, 1000);
19    }, 2000);
20 });
```

Funk. 2: Dinamički ispisani divovi, slike i informacije proizvoda.

U JS dinamički ispisivanje 9 divova čije sam putanje, imena i cene ubacio u niz kako bih posle kroz petlju primenio tamo gde je potrebno, putanje u atribut, imena i cene kao novi h3 i h2 elementi, a pritom imena idu ujedno i u alt atribut slika. Tri po jednom redu će se stilizovati.

Od HTML samo:

```
<div class="naslov" id="jewelryPath">
  <h1>Jewelry</h1>
</div>
<div class="jewelry"></div>
```

JS:

```
// dinamicki ispis proizvoda
document.addEventListener("DOMContentLoaded", function () {
  var jewelryDiv = document.querySelector(".jewelry");

  var jewelryImages = [
    "ogrlica7.jpeg", "ogrlica8.jpeg", "ogrlica11.jpeg",
    "narukvica2.jpg", "narukvica1.jpg", "narukvica3.jpg",
    "prsten13.jpg", "prsten4.jpg", "prsten10.jpg"
  ];
  var jewelryNames = [
    "Diamond necklace", "Diamond necklace", "Diamond necklace",
    "Silver bracelet", "Diamond bracelet", "Diamond bracelet",
    "Diamond ring", "Diamond ring", "Diamond ring"
  ];
  var jewelryPrices = [1050, 1150, 1000, 130, 450, 600, 120, 100, 110];

  for (let i = 0; i < jewelryImages.length; i++) {
    var jewelDiv = document.createElement("div");
    jewelDiv.className = "jewel";

    var img = document.createElement("img");
    img.src = "assets/img/" + jewelryImages[i];
    img.alt = jewelryNames[i];
    img.className = "imgRadius";

    var jewelInfoDiv = document.createElement("div");
    jewelInfoDiv.className = "jewelInfo";
    jewelInfoDiv.classList.add("jewelInfoRadius");
```

```

var name = document.createElement("h3");
name.textContent = jewelryNames[i];
name.className = "jewelInfoStyle";

var price = document.createElement("h2");
price.textContent = jewelryPrices[i] + "$";
price.className = "jewelInfoStyle";

jewelInfoDiv.appendChild(name);
jewelInfoDiv.appendChild(price);

jewelDiv.appendChild(img);
jewelDiv.appendChild(jewelInfoDiv);

jewelryDiv.appendChild(jewelDiv);
}
});

```



Diamond necklace 1050\$



Diamond necklace 1150\$



Diamond necklace 1000\$



Silver bracelet 130\$



Diamond bracelet 450\$



Diamond bracelet 600\$



Diamond ring

120\$



Diamond ring

100\$



Diamond ring

110\$

Funk. 3: "Show more" i "Show less"

Ovaj kod meni na hostovanom sajtu neće uopšte da prikaže, ni HTML ni JavaScript jedino button iz HTMLA hoće. U HTML napravio sam 4 diva u jednom redu koji će imati logoe nekih brendova. Button je ispod njih, na prvi klik se ponaša i zove se "Show More", dinamički pravi i dodaje 4 nova diva sa novim logoima nekih brendova, potom on postaje "Show less" dugme i na klik on vraća sve u prvobitno stanje tj. briše novonapravljene divove.

Inicijalno ↓



See More

Posle prvog klika ↓



Show Less

Posle drugog klika ↓



See More

HTML kod:

```
<div class="sponsors">
  <div class="sponsorBox">
    
  </div>
  <div class="sponsorBox">
    
  </div>
  <div class="sponsorBox">
    
  </div>
  <div class="sponsorBox">
    
  </div>
</div>

<button id="seeMoreBtn">See More</button>
```

JS kod:

```
// sponsor button show more
document.addEventListener("DOMContentLoaded", function () {
  var sponsorsDiv = document.querySelector('.sponsors');
  var newSponsorImages = [
    'assets/img/versace.jpeg',
    'assets/img/gucci2.jpeg',
    'assets/img/lv.jpeg',
    'assets/img/lacoste.jpeg'
  ];
  var seeMoreBtn = document.getElementById('seeMoreBtn');
  var isExpanded = false;

  seeMoreBtn.addEventListener('click', function () {
    isExpanded = !isExpanded;

    if (isExpanded) {
      showNewSponsors();
      seeMoreBtn.textContent = 'Show Less';
    } else {
      hideNewSponsors();
      seeMoreBtn.textContent = 'Show More';
    }
  });
});
```

```

function showNewSponsors() {
    newSponsorImages.forEach(function (imagePath) {
        var newSponsorDiv = document.createElement('div');
        newSponsorDiv.className = 'sponsorBox';

        var newImage = document.createElement('img');
        newImage.src = imagePath;
        newImage.alt = imagePath + 'logo';
        newImage.type = 'image/jpeg';

        newSponsorDiv.appendChild(newImage);
        sponsorsDiv.appendChild(newSponsorDiv);

        setTimeout(function () {
            newSponsorDiv.style.opacity = 1;
        }, 0);
    });
}

function hideNewSponsors() {
    var newSponsorDivs = document.querySelectorAll('.sponsors .sponsorBox:nth-child(n+5)');
    newSponsorDivs.forEach(function (box) {
        box.style.opacity = 0;

        setTimeout(function () {
            box.remove();
        }, 300);
    });
}
});

```

Funk. 4: Slider carousel

Koji sadrži 5 slika a vidljiva je jedna samo, strelice ispod te slike kako bi moglo da se šalta levo ili desno, ako se dodje do kraja vraća se slika sa početka ili kraja u zavisnosti u kome se pravcu menjala slika. Dinamički su ispisane sve slike, sem početne koja je u HTML postavljena kako bi inicijalno stala ta prva i bila vidljiva. Njihove putanje su upisane u niz kako bi se posle mogle implementirati kroz petlji u src atribut

HTML kod:

```
<div class="stones">
  <div class="stoneSlider">
    <div class="stoneImgBox">
      
    </div>
    <!-- <div class="stoneButtonHolder"> -->
    <button class="stoneBtn" onclick="prev()">&lt;</button>
    <button class="stoneBtn" onclick="next()">&gt;</button>
    <!-- </div> -->
  </div>
</div>
```

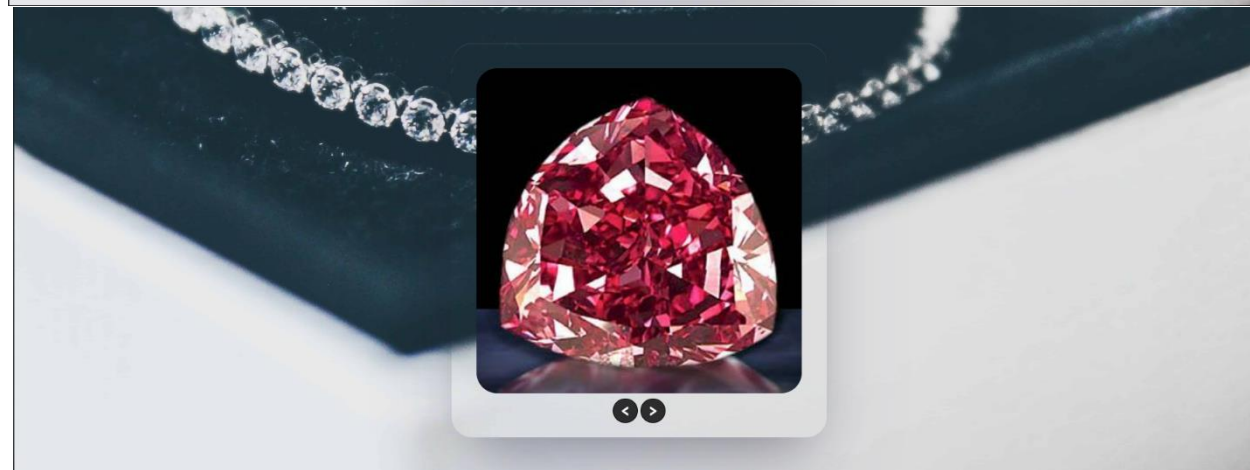
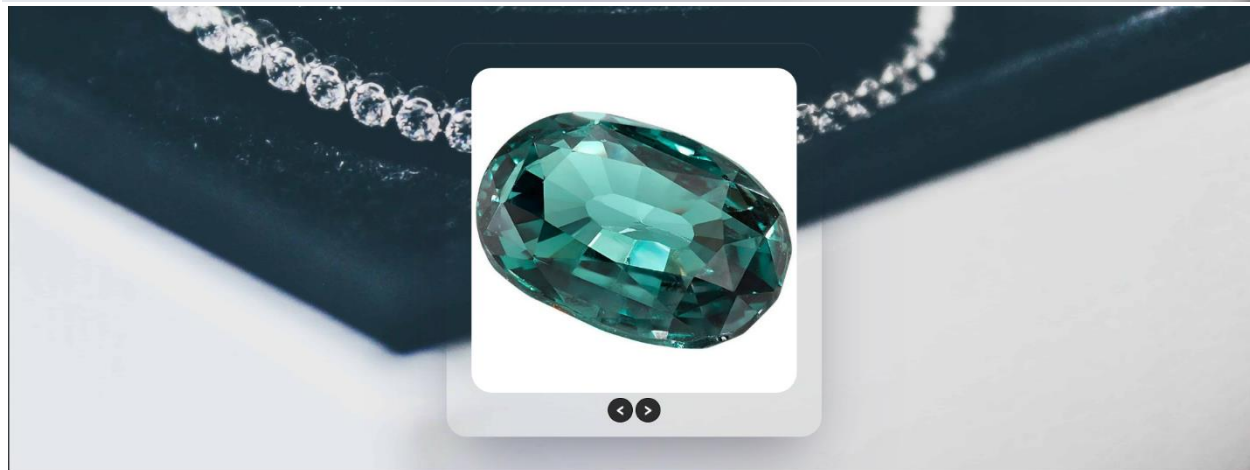
JS kod:

```
// stone slide
var sliderImg = document.querySelector(".stoneSliderImg");
var images = [ 'assets/img/kamen9.jpeg',
               'assets/img/kamen17.jpeg',
               'assets/img/kamen10.jpeg',
               'assets/img/kamen18.jpeg',
               'assets/img/kamen14.jpeg'
             ];
let j = 0;

function prev(){
  if(j <= 0) j = images.length;
  j--;
  return setImg();
}

function next(){
  if( j == images.length - 1) j = -1;
  j++;
  return setImg();
}

function setImg(){
  return sliderImg.setAttribute('src', images[j]);
}
```





Funk. 5: Validacija Forme

Polja Ime i Prezime moraju početi sa velikim slovom a ostala su mala, Mail mora sadržati @ u sebi, Phone mora početi sa +3816 pa zatim neki od brojeva iz intervala 0 – 4 i narednih 6 ili 7 cifara mogu biti bilo koje, Genders mora biti selektovan jedan radio button, DDL mora biti selektovano bilo koje polje a da nije ono prvo početno. Ako neko polje nije popunjeno kako treba ono dobije klasu .fault koja menja boju texta u crvenu, po pravilnom unosu podatka u polje ono gubi istu klasu. Ne može da se izvrši submit dugme sve dok ne budu sva pravila unosa ispunjena.

HTML

```
<div class="formHolder">
  <form action="" id="form">
    <div class="divNameSurname">
      <div class="divName">
        <label for="name">Name</label>
        <input type="text" name="name" id="name" placeholder="Uros"/>
      </div>
      <div class="divSurname">
        <label for="surname">Last Name</label>
        <input type="text" name="surname" id="surname" placeholder="Boskovic"/>
      </div>
    </div>
    <div class="divMailPhone">
      <div class="divMail">
        <label for="mail">Email</label>
        <input type="mail" name="mail" id="mail" placeholder="uros.boskovic.140.22@gmail.com"/>
      </div>
      <div class="divPhone">
        <label for="phone">Phone</label>
        <input type="text" name="phone" id="phone" placeholder="zovi me - RR"/>
      </div>
    </div>
  </form>
</div>
```

```
    <div class="divGendersPurpose">
      <div class="divGenders">
        <label for="male">Male</label>
        <input type="radio" name="genders" value="male" id="male"/>
        <label for="female">Female</label>
        <input type="radio" name="genders" value="female" id="female"/>
      </div>
      <div class="divPurpose">
        <label for="purpose"></label>
        <select name="purpose" id="purpose">
          <option value="null">Choose</option>
          <option value="customer">Customer</option>
          <option value="job">Applying for a job</option>
          <option value="business">Businesswise</option>
        </select>
      </div>
    </div>
    <div class="divTextArea">
      <label for="textArea">Anything we need to know?</label>
      <textarea name="area" id="textArea" cols="30" rows="10"></textarea>
    </div>
    <button type="submit" method="get" action="#form" class="submitBtn">Submit</button>
  </form>
</div>
```

JS

```
// form validation
var btn = document.getElementById("submitBtn");

function proverPolja(regex, polje) {
    if (!regex.test(polje.value)) {
        polje.classList.add('fault');
        return false;
    } else {
        polje.classList.remove('fault');
        return true;
    }
}

function proverRadio(radioButtons) {
    for (const radioBtn of radioButtons) {
        if (radioBtn.checked) {
            return true;
        }
    }
    return false;
}

function proverSelect(select) {
    return select.value !== 'null';
}

let regIme = /^[A-Z][a-z]*$/;
let regPrezime = /^[A-Z][a-z]*$/;
let regEmail = /^[^s@]+@[^s@]+\.[^s@]+$/;
let regTelefon = /\+3816[0-4][0-9]{6,7}$/;
```

```
let ime = document.getElementById('name');
document.getElementById('name').addEventListener("blur", function () {
    proverPolja(regIme, ime);
});

let prezime = document.getElementById('surname');
document.getElementById('surname').addEventListener("blur", function () {
    proverPolja(regPrezime, prezime);
});

let mail = document.getElementById('mail');
document.getElementById('mail').addEventListener("blur", function () {
    proverPolja(regEmail, mail);
});

let brTel = document.getElementById('phone');
document.getElementById('phone').addEventListener("blur", function () {
    proverPolja(regTelefon, brTel);
});

let genderMale = document.getElementById('male');
let genderFemale = document.getElementById('female');
let genderRadioButtons = [genderMale, genderFemale];

document.getElementById('form').addEventListener("change", function () {
    proverRadio(genderRadioButtons);
});

let purposeSelect = document.getElementById('purpose');
document.getElementById('form').addEventListener("change", function () {
    proverSelect(purposeSelect);
});
```



```

function provera() {
  const isImeValid = proverPolja(regIme, ime);
  const isPrezimeValid = proverPolja(regPrezime, prezime);
  const isEmailValid = proverPolja(regExEmail, mail);
  const isTelValid = proverPolja(regTelefon, brTel);
  const isGenderValid = proverRadio(genderRadioButtons);
  const isPurposeValid = proverSelect(purposeSelect);

  console.log("Ime validno:", isImeValid);
  console.log("Prezime validno:", isPrezimeValid);
  console.log("Email validno:", isEmailValid);
  console.log("Telefon validan:", isTelValid);
  console.log("Radio button validan:", isGenderValid);
  console.log("Select lista validna:", isPurposeValid);

  if (!(isImeValid && isPrezimeValid && isEmailValid && isTelValid && isGenderValid && isPurposeValid)) {
    console.log("Forma nije ispravna. Molimo ispravite greške.");
    return false;
  }
  return true;
}

document.getElementById('form').addEventListener('submit', function (event) {
  event.preventDefault();

  if (provera()) {
    this.submit();
  } else {
    console.log("Forma nije ispravna. Molimo ispravite greške.");
  }
});

```

Ovako izgleda kad se primeni .fault klasa.

Name <input type="text" value="lose upisano"/>	Last Name <input type="text" value="lose upisano"/>
Email <input type="text" value="lose upisano"/>	Phone <input type="text" value="lose upisano"/>