

# PROJEKTNI ZADATAK

## *MobilShop website*



Predmet: Web Programiranje 1

Student: Nikola Gajić

Predmetni Profesor: Nenad Kojić

Indeks Studenta: 160/23

Predmetni Asistenti: Luka Lukić, Milena Čančarević

<https://ngbusiness.github.io/MobilShop/index.html>

# Sadržaj

■	Uvod .....	3
-	Korišćeni Programske jezici.....	3
-	Funkcionalnosti.....	3
■	2. Organizacija .....	20
-	Organizaciona šema .....	20
-	Mapa sajta .....	20
-	Slike stranica i opisi funkcionalnosti .....	21
	Kodovi .....	22
-	Index strana (index.html) .....	22
-	Autor strana (author.html) .....	26
-	Javascript (main.js).....	27
	CSS(style.css) .....	51

# Uvod

- Korišćeni Programske jezici
  - Za izradu sajta korišćeni su sledeći programski jezici i tehnologije: HTML5 i CSS3 za strukturu i stil stranica, JavaScript za dinamičke funkcionalnosti i validaciju forme, biblioteka jQuery za jednostavnije upravljanje DOM elementima, kao i Font Awesome za ikonice. Sajt je razvijen u Visual Studio Code okruženju i testiran na modernim web pregledačima.

## Funkcionalnosti

### 1. Header podaci

```
- const siteData = {
-   // Header podaci
-   header: {
-     logoText: "MobilShop",
-     logoIcon: "💻",
-     phone: "+381 11 123 4567",
-     email: "info@mobilshop.rs",
-     workHours: "Pon-Sub: 09-20h",
-   },
}
```

- Ovaj objekat sadrži **sve informacije za prikaz u headeru sajta**.
- `logoText` – tekstualni logo sajta, HTML span tag je korišćen da deo teksta može imati drugačiji stil (npr. druga boja).
- `logoIcon` – emoji ili ikonica koja ide uz logo.
- `phone` i `email` – kontakt informacije koje se prikazuju u headeru.
- `workHours` – radno vreme koje se takođe dinamički prikazuje.

Ovim pristupom, sve informacije za header su centralizovane, i lako se menjaju bez potrebe da se menja HTML.

### 2. Navigacija

```
// Navigacija
navigation: [
  { name: "Početna", href: "#hero", active: true },
  { name: "Telefoni", href: "#featured-section" },
  { name: "Brendovi", href: "#brands-section" },
  { name: "Kontakt", href: "#contact-section" },
  { name: "O nama", href: "#why-us-section" },
  { name: "Autor", href: "autor.html" },
],
```

- Ovaj niz objekata sadrži **linkove za navigacioni meni**.
- name – tekst koji se prikazuje na meniju.
- href – destinacija linka, može biti **id sekcije** na istoj stranici (#hero) ili link ka drugoj stranici (autor.html).
- active – označava koja je stavka aktivna po defaultu (npr. Početna).

Kod se koristi za **dinamičko generisanje menija**, tako da dodavanje novih stranica ili sekcija ne zahteva ručno menjanje HTML-a.

### 3. Hero sekcija

```
// Hero sekcija
hero: {
  title: "Pronađite Vaš <span>Savršeni</span> Telefon",
  subtitle: "Najnoviji modeli vodećih proizvođača po najboljim cenama.
Besplatna dostava i 2 godine garancije.",
  badge: "NOVO 2024",
  buttons: [
    { text: "Pogledaj ponudu", class: "btn-primary", href: "#featured-section" },
    { text: "Kontaktiraj nas", class: "btn-secondary", href: "#contact-section" },
  ],
},
```

- Objekat hero sadrži **sve podatke koji se prikazuju u hero sekciji sajta** (gornji deo sajta, prvi vizuelni utisak).
- title – glavni naslov sa opcionalnim HTML <span> tagom za naglašavanje dela teksta.
- subtitle – kratak opis/poruka koja se prikazuje ispod naslova.
- badge – oznaka ili etiketa (npr. NOVO, POPUST).
- buttons – niz dugmadi koja se prikazuju u hero sekciji. Svako dugme ima:
  - text – tekst dugmeta,
  - class – CSS klasa za stil dugmeta (btn-primary, btn-secondary),
  - href – destinacija klika (sekcija na istoj stranici).

Ovako definisana hero sekcija omogućava **brze promene naslova, podnaslova ili dugmadi** bez diranja HTML-a.

### 4. Prikaz proizvoda

```
// Proizvodi
products: [
  {
    id: 1,
    brand: "Apple",
```

```
name: "iPhone 15 Pro Max",
specs: "256GB | Titanijum | A17 Pro",
price: 189990,
oldPrice: 209990,
badge: "sale",
image: "assets/img/iphone-15-pro-max.png",
fullSpecs: {
    Ekran: '6.7" Super Retina XDR',
    Procesor: "A17 Pro",
    RAM: "8GB",
    Memorija: "256GB",
    Kamera: "48MP + 12MP + 12MP",
    Baterija: "4422mAh",
},
},
{
id: 2,
brand: "Samsung",
name: "Galaxy S24 Ultra",
specs: "512GB | Titanijum | SD 8 Gen 3",
price: 174990,
oldPrice: null,
badge: "new",
image: "assets/img/s24-ultra.jpg",
fullSpecs: {
    Ekran: '6.8" Dynamic AMOLED 2X',
    Procesor: "Snapdragon 8 Gen 3",
    RAM: "12GB",
    Memorija: "512GB",
    Kamera: "200MP + 50MP + 12MP + 10MP",
    Baterija: "5000mAh",
},
},
{
id: 3,
brand: "Xiaomi",
name: "14 Ultra",
specs: "512GB | Crna | SD 8 Gen 3",
price: 149990,
oldPrice: 169990,
badge: "popular",
image: "assets/img/xiaomi-14-ultra.png",
fullSpecs: {
    Ekran: '6.73" LTPO AMOLED',
    Procesor: "Snapdragon 8 Gen 3",
}
```

```
        RAM: "16GB",
        Memorija: "512GB",
        Kamera: "50MP + 50MP + 50MP",
        Baterija: "5000mAh",
    },
},
{
    id: 4,
    brand: "OnePlus",
    name: "12",
    specs: "256GB | Zelena | SD 8 Gen 3",
    price: 99990,
    oldPrice: null,
    badge: "new",
    image: "assets/img/oneplus-12-green.png",
    fullSpecs: {
        Ekran: '6.82" LTPO AMOLED',
        Procesor: "Snapdragon 8 Gen 3",
        RAM: "12GB",
        Memorija: "256GB",
        Kamera: "50MP + 48MP + 64MP",
        Baterija: "5400mAh",
    },
},
{
    id: 5,
    brand: "Google",
    name: "Pixel 8 Pro",
    specs: "256GB | Obsidian | Tensor G3",
    price: 119990,
    oldPrice: 134990,
    badge: "sale",
    image: "assets/img/google-pixel-8-pro.png",
    fullSpecs: {
        Ekran: '6.7" LTPO OLED',
        Procesor: "Google Tensor G3",
        RAM: "12GB",
        Memorija: "256GB",
        Kamera: "50MP + 48MP + 48MP",
        Baterija: "5050mAh",
    },
},
{
    id: 6,
    brand: "Samsung",
```

```

name: "Galaxy Z Fold 5",
specs: "512GB | Crna | Preklopni",
price: 219990,
oldPrice: null,
badge: "popular",
image: "assets/img/Samsung-Galaxy-Z-Fold-5.webp",
fullSpecs: [
    Ekran: '7.6" + 6.2" Dynamic AMOLED',
    Procesor: "Snapdragon 8 Gen 2",
    RAM: "12GB",
    Memorija: "512GB",
    Kamera: "50MP + 12MP + 10MP",
    Baterija: "4400mAh",
],
],
],
],

```

- Lista proizvoda se čita iz `siteData.products`.
- Za svaki proizvod se dinamički kreira kartica sa nazivom, slikom, cenom i dugmadima.

## 5. Dodavanje u korpu(toast notifikacija)

```

- // ===== FUNKCIJA ZA PRIKAZ TOAST PORUKE =====
- function showToast(message, duration = 3000) {
-     let toast = document.getElementById("toast")
-
-     // Ako toast ne postoji, kreiraj ga i ubaci u body
-     if (!toast) {
-         toast = document.createElement("div")
-         toast.id = "toast"
-         toast.classList.add("hidden") // Početno sakriven
-         document.body.appendChild(toast)
-     }
-
-     toast.textContent = message
-     toast.classList.remove("hidden")
-     toast.classList.add("show")
-
-     // Automatski sakrij nakon duration milisekundi
-     setTimeout(() => {
-         toast.classList.remove("show")
-         toast.classList.add("hidden")
-     }, duration)
- }

```

- Dugme „Dodaj u korpu“ prikazuje toast notifikaciju sa imenom proizvoda (showToast).
- Toast se animira sa show i hidden klasama, a zatim automatski nestaje posle 3 sekunde.

## 6. Render funkcije za sve sekcije sajta

```

- // Kreiranje headera
- function renderHeader() {
-   const header = document.getElementById("header")
-   const data = siteData.header

-   header.innerHTML =
-     `<div class="container">
-       <div class="header-content">
-         <div class="logo">
-           <div class="logo-icon">${data.logoIcon}</div>
-           <div class="logo-text">${data.logoText}</div>
-         </div>
-         <div class="header-info">
-           <div class="header-info-item">
-             <span>📞</span>
-             <span>${data.phone}</span>
-           </div>
-           <div class="header-info-item">
-             <span>✉</span>
-             <span>${data.email}</span>
-           </div>
-           <div class="header-info-item">
-             <span>⌚</span>
-             <span>${data.workHours}</span>
-           </div>
-         </div>
-       </div>
-     `

- }

- // Kreiranje navigacije
- function renderNavigation() {
-   const nav = document.getElementById("main-nav")
-   const navItems = siteData.navigation

-   let menuHTML = ""
-   navItems.forEach((item) => {

```

```

-     const activeClass = item.active ? "active" : ""
-     menuHTML += `<li><a href="${item.href}" class="${activeClass}">${item.name}</a></li>`
-   })
-
-   nav.innerHTML =
-     <div class="nav-container">
-       <button class="mobile-menu-toggle" aria-label="Otvořit meni">☰</button>
-       <ul class="nav-menu">
-         ${menuHTML}
-       </ul>
-       <div class="nav-search">
-         <input type="search" placeholder="Pretraži telefone..." id="search-input">
-       </div>
-     </div>
-
-   // Dodaj event listener za mobilni meni
-   setupMobileMenu()
- }
-
// Kreiranje hero sekcije
function renderHero() {
  const hero = document.getElementById("hero")
  const data = siteData.hero
-
  let buttonsHTML = ""
  data.buttons.forEach((btn) => {
    buttonsHTML += `<a href="${btn.href}" class="btn ${btn.class}">${btn.text}</a>`
  })
-
  hero.innerHTML =
    <div class="container">
      <div class="hero-content">
        <div class="hero-text animate-in">
          <h1>${data.title}</h1>
          <p>${data.subtitle}</p>
          <div class="hero-buttons">
            ${buttonsHTML}
          </div>
        </div>
      </div>
      <div class="hero-image">

```

```

        <div class="hero-phone-display">
            <div class="phone-mockup">
                <div class="phone-screen">📱 </div>
            </div>
            <div class="hero-badge
pulse">${data.badge}</div>
            </div>
        </div>
    </div>
    .
}

// Kreiranje proizvoda
function renderProducts() {
    const container = document.getElementById("featured-products")
    const titleEl = document.getElementById("featured-title")
    const subtitleEl = document.getElementById("featured-subtitle")

    titleEl.textContent = siteData.sections.featured.title
    subtitleEl.textContent = siteData.sections.featured.subtitle
    subtitleEl.className = "section-subtitle text-center"
    titleEl.className = "text-center"

    let productsHTML = ""

    siteData.products.forEach((product, index) => {
        const badgeClass = product.badge === "sale" ? "badge-sale" :
product.badge === "new" ? "badge-new" : "badge-popular"
        const badgeText = product.badge === "sale" ? "Akcija" :
product.badge === "new" ? "Novo" : "Popularno"

        const oldPriceHTML = product.oldPrice ? `<span class="price-
old">${formatPrice(product.oldPrice)}</span>` : ""

        productsHTML += `
            <div class="product-card animate-in" style="animation-
delay: ${index * 0.1}s" data-product-id="${product.id}">
                <span class="product-badge
${badgeClass}">${badgeText}</span>
                <div class="product-image">
                    
                </div>
                <div class="product-info">
                    <div class="product-brand">${product.brand}</div>

```

```

        <h3 class="product-name">${product.name}</h3>
        <p class="product-specs">${product.specs}</p>
        <div class="product-price">
            <span class="price-current">${formatPrice(product.price)}</span>
            ${oldPriceHTML}
        </div>
        <div class="product-actions">
            <button class="btn btn-primary btn-details" data-id="${product.id}">Detalji</button>
            <button class="btn btn-accent btn-cart" data-id="${product.id}">Korpa</button>
        </div>
    </div>
}

container.innerHTML = productsHTML

// Dodaj event listenere za proizvode
setupProductEvents()
}

// Kreiranje brendova
function renderBrands() {
    const container = document.getElementById("brands-container")
    const titleEl = document.getElementById("brands-title")

    titleEl.textContent = siteData.sections.brands.title
    titleEl.className = "text-center"

    let brandsHTML = ""

    siteData.brands.forEach((brand, index) => {
        brandsHTML += `
            <div class="brand-card animate-in" style="animation-delay: ${index * 0.1}s" data-brand="${brand.name}">
                <div class="brand-icon">${brand.icon}</div>
                <div class="brand-name">${brand.name}</div>
                <div class="brand-count">${brand.count} modela</div>
            </div>
        `
    })
}

```

```

- container.innerHTML = brandsHTML

-
- // Dodaj event listenere za brendove
- setupBrandFilter()
}

-
// Kreiranje prednosti
function renderFeatures() {
    const container = document.getElementById("why-us-container")
    const titleEl = document.getElementById("why-us-title")

    titleEl.textContent = siteData.sections.whyUs.title
    titleEl.className = "text-center"

    let featuresHTML = ""

    siteData.features.forEach((feature, index) => {
        featuresHTML += `
            <div class="feature-card animate-in" style="animation-delay: ${index * 0.1}s">
                <div class="feature-icon">${feature.icon}</div>
                <h3 class="feature-title">${feature.title}</h3>
                <p class="feature-description">${feature.description}</p>
            </div>
        `

    })

    container.innerHTML = featuresHTML
}

-
// Kreiranje footer-a
function renderFooter() {
    const footer = document.getElementById("footer")
    const data = siteData.footer

    let quickLinksHTML = ""
    data.quickLinks.forEach((link) => {
        quickLinksHTML += `<li><a href="${link.href}">${link.name}</a></li>`
    })

    let supportLinksHTML = ""
    data.support.forEach((link) => {

```

```

-     supportLinksHTML += `<li><a
- href="${link.href}">${link.name}</a></li>`
-   })

-   let socialHTML = ""
-   data.social.forEach((icon) => {
-     socialHTML += `<a href="#" class="social-link">${icon}</a>`
-   })

-   footer.innerHTML =
-     `<div class="container">
-       <div class="footer-content">
-         <div class="footer-section">
-           <h3>O Nama</h3>
-           <p>${data.about}</p>
-           <div class="social-links">
-             ${socialHTML}
-           </div>
-         </div>
-         <div class="footer-section">
-           <h3>Brzi Linkovi</h3>
-           <ul>${quickLinksHTML}</ul>
-         </div>
-         <div class="footer-section">
-           <h3>Podrška</h3>
-           <ul>${supportLinksHTML}</ul>
-         </div>
-         <div class="footer-section">
-           <h3>Kontakt</h3>
-           <p>📍 ${data.contact.address}</p>
-           <p>📞 ${data.contact.phone}</p>
-           <p>✉️ ${data.contact.email}</p>
-         </div>
-       </div>
-       <div class="footer-bottom">
-         ${data.copyright}
-       </div>
-     </div>
-   `
- }

```

Za kreiranje svih glavnih sekcija sajta koristi se niz funkcija koje dinamički generišu HTML sadržaj na osnovu podataka iz objekta siteData. Svaka funkcija odgovara jednoj sekciji:

1. **renderHeader()** – generiše logo, kontakt informacije i osnovne elemente headera.

2. **renderNavigation()** – kreira meni sa linkovima, aktivnim stavkama i pretragom, uključujući podršku za mobilni meni.
3. **renderHero()** – prikazuje hero sekciju sa naslovom, podnaslovom, bedžem i dugmadima.
4. **renderProducts()** – prikazuje listu proizvoda sa slikama, cenama, bedževima i dugmadima za detalje i dodavanje u korpu.
5. **renderBrands()** – prikazuje sekciju brendova sa ikonama i brojem modela, uz filter funkcionalnost.
6. **renderFeatures()** – prikazuje sekciju prednosti sajta sa ikonama, naslovima i opisima.
7. **renderFooter()** – kreira footer sa odeljcima: O nama, Brzi linkovi, Podrška, Kontakt i društvene mreže.

Sve funkcije koriste **siteData** za čuvanje i dohvatanje podataka, što omogućava jednostavno ažuriranje i održavanje sajta bez direktnog menjanja HTML-a.

## 7. Naslovi kontakt sekcije

```
function renderContactTitles() {
  const titleEl = document.getElementById("contact-title")
  const subtitleEl = document.getElementById("contact-subtitle")

  titleEl.textContent = siteData.sections.contact.title
  titleEl.className = "text-center"
  subtitleEl.textContent = siteData.sections.contact.subtitle
  subtitleEl.className = "section-subtitle text-center"
}
```

### **renderContactTitles()**

Ova funkcija postavlja naslov i podnaslov za kontakt sekciju koristeći podatke iz `siteData.sections.contact`. Naslovi se centriraju korišćenjem CSS klase, a funkcija omogućava dinamičko ažuriranje teksta bez direktnog menjanja HTML-a.

## 8. Validacija forme

```
// ===== VALIDACIJA FORME =====

function validateField(field, pattern, errorElementId) {
  const value = field.value.trim()
  const errorElement = document.getElementById(errorElementId)

  if (value === "") {
    field.classList.remove("success", "error")
    errorElement.textContent = ""
    return false
  }
```

```

    if (pattern.test(value)) {
        field.classList.remove("error")
        field.classList.add("success")
        errorElement.textContent = ""
        return true
    } else {
        field.classList.remove("success")
        field.classList.add("error")
        const fieldName = errorElementId.replace("-error", "")
        errorElement.textContent = errorMessages[fieldName]
        return false
    }
}

function setupFormValidation() {
    const form = document.getElementById("contact-form")
    const imeField = document.getElementById("ime")
    const emailField = document.getElementById("email")
    const telefonField = document.getElementById("telefon")
    const porukaField = document.getElementById("poruka")
    const usloviField = document.getElementById("uslovi")
    const budzetField = document.getElementById("budzet")
    const budzetOutput = document.getElementById("budzet-output")

    // Real-time validacija
    imeField.addEventListener("input", () => validateField(imeField, patterns.ime, "ime-error"))
    emailField.addEventListener("input", () => validateField(emailField, patterns.email, "email-error"))
    telefonField.addEventListener("input", () => validateField(telefonField, patterns.telefon, "telefon-error"))
    porukaField.addEventListener("input", () => validateField(porukaField, patterns.poruka, "poruka-error"))

    // Budzet slider
    budzetField.addEventListener("input", () => {
        budzetOutput.textContent = formatPrice(Number.parseInt(budzetField.value))
    })

    // Form submit
    form.addEventListener("submit", (e) => {
        e.preventDefault()

        // Validacija svih polja
    })
}

```

```

const isImeValid = validateField(imeField, patterns.ime, "ime-error")
const isEmailValid = validateField(emailField, patterns.email, "email-error")
const isTelefonValid = validateField(telefonField, patterns.telefon,
"telefon-error")

// Provera uslova
const usloviError = document.getElementById("uslovi-error")
let isUsloviValid = true
if (!usloviField.checked) {
    usloviError.textContent = errorMessages.uslovi
    isUsloviValid = false
} else {
    usloviError.textContent = ""
}

// Ako je sve validno
if (isImeValid && isEmailValid && isTelefonValid && isUsloviValid) {
    // Prikupi podatke
    const formData = {
        ime: imeField.value.trim(),
        email: emailField.value.trim(),
        telefon: telefonField.value.trim(),
        brend: document.getElementById("brend").value,
        budzet: budzetField.value,
        interesovanja:
Array.from(document.querySelectorAll('input[name="interesovanje"]:checked')).map(
            (cb) => cb.value,
        ),
        poruka: porukaField.value.trim(),
    }

    // Prikazi uspesnu poruku
    showFormSuccess(formData)
}
}
}

function showFormSuccess(data) {
    const form = document.getElementById("contact-form")
    const successDiv = document.getElementById("form-success")

    const brendText = data.brend ? data.brend.charAt(0).toUpperCase() +
data.brend.slice(1) : "Nije izabrano"
    const interesovanjaText = data.interesovanja.length > 0 ?
data.interesovanja.join(", ") : "Nije izabrano"

```

```

successDiv.innerHTML = `

<div class="form-success-icon">✓</div>
<h3>Hvala Vam, ${data.ime}!</h3>
<p>Vaš upit je uspešno poslat. Odgovorićemo vam u najkraćem roku na
${data.email}.</p>
<div class="submitted-data">
  <h4>Poslati podaci:</h4>
  <p><strong>Ime:</strong> ${data.ime}</p>
  <p><strong>Email:</strong> ${data.email}</p>
  <p><strong>Telefon:</strong> ${data.telefon}</p>
  <p><strong>Brend:</strong> ${brendText}</p>
  <p><strong>Budžet:</strong>
    ${formatPrice(Number.parseInt(data.budzet))}</p>
    <p><strong>Interesovanja:</strong> ${interesovanjaText}</p>
    <p><strong>Poruka:</strong> ${data.poruka} || "Nema poruke"</p>
  </div>
  <button class="btn btn-primary" onclick="resetForm()" style="margin-top:
1rem;">Pošalji novi upit</button>
`


  form.classList.add("hidden")
  successDiv.classList.remove("hidden")
}

function resetForm() {
  const form = document.getElementById("contact-form")
  const successDiv = document.getElementById("form-success")

  form.reset()
  form.querySelectorAll("input, textarea, select").forEach((field) => {
    field.classList.remove("success", "error")
  })
  document.getElementById("budzet-output").textContent = "100,000 RSD"

  successDiv.classList.add("hidden")
  form.classList.remove("hidden")
}

```

## 1. validateField funkcija

Ovo je glavna funkcija za validaciju pojedinačnih polja forme.

### Parametri:

- field – HTML element input/textarea koji validiramo.
- pattern – regularni izraz (RegExp) koji definiše pravila za validaciju.
- errorElementId – ID elementa gde će se prikazivati poruka o grešci.

### Logika:

1. Uzima vrednost iz polja i uklanja prazne prostore sa početka i kraja (trim()).
2. Ako je polje prazno:
  - Uklanja klase error i success.
  - Briše poruku greške.
  - Vraća false.
3. Ako vrednost zadovoljava regularni izraz (pattern.test(value)):
  - Dodaje klasu success, uklanja error.
  - Briše poruku greške.
  - Vraća true.
4. Ako vrednost **ne zadovoljava uslove**:
  - Dodaje klasu error, uklanja success.
  - Prikazuje odgovarajuću poruku iz errorMessages.
  - Vraća false.

### 2. setupFormValidation funkcija

Ova funkcija postavlja sve event listener-e i pravila za formu.

### Koraci:

1. Dohvata HTML elemente forme i polja (ime, email, telefon, poruka, uslovi, budzet).
2. Real-time validacija – dok korisnik kuca:

Isto za email, telefon, poruka.

### 3. Budžet slider:

- Prikazuje trenutno izabrani budžet u tekstualnom elementu (budget-output).

### 4. Submit forme (form.addEventListener("submit", ...)):

- Sprečava default ponašanje forme (e.preventDefault()).
- Validira sve polja i checkbox za prihvatanje uslova.
- Ako je sve validno, kreira objekat formData sa svim vrednostima.
- Poziva funkciju showFormSuccess(formData) da prikaže uspešnu poruku.

### 3. showFormSuccess funkcija

Prikazuje potvrđni blok sa podacima koje je korisnik poslao.

### **Koraci:**

1. Sakriva formu (form.classList.add("hidden")).
  2. Prikazuje div sa potvrdom (successDiv.classList.remove("hidden")).
  3. Formatira podatke korisnika:
    - o Brend i interesovanja se lepo prikazuju ili piše „Nije izabrano“ ako nisu odabrani.
    - o Budžet se formatira funkcijom formatPrice().
  4. Dodaje dugme „Pošalji novi upit“ koje poziva resetForm().
- 

### **4. resetForm funkcija**

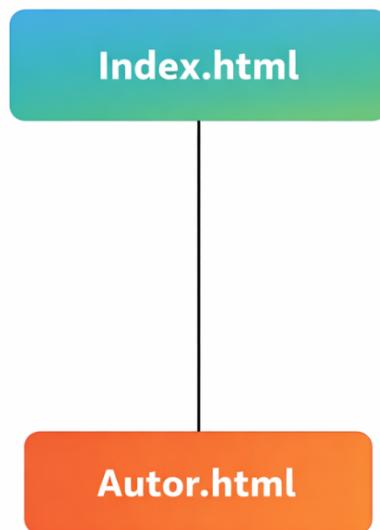
Vraća formu u prvobitno stanje nakon uspešnog slanja.

### **Koraci:**

1. Resetuje sve inpute (form.reset()).
2. Uklanja klase success i error.
3. Postavlja default vrednost slider-a.
4. Sakriva div sa potvrdom i prikazuje formu ponovo.

## 2. Organizacija

- Organizaciona šema



- Mapa sajta

```
- <?xml version="1.0" encoding="UTF-8"?>
- <urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">

-
-     <!-- Glavna stranica -->
-     <url>
-         <loc>https://ngbusiness.github.io/MobilShop/index.html</loc>
-         <lastmod>2026-01-12</lastmod>
-         <changefreq>weekly</changefreq>
-         <priority>1.0</priority>
-     </url>
-
-     <!-- Stranica autora -->
-     <url>
-         <loc>https://ngbusiness.github.io/MobilShop/autor.html</loc>
```

```
-      <lastmod>2026-01-12</lastmod>
-      <changefreq>monthly</changefreq>
-      <priority>0.5</priority>
-    </url>
-
-  </urlset>
-
```

- Slike stranica i opisi funkcionalnosti

Klikom na dugme Korpa izadje tekst u donjem desnom uglu (funkcija Toast) koji nestaje nakon 3 sekunde.

The screenshot shows a product page for the iPhone 15 Pro Max. At the top left is a red button labeled "AKCIJA". Below it are four iPhone 15 Pro Max phones standing vertically, showing their back panels in black, white, dark blue, and light gray colors. The Apple logo is visible on the light gray phone. The brand name "APPLE" is at the top center. Below it is the product name "iPhone 15 Pro Max". Underneath the name are the specifications "256GB | Titanijum | A17 Pro". The price is listed as "189.990 RSD" with a crossed-out original price of "209.990 RSD" next to it. At the bottom are two buttons: a blue "Detalji" button and an orange "Korpa" button. A black toast notification at the bottom says "iPhone 15 Pro Max je dodat u korpu!".

# Kodovi

## - Index strana (index.html)

```
- <!DOCTYPE html>
- <html lang="sr">
-   <head>
-     <meta charset="UTF-8">
-     <meta name="viewport" content="width=device-width, initial-
scale=1.0">
-     <meta name="description" content="MobilShop - Vaša destinacija za
najnovije mobilne telefone. Širok izbor Samsung, Apple, Xiaomi i drugih
brendova po najboljim cenama.">
-     <meta name="keywords" content="mobilni telefoni, smartphone,
Samsung, iPhone, Xiaomi, kupovina telefona, MobilShop">
-     <meta name="author" content="MobilShop">
-     <meta name="robots" content="index, follow">
-     <meta property="og:title" content="MobilShop - Premium Mobilni
Telefoni">
-     <meta property="og:description" content="Pronađite savršen telefon
za vas. Besplatna dostava, garancija i podrška.">
-     <meta property="og:type" content="website">
-     <title>MobilShop - Premium Mobilni Telefoni</title>
-     <link rel="stylesheet" href="/assets/css/style.css">
-   </head>
-   <body>
-     <header id="header">
-       <!-- Dinamicki generisan header -->
-     </header>
-
-     <nav id="main-nav">
-       <!-- Dinamicki generisan meni -->
-     </nav>
-
-     <main>
-       <section id="hero">
-         <!-- Dinamicki generisan hero -->
-       </section>
-
-       <section id="featured-section">
-         <div class="container">
-           <h2 id="featured-title"><!-- Dinamicki --></h2>
-           <p id="featured-subtitle"><!-- Dinamicki --></p>
-           <div id="featured-products" class="products-grid">
```

```

-          <!-- Dinamicki generisani proizvodi -->
-          </div>
-      </div>
-  </section>

-
<section id="brands-section">
    <div class="container">
        <h2 id="brands-title"><!-- Dinamicki --></h2>
        <div id="brands-container" class="brands-grid">
            <!-- Dinamicki generisani brendovi -->
            </div>
        </div>
    </section>

-
<section id="contact-section">
    <div class="container">
        <h2 id="contact-title"><!-- Dinamicki --></h2>
        <p id="contact-subtitle"><!-- Dinamicki --></p>

        <form id="contact-form" novalidate>
            <div class="form-row">
                <div class="form-group">
                    <label for="ime">Ime i prezime *</label>
                    <input type="text" id="ime" name="ime"
placeholder="Unesite ime i prezime">
                    <span class="error-message" id="ime-
error"></span>
                </div>

                <div class="form-group">
                    <label for="email">Email adresa *</label>
                    <input type="email" id="email" name="email"
placeholder="vas@email.com">
                    <span class="error-message" id="email-
error"></span>
                </div>
            </div>

            <div class="form-row">
                <div class="form-group">
                    <label for="telefon">Broj telefona
*</label>
                    <input type="tel" id="telefon"
name="telefon" placeholder="06X XXX XXXX">
                </div>
            </div>
        </form>
    </div>

```

```

-           <span class="error-message" id="telefon-
error"></span>
-       </div>

-       <div class="form-group">
-           <label for="brend">Preferirani
brend</label>
-           <select id="brend" name="brend">
-               <option value="">-- Izaberite brend --
-           </option>
-               <option value="apple">Apple</option>
-               <option
value="samsung">Samsung</option>
-               <option value="xiaomi">Xiaomi</option>
-               <option value="huawei">Huawei</option>
-               <option
value="oneplus">OnePlus</option>
-           </select>
-           <span class="error-message" id="brend-
error"></span>
-       </div>
-   </div>

-   <div class="form-group">
-       <label for="budzet">Budžet (RSD)</label>
-       <input type="range" id="budzet" name="budzet"
min="20000" max="300000" step="10000" value="100000">
-       <output id="budzet-output">100,000 RSD</output>
-   </div>

-   <div class="form-group">
-       <label>Šta vas zanima?</label>
-       <div class="checkbox-group">
-           <label class="checkbox-label">
-               <input type="checkbox"
name="interesovanje" value="novi">
-                   Novi telefoni
-               </label>
-               <label class="checkbox-label">
-                   <input type="checkbox"
name="interesovanje" value="polovni">
-                       Polovni telefoni
-                   </label>
-                   <label class="checkbox-label">

```

```

-           <input type="checkbox"
- name="interesovanje" value="akcije">
-                         Akcijske ponude
-           </label>
-           <label class="checkbox-label">
-             <input type="checkbox"
- name="interesovanje" value="oprema">
-                         Oprema i dodaci
-           </label>
-         </div>
-       </div>

-       <div class="form-group">
-         <label for="poruka">Vaša poruka</label>
-         <textarea id="poruka" name="poruka" rows="4"
- placeholder="Opisite šta tražite..."></textarea>
-         <span class="error-message" id="poruka-
error"></span>
-       </div>

-       <div class="form-group">
-         <label class="checkbox-label">
-           <input type="checkbox" id="uslovi"
- name="uslovi">
-             Prihvatom <a href="#">uslove korišćenja</a>
-             i <a href="#">politiku privatnosti</a> *
-           </label>
-           <span class="error-message" id="uslovi-
error"></span>
-       </div>

-           <button type="submit" class="btn btn-
primary">Pošalji upit</button>
-         </form>

-         <div id="form-success" class="form-success hidden">
-           <!-- Dinamicki prikaz uspesne poruke -->
-         </div>
-       </div>
-     </section>

-     <section id="why-us-section">
-       <div class="container">
-         <h2 id="why-us-title"><!-- Dinamicki --></h2>
-         <div id="why-us-container" class="features-grid">

```

```

-          <!-- Dinamicki generisane prednosti -->
-          </div>
-      </div>
-  </section>
- </main>

-
-  <footer id="footer">
-      <!-- Dinamicki generisan footer -->
-  </footer>

-
-  <div id="modal" class="modal hidden">
-      <div class="modal-content">
-          <span class="modal-close">&times;</span>
-          <div id="modal-body">
-              <!-- Dinamicki sadrzaj modala -->
-          </div>
-      </div>
-  </div>

-
-  <script src="js/main.js"></script>
-  <div id="toast" class="hidden"></div>

-
- </body>
- </html>
-
```

## - Autor strana (author.html)

```

-  <!DOCTYPE html>
-  <html lang="sr">
-  <head>
-      <meta charset="UTF-8">
-      <meta name="viewport" content="width=device-width, initial-
scale=1.0">
-      <title>Autor - Nikola Gajić</title>
-      <link rel="stylesheet" href="/assets/css/style.css">
-  </head>
-  <body>
-      <div id="autor-root"></div>

-
-      <script src="/js/main.js"></script>
-  </body>
- </html>
-
```

## - Javascript (main.js)

```
// ===== PODACI ZA DINAMICKI ISPIS =====

const siteData = {
    // Header podaci
    header: {
        logoText: "MobilShop",
        logoIcon: "█",
        phone: "+381 11 123 4567",
        email: "info@mobilshop.rs",
        workHours: "Pon-Sub: 09-20h",
    },
    // Navigacija
    navigation: [
        { name: "Početna", href: "#hero", active: true },
        { name: "Telefoni", href: "#featured-section" },
        { name: "Brendovi", href: "#brands-section" },
        { name: "Kontakt", href: "#contact-section" },
        { name: "O nama", href: "#why-us-section" },
        { name: "Autor", href: "autor.html" },
    ],
    // Hero sekcija
    hero: {
        title: "Pronađite Vaš Savršeni Telefon",
        subtitle: "Najnoviji modeli vodećih proizvođača po najboljim cenama. Besplatna dostava i 2 godine garancije.",
        badge: "NOVO 2024",
        buttons: [
            { text: "Pogledaj ponudu", class: "btn-primary", href: "#featured-section" },
            { text: "Kontaktiraj nas", class: "btn-secondary", href: "#contact-section" },
        ],
    },
    // Naslovi sekcija
    sections: {
        featured: {
            title: "Izdvojeni Telefoni",
            subtitle: "Najpopularniji modeli koje naši kupci obožavaju",
        }
    }
}
```

```
- },
- brands: {
-   title: "Brendovi u Ponudi",
-   subtitle: "Radimo samo sa proverenim proizvođačima",
- },
- contact: {
-   title: "Kontaktirajte Nas",
-   subtitle: "Imate pitanja? Pošaljite nam upit i odgovorićemo vam u
najkraćem roku.",
- },
- whyUs: {
-   title: "Zašto Izabratи Nas?",
- },
- },
- // Proizvodi
products: [
  {
    id: 1,
    brand: "Apple",
    name: "iPhone 15 Pro Max",
    specs: "256GB | Titanijum | A17 Pro",
    price: 189990,
    oldPrice: 209990,
    badge: "sale",
    image: "assets/img/iphone-15-pro-max.png",
    fullSpecs: {
      Ekran: '6.7" Super Retina XDR',
      Procesor: "A17 Pro",
      RAM: "8GB",
      Memorija: "256GB",
      Kamera: "48MP + 12MP + 12MP",
      Baterija: "4422mAh",
    },
  },
  {
    id: 2,
    brand: "Samsung",
    name: "Galaxy S24 Ultra",
    specs: "512GB | Titanijum | SD 8 Gen 3",
    price: 174990,
    oldPrice: null,
    badge: "new",
    image: "assets/img/s24-ultra.jpg",
    fullSpecs: {
```

```
-        Ekran: '6.8" Dynamic AMOLED 2X',
-        Procesor: "Snapdragon 8 Gen 3",
-        RAM: "12GB",
-        Memorija: "512GB",
-        Kamera: "200MP + 50MP + 12MP + 10MP",
-        Baterija: "5000mAh",
-    },
-},
{
-    id: 3,
-    brand: "Xiaomi",
-    name: "14 Ultra",
-    specs: "512GB | Crna | SD 8 Gen 3",
-    price: 149990,
-    oldPrice: 169990,
-    badge: "popular",
-    image: "assets/img/xiaomi-14-ultra.png",
-    fullSpecs: {
-        Ekran: '6.73" LTPO AMOLED',
-        Procesor: "Snapdragon 8 Gen 3",
-        RAM: "16GB",
-        Memorija: "512GB",
-        Kamera: "50MP + 50MP + 50MP",
-        Baterija: "5000mAh",
-    },
-},
{
-    id: 4,
-    brand: "OnePlus",
-    name: "12",
-    specs: "256GB | Zelena | SD 8 Gen 3",
-    price: 99990,
-    oldPrice: null,
-    badge: "new",
-    image: "assets/img/oneplus-12-green.png",
-    fullSpecs: {
-        Ekran: '6.82" LTPO AMOLED',
-        Procesor: "Snapdragon 8 Gen 3",
-        RAM: "12GB",
-        Memorija: "256GB",
-        Kamera: "50MP + 48MP + 64MP",
-        Baterija: "5400mAh",
-    },
-},
{

```

```

        id: 5,
        brand: "Google",
        name: "Pixel 8 Pro",
        specs: "256GB | Obsidian | Tensor G3",
        price: 119990,
        oldPrice: 134990,
        badge: "sale",
        image: "assets/img/google-pixel-8-pro.png",
        fullSpecs: {
            Ekran: '6.7" LTPO OLED',
            Procesor: "Google Tensor G3",
            RAM: "12GB",
            Memorija: "256GB",
            Kamera: "50MP + 48MP + 48MP",
            Baterija: "5050mAh",
        },
    },
    {
        id: 6,
        brand: "Samsung",
        name: "Galaxy Z Fold 5",
        specs: "512GB | Crna | Preklopni",
        price: 219990,
        oldPrice: null,
        badge: "popular",
        image: "assets/img/Samsung-Galaxy-Z-Fold-5.webp",
        fullSpecs: {
            Ekran: '7.6" + 6.2" Dynamic AMOLED',
            Procesor: "Snapdragon 8 Gen 2",
            RAM: "12GB",
            Memorija: "512GB",
            Kamera: "50MP + 12MP + 10MP",
            Baterija: "4400mAh",
        },
    },
],
]

// Brendovi
brands: [
    { name: "Apple", icon: "\ud83d\udcbb", count: 24 },
    { name: "Samsung", icon: "\ud83d\udcbb", count: 38 },
    { name: "Xiaomi", icon: "\u26bd\ufe0f", count: 45 },
    { name: "Huawei", icon: "\ud83d\udcbb", count: 18 },
    { name: "OnePlus", icon: "\ud83d\udcbb", count: 12 },
    { name: "Google", icon: "\ud83d\udcbb", count: 8 },
]

```

```

- ],
-
- // Prednosti
- features: [
- {
-   icon: "🚚",
-   title: "Besplatna Dostava",
-   description: "Za sve porudžbine iznad 50.000 RSD dostava je potpuno besplatna.",
- },
- {
-   icon: "⌚",
-   title: "2 Godine Garancije",
-   description: "Svi uređaji dolaze sa punom proizvođačkom garancijom.",
- },
- {
-   icon: "💳",
-   title: "Plaćanje na Rate",
-   description: "Mogućnost plaćanja do 24 rate bez kamate.",
- },
- {
-   icon: "📱",
-   title: "Zamena Starog Telefona",
-   description: "Zamenite stari telefon za popust na novi model.",
- },
- ],
-
- // Footer
- footer: {
-   about: "MobilShop je vaša pouzdana destinacija za kupovinu mobilnih telefona od 2015. godine.",
-   quickLinks: [
-     { name: "Svi telefoni", href: "#" },
-     { name: "Akcijske ponude", href: "#" },
-     { name: "Novi modeli", href: "#" },
-     { name: "Oprema", href: "#" },
-   ],
-   support: [
-     { name: "Česta pitanja", href: "#" },
-     { name: "Garancija", href: "#" },
-     { name: "Dostava", href: "#" },
-     { name: "Reklamacije", href: "#" },
-   ],
-   contact: {
-
```

```

        address: "Knez Mihailova 25, Beograd",
        phone: "+381 11 123 4567",
        email: "info@mobilshop.rs",
    },
    social: ["", "", "", ""],
    copyright: "© 2026 MobilShop. Sva prava zadržana.",
},
}

// ===== REGULARNI IZRAZI ZA VALIDACIJU =====

const patterns = {
    // Ime: samo slova, razmaci, min 3 karaktera
    ime: /^[A-Za-zĂćĂčĂžĂšĂđă\s]{3,50}\$/,

    // Email: standardni email format
    email: /^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}\$/,

    // Telefon: srpski format (06X XXX XXXX ili +381...)
    telefon: ^(\+381|0)6[0-9]\s?[0-9]{3}\s?[0-9]{3,4}\$/, 

    // Poruka: min 10 karaktera
    poruka: ^.{10,500}\$/,
}

// Poruke gresaka
const errorMessages = {
    ime: "Ime mora sadržati 3-50 slova (bez brojeva i specijalnih znakova)",
    email: "Unesite validnu email adresu (npr. ime@domen.com)",
    telefon: "Unesite validan broj telefona (npr. 064 123 4567)",
    poruka: "Poruka mora imati između 10 i 500 karaktera",
    uslovi: "Morate prihvati uslove korišćenja",
}

// ===== FUNKCIJA ZA PRIKAZ TOAST PORUKE =====
function showToast(message, duration = 3000) {
    let toast = document.getElementById("toast")

    // Ako toast ne postoji, kreiraj ga i ubaci u body
    if (!toast) {
        toast = document.createElement("div")
        toast.id = "toast"
        toast.classList.add("hidden") // Početno sakriven
        document.body.appendChild(toast)
    }
}

```

```

-     toast.textContent = message
-     toast.classList.remove("hidden")
-     toast.classList.add("show")

-
-     // Automatski sakrij nakon duration milisekundi
-     setTimeout(() => {
-         toast.classList.remove("show")
-         toast.classList.add("hidden")
-     }, duration)
- }

-
// ===== FUNKCIJE ZA DINAMICKO KREIRANJE SADRZAJA =====

-
// Formatiranje cene
function formatPrice(price) {
    return price.toLocaleString("sr-RS") + " RSD"
}

-
// Kreiranje headera
function renderHeader() {
    const header = document.getElementById("header")
    const data = siteData.header

    header.innerHTML =
        <div class="container">
            <div class="header-content">
                <div class="logo">
                    <div class="logo-icon">${data.logoIcon}</div>
                    <div class="logo-text">${data.logoText}</div>
                </div>
                <div class="header-info">
                    <div class="header-info-item">
                        <span>📞</span>
                        <span>${data.phone}</span>
                    </div>
                    <div class="header-info-item">
                        <span>✉</span>
                        <span>${data.email}</span>
                    </div>
                    <div class="header-info-item">
                        <span>⌚</span>
                        <span>${data.workHours}</span>
                    </div>
                </div>
            </div>
        </div>

```

```

        </div>
    </div>

}

// Kreiranje navigacije
function renderNavigation() {
    const nav = document.getElementById("main-nav")
    const navItems = siteData.navigation

    let menuHTML = ""
    navItems.forEach((item) => {
        const activeClass = item.active ? "active" : ""
        menuHTML += `<li><a href="${item.href}" class="${activeClass}">${item.name}</a></li>`
    })

    nav.innerHTML = `
        <div class="nav-container">
            <button class="mobile-menu-toggle" aria-label="Otvori meni">☰</button>
            <ul class="nav-menu">
                ${menuHTML}
            </ul>
            <div class="nav-search">
                <input type="search" placeholder="Pretraži telefone..." id="search-input">
            </div>
        </div>
    `

    // Dodaj event listener za mobilni meni
    setupMobileMenu()
}

// Kreiranje hero sekcije
function renderHero() {
    const hero = document.getElementById("hero")
    const data = siteData.hero

    let buttonsHTML = ""
    data.buttons.forEach((btn) => {
        buttonsHTML += `<a href="${btn.href}" class="btn ${btn.class}">${btn.text}</a>`
    })
}

```

```

- hero.innerHTML = `
-   <div class="container">
-     <div class="hero-content">
-       <div class="hero-text animate-in">
-         <h1>${data.title}</h1>
-         <p>${data.subtitle}</p>
-         <div class="hero-buttons">
-           ${buttonsHTML}
-         </div>
-       </div>
-       <div class="hero-image">
-         <div class="hero-phone-display">
-           <div class="phone-mockup">
-             <div class="phone-screen">📱</div>
-           </div>
-           <div class="hero-badge
pulse">${data.badge}</div>
-         </div>
-       </div>
-     </div>
-   `

}

// Kreiranje proizvoda
function renderProducts() {
  const container = document.getElementById("featured-products")
  const titleEl = document.getElementById("featured-title")
  const subtitleEl = document.getElementById("featured-subtitle")

  titleEl.textContent = siteData.sections.featured.title
  subtitleEl.textContent = siteData.sections.featured.subtitle
  subtitleEl.className = "section-subtitle text-center"
  titleEl.className = "text-center"

  let productsHTML = ""

  siteData.products.forEach((product, index) => {
    const badgeClass = product.badge === "sale" ? "badge-sale" :
product.badge === "new" ? "badge-new" : "badge-popular"
    const badgeText = product.badge === "sale" ? "Akcija" :
product.badge === "new" ? "Novo" : "Popularno"
  })
}

```

```

        const oldPriceHTML = product.oldPrice ? `<span class="price-old">${formatPrice(product.oldPrice)}</span>` : ""

        productsHTML += `
            <div class="product-card animate-in" style="animation-delay: ${index * 0.1}s" data-product-id="${product.id}">
                <span class="product-badge ${badgeClass}">${badgeText}</span>
                <div class="product-image">
                    
                </div>
                <div class="product-info">
                    <div class="product-brand">${product.brand}</div>
                    <h3 class="product-name">${product.name}</h3>
                    <p class="product-specs">${product.specs}</p>
                    <div class="product-price">
                        <span class="price-current">${formatPrice(product.price)}</span>
                        ${oldPriceHTML}
                    </div>
                    <div class="product-actions">
                        <button class="btn btn-primary btn-details" data-id="${product.id}">Detalji</button>
                        <button class="btn btn-accent btn-cart" data-id="${product.id}">Korpa</button>
                    </div>
                </div>
            </div>
        `

    })

    container.innerHTML = productsHTML

    // Dodaj event listener za proizvode
    setupProductEvents()
}

// Kreiranje brendova
function renderBrands() {
    const container = document.getElementById("brands-container")
    const titleEl = document.getElementById("brands-title")

    titleEl.textContent = siteData.sections.brands.title
    titleEl.className = "text-center"
}

```

```

let brandsHTML = ""

siteData.brands.forEach((brand, index) => {
    brandsHTML += `
        <div class="brand-card animate-in" style="animation-delay: ${index * 0.1}s" data-brand="${brand.name}">
            <div class="brand-icon">${brand.icon}</div>
            <div class="brand-name">${brand.name}</div>
            <div class="brand-count">${brand.count} modela</div>
        </div>
    `
})

container.innerHTML = brandsHTML

// Dodaj event listenere za brendove
setupBrandFilter()
}

// Kreiranje prednosti
function renderFeatures() {
    const container = document.getElementById("why-us-container")
    const titleEl = document.getElementById("why-us-title")

    titleEl.textContent = siteData.sections.whyUs.title
    titleEl.className = "text-center"

    let featuresHTML = ""

    siteData.features.forEach((feature, index) => {
        featuresHTML += `
            <div class="feature-card animate-in" style="animation-delay: ${index * 0.1}s">
                <div class="feature-icon">${feature.icon}</div>
                <h3 class="feature-title">${feature.title}</h3>
                <p class="feature-description">${feature.description}</p>
            </div>
        `
    })

    container.innerHTML = featuresHTML
}

// Kreiranje footer-a

```

```

function renderFooter() {
  const footer = document.getElementById("footer")
  const data = siteData.footer

  let quickLinksHTML = ""
  data.quickLinks.forEach((link) => {
    quickLinksHTML += `<li><a href="${link.href}">${link.name}</a></li>`
  })

  let supportLinksHTML = ""
  data.support.forEach((link) => {
    supportLinksHTML += `<li><a href="${link.href}">${link.name}</a></li>`
  })

  let socialHTML = ""
  data.social.forEach((icon) => {
    socialHTML += `<a href="#" class="social-link">${icon}</a>`
  })

  footer.innerHTML = `
    <div class="container">
      <div class="footer-content">
        <div class="footer-section">
          <h3>O Nama</h3>
          <p>${data.about}</p>
          <div class="social-links">
            ${socialHTML}
          </div>
        </div>
        <div class="footer-section">
          <h3>Brzi Linkovi</h3>
          <ul>${quickLinksHTML}</ul>
        </div>
        <div class="footer-section">
          <h3>Podrška</h3>
          <ul>${supportLinksHTML}</ul>
        </div>
        <div class="footer-section">
          <h3>Kontakt</h3>
          <p>📍 ${data.contact.address}</p>
          <p>📞 ${data.contact.phone}</p>
          <p>✉️ ${data.contact.email}</p>
        </div>
    </div>
  
```

```

        </div>
    <div class="footer-bottom">
        ${data.copyright}
    </div>
</div>

}

// Postavljanje naslova kontakt sekcije
function renderContactTitles() {
    const titleEl = document.getElementById("contact-title")
    const subtitleEl = document.getElementById("contact-subtitle")

    titleEl.textContent = siteData.sections.contact.title
    titleEl.className = "text-center"
    subtitleEl.textContent = siteData.sections.contact.subtitle
    subtitleEl.className = "section-subtitle text-center"
}

// ===== VALIDACIJA FORME =====

function validateField(field, pattern, errorElementId) {
    const value = field.value.trim()
    const errorElement = document.getElementById(errorElementId)

    if (value === "") {
        field.classList.remove("success", "error")
        errorElement.textContent = ""
        return false
    }

    if (pattern.test(value)) {
        field.classList.remove("error")
        field.classList.add("success")
        errorElement.textContent = ""
        return true
    } else {
        field.classList.remove("success")
        field.classList.add("error")
        const fieldName = errorElementId.replace("-error", "")
        errorElement.textContent = errorMessages[fieldName]
        return false
    }
}

```

```

- function setupFormValidation() {
-   const form = document.getElementById("contact-form")
-   const imeField = document.getElementById("ime")
-   const emailField = document.getElementById("email")
-   const telefonField = document.getElementById("telefon")
-   const porukaField = document.getElementById("poruka")
-   const usloviField = document.getElementById("uslovi")
-   const budzetField = document.getElementById("budzet")
-   const budzetOutput = document.getElementById("budzet-output")
-
-   // Real-time validacija
-   imeField.addEventListener("input", () => validateField(imeField,
-     patterns.ime, "ime-error"))
-   emailField.addEventListener("input", () => validateField(emailField,
-     patterns.email, "email-error"))
-   telefonField.addEventListener("input", () =>
-     validateField(telefonField, patterns.telefon, "telefon-error"))
-   porukaField.addEventListener("input", () =>
-     validateField(porukaField, patterns.poruka, "poruka-error"))
-
-   // Budzet slider
-   budzetField.addEventListener("input", () => {
-     budzetOutput.textContent =
-       formatPrice(Number.parseInt(budzetField.value))
-   })
-
-   // Form submit
-   form.addEventListener("submit", (e) => {
-     e.preventDefault()
-
-     // Validacija svih polja
-     const isImeValid = validateField(imeField, patterns.ime, "ime-
-       error")
-     const isEmailValid = validateField(emailField, patterns.email,
-       "email-error")
-     const isTelefonValid = validateField(telefonField,
-       patterns.telefon, "telefon-error")
-
-     // Provera uslova
-     const usloviError = document.getElementById("uslovi-error")
-     let isUsloviValid = true
-     if (!usloviField.checked) {
-       usloviError.textContent = errorMessages.uslovi
-       isUsloviValid = false
-     } else {
-
```

```

        usloviError.textContent = ""
    }

    // Ako je sve validno
    if (isImeValid && isEmailValid && isTelefonValid && isUsloviValid)
    {
        // Prikupi podatke
        const formData = {
            ime: imeField.value.trim(),
            email: emailField.value.trim(),
            telefon: telefonField.value.trim(),
            brend: document.getElementById("brend").value,
            budzet: budzetField.value,
            interesovanja:
                Array.from(document.querySelectorAll('input[name="interesovanje"]:checked')).map(
                    (cb) => cb.value,
                ),
            poruka: porukaField.value.trim(),
        }

        // Prikazi uspesnu poruku
        showFormSuccess(formData)
    }
}

function showFormSuccess(data) {
    const form = document.getElementById("contact-form")
    const successDiv = document.getElementById("form-success")

    const brendText = data.brend ? data.brend.charAt(0).toUpperCase() + data.brend.slice(1) : "Nije izabrano"
    const interesovanjaText = data.interesovanja.length > 0 ? data.interesovanja.join(", ") : "Nije izabrano"

    successDiv.innerHTML = `
        <div class="form-success-icon">✓</div>
        <h3>Hvala Vam, ${data.ime}!</h3>
        <p>Vaš upit je uspešno poslat. Odgovorićemo vam u najkraćem
        roku na ${data.email}.</p>
        <div class="submitted-data">
            <h4>Poslati podaci:</h4>
            <p><strong>Ime:</strong> ${data.ime}</p>
            <p><strong>Email:</strong> ${data.email}</p>
    
```

```

-           <p><strong>Telefon:</strong> ${data.telefon}</p>
-           <p><strong>Brend:</strong> ${brendText}</p>
-           <p><strong>Budžet:</strong>
-           ${formatPrice(Number.parseInt(data.budzet))}</p>
-           <p><strong>Interesovanja:</strong> ${interesovanjaText}</p>
-           <p><strong>Poruka:</strong> ${data.poruka} || "Nema
poruke"</p>
-           </div>
-           <button class="btn btn-primary" onclick="resetForm()"
style="margin-top: 1rem;">Pošalji novi upit</button>
-
-
form.classList.add("hidden")
successDiv.classList.remove("hidden")
}

function resetForm() {
const form = document.getElementById("contact-form")
const successDiv = document.getElementById("form-success")

form.reset()
form.querySelectorAll("input, textarea, select").forEach((field) => {
  field.classList.remove("success", "error")
})
document.getElementById("budzet-output").textContent = "100,000 RSD"

successDiv.classList.add("hidden")
form.classList.remove("hidden")
}

// ===== MOBILNI MENI =====

function setupMobileMenu() {
const toggle = document.querySelector(".mobile-menu-toggle")
const menu = document.querySelector(".nav-menu")

if (toggle && menu) {
  toggle.addEventListener("click", () => {
    menu.classList.toggle("active")
    toggle.textContent = menu.classList.contains("active") ? "X" :
"☰"
  })
}

// Zatvori meni kada se klikne na link
menu.querySelectorAll("a").forEach((link) => {

```

```

        link.addEventListener("click", () => {
            menu.classList.remove("active")
            toggle.textContent = "☰"
        })
    })
}

// ===== MODAL ZA PROIZVODE =====

function showProductModal(productId) {
    const product = siteData.products.find((p) => p.id === productId)
    if (!product) return

    const modal = document.getElementById("modal")
    const modalBody = document.getElementById("modal-body")

    let specsHTML = ""
    for (const [key, value] of Object.entries(product.fullSpecs)) {
        specsHTML += `<li><span>${key}</span><span>${value}</span></li>`
    }

    const oldPriceHTML = product.oldPrice ? `<span class="price-old">${formatPrice(product.oldPrice)}</span>` : ""

    modalBody.innerHTML =
        <div class="modal-product-image">
            
        </div>
        <div class="modal-product-info">
            <div class="product-brand">${product.brand}</div>
            <h2>${product.name}</h2>
            <div class="product-price" style="margin: 1rem 0;">
                <span class="price-current">${formatPrice(product.price)}</span>
                ${oldPriceHTML}
            </div>
            <p>${product.specs}</p>
            <div class="modal-specs">
                <h4>Specifikacije:</h4>
                <ul>${specsHTML}</ul>
            </div>
            <div class="product-actions">
                <button class="btn btn-accent btn-cart" data-id="${product.id}">Dodaj u korpu</button>
            </div>
        </div>

```

```

        <a href="#contact-section" class="btn btn-primary"
    onclick="closeModal()">Pošalji upit</a>
        </div>
    </div>
    <br>

    modal.classList.remove("hidden")
}

function closeModal() {
    const modal = document.getElementById("modal")
    modal.classList.add("hidden")
}

function setupProductEvents() {
    // Detalji dugmici
    document.querySelectorAll(".btn-details").forEach((btn) => {
        btn.addEventListener("click", function () {
            const productId = Number.parseInt(this.getAttribute("data-id"))
            showProductModal(productId)
        })
    })
}

// Korpa dugmici
document.querySelectorAll(".btn-cart").forEach((btn) => {
    btn.addEventListener("click", function () {
        const productId = Number.parseInt(this.getAttribute("data-id"))
        const product = siteData.products.find((p) => p.id === productId)
        if (product) {
            showToast(` ${product.name} je dodat u korpu! `)
        }
    })
})

// Modal close events
function setupModalEvents() {
    const modal = document.getElementById("modal")
    const closeBtn = document.querySelector(".modal-close")

    closeBtn.addEventListener("click", closeModal)

    modal.addEventListener("click", (e) => {
        if (e.target === modal) {
            closeModal()
        }
    })
}

```

```

-     }
-   })

-   document.addEventListener("keydown", (e) => {
-     if (e.key === "Escape") {
-       closeModal()
-     }
-   })
}

// ===== FILTER PO BRENDU =====

function setupBrandFilter() {
  document.querySelectorAll(".brand-card").forEach((card) => {
    card.addEventListener("click", function () {
      const brandName = this.getAttribute("data-brand")
      filterProductsByBrand(brandName)
    })
  })
}

function filterProductsByBrand(brandName) {
  const container = document.getElementById("featured-products")
  const filteredProducts = siteData.products.filter((p) => p.brand ===
brandName)

  if (filteredProducts.length === 0) {
    container.innerHTML = `<p class="text-center" style="grid-column: 1/-1;">Nema proizvoda za izabrani brend.</p>`
    return
  }

  // Dinamicki prikazi filtrirane proizvode
  let productsHTML = ""
  filteredProducts.forEach((product, index) => {
    const badgeClass = product.badge === "sale" ? "badge-sale" :
product.badge === "new" ? "badge-new" : "badge-popular"
    const badgeText = product.badge === "sale" ? "Akcija" :
product.badge === "new" ? "Novo" : "Popularno"

    const oldPriceHTML = product.oldPrice ? `<span class="price-old">${formatPrice(product.oldPrice)}</span>` : ""

    productsHTML += `
```

```

        <div class="product-card animate-in" style="animation-
delay: ${index * 0.1}s" data-product-id="${product.id}">
            <span class="product-badge"
${badgeClass}">${badgeText}</span>
            <div class="product-image">
                
            </div>
            <div class="product-info">
                <div class="product-brand">${product.brand}</div>
                <h3 class="product-name">${product.name}</h3>
                <p class="product-specs">${product.specs}</p>
                <div class="product-price">
                    <span class="price-
current">${formatPrice(product.price)}</span>
                    ${oldPriceHTML}
                </div>
                <div class="product-actions">
                    <button class="btn btn-primary btn-details"
data-id="${product.id}">Detalji</button>
                    <button class="btn btn-accent btn-cart" data-
id="${product.id}">${Korpa}</button>
                </div>
            </div>
        </div>
    <)
}

productsHTML += `<button class="btn btn-secondary"
onclick="renderProducts()" style="grid-column: 1/-1; margin-top:
1rem;">Prikaži sve proizvode</button>

container.innerHTML = productsHTML
setupProductEvents()

// Scroll do proizvoda
document.getElementById("featured-section").scrollIntoView({
behavior: "smooth" })
}

// ===== PRETRAGA =====

function setupSearch() {
    const searchInput = document.getElementById("search-input")

    if (searchInput) {

```

```

- searchInput.addEventListener("input", function () {
-   const query = this.value.toLowerCase().trim()
-
-   if (query.length < 2) {
-     renderProducts()
-     return
-   }
-
-   const filteredProducts = siteData.products.filter(
-     (p) =>
-       p.name.toLowerCase().includes(query) ||
-       p.brand.toLowerCase().includes(query) ||
-       p.specs.toLowerCase().includes(query),
-   )
-
-   displayFilteredProducts(filteredProducts)
- )
- }
}
}

function displayFilteredProducts(products) {
  const container = document.getElementById("featured-products")
-
  if (products.length === 0) {
    container.innerHTML = `<p class="text-center" style="grid-column: 1/-1;">Nema rezultata pretrage.</p>`
    return
  }
-
  let productsHTML = ""
  products.forEach((product, index) => {
    const badgeClass = product.badge === "sale" ? "badge-sale" :
product.badge === "new" ? "badge-new" : "badge-popular"
    const badgeText = product.badge === "sale" ? "Akcija" :
product.badge === "new" ? "Novo" : "Popularno"
-
    const oldPriceHTML = product.oldPrice ? `<span class="price-old">${formatPrice(product.oldPrice)}</span>` : ""
-
    productsHTML += `
      <div class="product-card animate-in" style="animation-delay: ${index * 0.1}s" data-product-id="${product.id}">
        <span class="product-badge ${badgeClass}">${badgeText}</span>
        <div class="product-image">

```

```

        
    </div>
    <div class="product-info">
        <div class="product-brand">${product.brand}</div>
        <h3 class="product-name">${product.name}</h3>
        <p class="product-specs">${product.specs}</p>
        <div class="product-price">
            <span class="price-
current">${formatPrice(product.price)}</span>
            ${oldPriceHTML}
        </div>
        <div class="product-actions">
            <button class="btn btn-primary btn-details" data-id="${product.id}">Detalji</button>
            <button class="btn btn-accent btn-cart" data-id="${product.id}">Korpa</button>
        </div>
    </div>
`)

})

container.innerHTML = productsHTML
setupProductEvents()
}

// ===== AKTIVNA NAVIGACIJA PRI SKROLOVANJU =====

function setupScrollSpy() {
const sections = document.querySelectorAll("section[id]")
const navLinks = document.querySelectorAll(".nav-menu a")

window.addEventListener("scroll", () => {
let current = ""

sections.forEach((section) => {
const sectionTop = section.offsetTop - 100
if (window.scrollY >= sectionTop) {
current = section.getAttribute("id")
}
})

navLinks.forEach((link) => {
link.classList.remove("active")
if (link.getAttribute("href") === "#" + current) {

```

```

        link.classList.add("active")
    }
})
}
}

// ===== Podaci za autora =====
const autorData = {
    ime: "Nikola Gajić",
    brojIndeksa: "160/23",
    slika: "/assets/img/ng23.jpg",
    tekst: "Student sam ICT-a, zainteresovan za web development i
moderne tehnologije. " +
        "Ovo je dinamički kreirana stranica autora pomoću
JavaScript-a."
};

// ===== Funkcija za kreiranje autor stranice =====
function kreirajAutorStranicu(data) {
    const root = document.getElementById("autor-root");

    // Kreiranje glavnog kontejnera
    const container = document.createElement("div");
    container.className = "author-container";

    // Slika
    const img = document.createElement("img");
    img.src = data.slika;
    img.alt = data.ime;
    img.className = "author-image";
    container.appendChild(img);

    // Ime i prezime
    const ime = document.createElement("div");
    ime.className = "author-name";
    ime.textContent = data.ime;
    container.appendChild(ime);

    // Broj indeksa
    const indeks = document.createElement("div");
    indeks.className = "author-index";
    indeks.textContent = `Broj indeksa: ${data.brojIndeksa}`;
    container.appendChild(indeks);

    // Tekst

```

```

-     const tekst = document.createElement("div");
-     tekst.className = "author-text";
-     tekst.textContent = data.tekst;
-     container.appendChild(tekst);

-
-     // Dugme nazad
-     const btn = document.createElement("a");
-     btn.href = "index.html";
-     btn.className = "btn-back";
-     btn.textContent = "Nazad na Početnu";
-     container.appendChild(btn);

-
-     // Ubacivanje u root div
-     root.appendChild(container);
- }

-
- // Ako postoji div za autora, kreiraj sadržaj
- const autorRoot = document.getElementById("autor-root");
- if (autorRoot) {
-     kreirajAutorStranicu(autorData);
- }

-
- // ===== INICIJALIZACIJA =====

-
- document.addEventListener("DOMContentLoaded", () => {
-     // Dinamicki ispis svih sekcija
-     renderHeader()
-     renderNavigation()
-     renderHero()
-     renderProducts()
-     renderBrands()
-     renderFeatures()
-     renderContactTitles()
-     renderFooter()

-
-     // Postavljanje event listenera
-     setupFormValidation()
-     setupModalEvents()
-     setupSearch()
-     setupScrollSpy()
- })
-
```

## CSS(style.css)

```
/* ===== CSS RESET I OSNOVA ===== */
* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
}

:root {
    --primary-color: #1a56db;
    --primary-dark: #1e40af;
    --secondary-color: #0f172a;
    --accent-color: #f59e0b;
    --text-color: #334155;
    --text-light: #64748b;
    --background: #f8fafc;
    --white: #ffffff;
    --error-color: #dc2626;
    --success-color: #16a34a;
    --border-color: #e2e8f0;
    --shadow: 0 4px 6px -1px rgba(0, 0, 0, 0.1);
    --shadow-lg: 0 10px 15px -3px rgba(0, 0, 0, 0.1);
    --radius: 8px;
    --transition: all 0.3s ease;
}

html {
    scroll-behavior: smooth;
}

body {
    font-family: "Segoe UI", Tahoma, Geneva, Verdana, sans-serif;
    line-height: 1.6;
    color: var(--text-color);
    background-color: var(--background);
}

/* ===== TIPOGRAFIJA ===== */
h1,
h2,
h3,
h4 {
    color: var(--secondary-color);
    line-height: 1.2;
```

```
}

h1 {
  font-size: 2.5rem;
}

h2 {
  font-size: 2rem;
  margin-bottom: 1rem;
}

h3 {
  font-size: 1.25rem;
}

a {
  color: var(--primary-color);
  text-decoration: none;
  transition: var(--transition);
}

a:hover {
  color: var(--primary-dark);
}

/* ===== LAYOUT ===== */
.container {
  max-width: 1200px;
  margin: 0 auto;
  padding: 0 1.5rem;
}

section {
  padding: 4rem 0;
}

/* ===== HEADER ===== */
#header {
  background-color: var(--secondary-color);
  color: var(--white);
  padding: 0.75rem 0;
}

.header-content {
  display: flex;
```

```
justify-content: space-between;
align-items: center;
flex-wrap: wrap;
gap: 1rem;
}

.logo {
  display: flex;
  align-items: center;
  gap: 0.5rem;
}

.logo-icon {
  width: 40px;
  height: 40px;
  background: linear-gradient(135deg, var(--primary-color), var(--accent-color));
  border-radius: var(--radius);
  display: flex;
  align-items: center;
  justify-content: center;
  font-weight: bold;
  font-size: 1.25rem;
}

.logo-text {
  font-size: 1.5rem;
  font-weight: bold;
}

.logo-text span {
  color: var(--accent-color);
}

.header-info {
  display: flex;
  gap: 2rem;
  font-size: 0.9rem;
}

.header-info-item {
  display: flex;
  align-items: center;
  gap: 0.5rem;
}
```

```
/* ===== NAVIGACIJA ===== */
#main-nav {
    background-color: var(--white);
    box-shadow: var(--shadow);
    position: sticky;
    top: 0;
    z-index: 100;
}

.nav-container {
    display: flex;
    justify-content: space-between;
    align-items: center;
    padding: 0 1.5rem;
    max-width: 1200px;
    margin: 0 auto;
}

.nav-menu {
    display: flex;
    list-style: none;
    gap: 0;
}

.nav-menu li a {
    display: block;
    padding: 1rem 1.5rem;
    color: var(--text-color);
    font-weight: 500;
    transition: var(--transition);
    position: relative;
}

.nav-menu li a:hover,
.nav-menu li a.active {
    color: var(--primary-color);
    background-color: var(--background);
}

.nav-menu li a::after {
    content: "";
    position: absolute;
    bottom: 0;
    left: 50%;
    transform: translateX(-50%);
```

```
width: 0;
height: 3px;
background-color: var(--primary-color);
transition: var(--transition);
}

.nav-menu li a:hover::after,
.nav-menu li a.active::after {
  width: 100%;
}

.nav-search {
  display: flex;
  align-items: center;
  gap: 0.5rem;
}

.nav-search input {
  padding: 0.5rem 1rem;
  border: 1px solid var(--border-color);
  border-radius: var(--radius);
  font-size: 0.9rem;
  width: 200px;
  transition: var(--transition);
}

.nav-search input:focus {
  outline: none;
  border-color: var(--primary-color);
  box-shadow: 0 0 0 3px rgba(26, 86, 219, 0.1);
}

.mobile-menu-toggle {
  display: none;
  background: none;
  border: none;
  font-size: 1.5rem;
  cursor: pointer;
  color: var(--text-color);
}

/* ===== HERO SEKCIJA ===== */
#hero {
  background: linear-gradient(135deg, var(--secondary-color) 0%, #1e3a5f 100%);
  color: var(--white);
```

```
padding: 5rem 0;
position: relative;
overflow: hidden;
}

#hero::before {
  content: "";
  position: absolute;
  top: -50%;
  right: -20%;
  width: 600px;
  height: 600px;
  background: radial-gradient(circle, rgba(245, 158, 11, 0.1) 0%, transparent 70%);
  border-radius: 50%;
}

.hero-content {
  display: flex;
  align-items: center;
  justify-content: space-between;
  gap: 3rem;
  position: relative;
  z-index: 1;
}

.hero-text {
  flex: 1;
}

.hero-text h1 {
  color: var(--white);
  font-size: 3rem;
  margin-bottom: 1rem;
}

.hero-text h1 span {
  color: var(--accent-color);
}

.hero-text p {
  font-size: 1.25rem;
  opacity: 0.9;
  margin-bottom: 2rem;
}
```

```
.hero-buttons {
  display: flex;
  gap: 1rem;
}

.hero-image {
  flex: 1;
  text-align: center;
}

.hero-phone-display {
  position: relative;
  display: inline-block;
}

.phone-mockup {
  width: 250px;
  height: 500px;
  background: linear-gradient(145deg, #2d2d2d, #1a1a1a);
  border-radius: 35px;
  padding: 10px;
  box-shadow: var(--shadow-lg);
  position: relative;
}

.phone-screen {
  width: 100%;
  height: 100%;
  background: linear-gradient(180deg, var(--primary-color), var(--primary-dark));
  border-radius: 28px;
  display: flex;
  align-items: center;
  justify-content: center;
  font-size: 3rem;
}

.hero-badge {
  position: absolute;
  top: 20px;
  right: -20px;
  background-color: var(--accent-color);
  color: var(--secondary-color);
  padding: 0.5rem 1rem;
  border-radius: var(--radius);
```

```
font-weight: bold;
transform: rotate(15deg);
box-shadow: var(--shadow);
}

/* ===== DUGMICI ===== */
.btn {
  display: inline-block;
  padding: 0.875rem 1.75rem;
  font-size: 1rem;
  font-weight: 600;
  text-align: center;
  border: none;
  border-radius: var(--radius);
  cursor: pointer;
  transition: var(--transition);
}

.btn-primary {
  background-color: var(--primary-color);
  color: var(--white);
}

.btn-primary:hover {
  background-color: var(--primary-dark);
  transform: translateY(-2px);
  box-shadow: var(--shadow-lg);
}

.btn-secondary {
  background-color: transparent;
  color: var(--white);
  border: 2px solid var(--white);
}

.btn-secondary:hover {
  background-color: var(--white);
  color: var(--secondary-color);
}

.btn-accent {
  background-color: var(--accent-color);
  color: var(--secondary-color);
}
```

```
.btn-accent:hover {
  background-color: #d97706;
}

/* ===== PROIZVODI GRID ===== */
.products-grid {
  display: grid;
  grid-template-columns: repeat(auto-fill, minmax(280px, 1fr));
  gap: 1.5rem;
}
.product-image {
  width: 100%;
  height: 180px;
  display: flex;
  align-items: center;
  justify-content: center;
}
.product-image img {
  max-width: 100%;
  max-height: 100%;
  object-fit: contain;
}

.product-card {
  background-color: var(--white);
  border-radius: var(--radius);
  overflow: hidden;
  box-shadow: var(--shadow);
  transition: var(--transition);
  position: relative;
}
.product-card:hover {
  transform: translateY(-5px);
  box-shadow: var(--shadow-lg);
}

.product-badge {
  position: absolute;
  top: 1rem;
  left: 1rem;
  padding: 0.25rem 0.75rem;
  border-radius: 20px;
  font-size: 0.75rem;
```

```
    font-weight: 600;
    text-transform: uppercase;
}

.badge-new {
    background-color: var(--success-color);
    color: var(--white);
}

.badge-sale {
    background-color: var(--error-color);
    color: var(--white);
}

.badge-popular {
    background-color: var(--accent-color);
    color: var(--secondary-color);
}

.product-image {
    height: 200px;
    display: flex;
    align-items: center;
    justify-content: center;
    background-color: var(--background);
    font-size: 4rem;
}

.product-info {
    padding: 1.25rem;
}

.product-brand {
    font-size: 0.8rem;
    color: var(--text-light);
    text-transform: uppercase;
    letter-spacing: 1px;
}

.product-name {
    font-size: 1.1rem;
    margin: 0.5rem 0;
    color: var(--secondary-color);
}
```

```
.product-specs {  
  font-size: 0.85rem;  
  color: var(--text-light);  
  margin-bottom: 1rem;  
}  
  
.product-price {  
  display: flex;  
  align-items: center;  
  gap: 0.75rem;  
  margin-bottom: 1rem;  
}  
  
.price-current {  
  font-size: 1.25rem;  
  font-weight: bold;  
  color: var(--primary-color);  
}  
  
.price-old {  
  font-size: 0.9rem;  
  color: var(--text-light);  
  text-decoration: line-through;  
}  
  
.product-actions {  
  display: flex;  
  gap: 0.5rem;  
}  
  
.product-actions .btn {  
  flex: 1;  
  padding: 0.625rem 1rem;  
  font-size: 0.875rem;  
}  
  
/* ===== BRENDÖVI ===== */  
.brands-grid {  
  display: grid;  
  grid-template-columns: repeat(auto-fill, minmax(150px, 1fr));  
  gap: 1.5rem;  
}  
  
.brand-card {  
  background-color: var(--white);
```

```
padding: 2rem;
border-radius: var(--radius);
text-align: center;
box-shadow: var(--shadow);
transition: var(--transition);
cursor: pointer;
}

.brand-card:hover {
  transform: scale(1.05);
  box-shadow: var(--shadow-lg);
}

.brand-icon {
  font-size: 2.5rem;
  margin-bottom: 0.5rem;
}

.brand-name {
  font-weight: 600;
  color: var(--secondary-color);
}

.brand-count {
  font-size: 0.85rem;
  color: var(--text-light);
}

/* ===== PREDNOSTI ===== */
.features-grid {
  display: grid;
  grid-template-columns: repeat(auto-fill, minmax(250px, 1fr));
  gap: 1.5rem;
}

.feature-card {
  background-color: var(--white);
  padding: 2rem;
  border-radius: var(--radius);
  text-align: center;
  box-shadow: var(--shadow);
  transition: var(--transition);
}

.feature-card:hover {
```

```
    transform: translateY(-3px);
}

.feature-icon {
  width: 60px;
  height: 60px;
  background: linear-gradient(135deg, var(--primary-color), var(--primary-dark));
  border-radius: 50%;
  display: flex;
  align-items: center;
  justify-content: center;
  margin: 0 auto 1rem;
  font-size: 1.5rem;
}

.feature-title {
  margin-bottom: 0.5rem;
}

.feature-description {
  color: var(--text-light);
  font-size: 0.9rem;
}

/* ===== FORMA ===== */
#contact-section {
  background-color: var(--white);
}

#contact-form {
  max-width: 700px;
  margin: 2rem auto 0;
}

.form-row {
  display: grid;
  grid-template-columns: repeat(auto-fit, minmax(250px, 1fr));
  gap: 1.5rem;
}

.form-group {
  margin-bottom: 1.5rem;
}

.form-group label {
```

```
display: block;
margin-bottom: 0.5rem;
font-weight: 500;
color: var(--secondary-color);
}

.form-group input[type="text"],
.form-group input[type="email"],
.form-group input[type="tel"],
.form-group select,
.form-group textarea {
  width: 100%;
  padding: 0.75rem 1rem;
  border: 2px solid var(--border-color);
  border-radius: var(--radius);
  font-size: 1rem;
  font-family: inherit;
  transition: var(--transition);
}

.form-group input:focus,
.form-group select:focus,
.form-group textarea:focus {
  outline: none;
  border-color: var(--primary-color);
  box-shadow: 0 0 0 3px rgba(26, 86, 219, 0.1);
}

.form-group input.error,
.form-group select.error,
.form-group textarea.error {
  border-color: var(--error-color);
}

.form-group input.success,
.form-group select.success,
.form-group textarea.success {
  border-color: var(--success-color);
}

.error-message {
  display: block;
  color: var(--error-color);
  font-size: 0.85rem;
  margin-top: 0.25rem;
```

```
min-height: 1.25rem;
}

/* Range input */
input[type="range"] {
  width: 100%;
  height: 8px;
  border-radius: 4px;
  background: var(--border-color);
  outline: none;
  -webkit-appearance: none;
}

input[type="range"]::-webkit-slider-thumb {
  -webkit-appearance: none;
  width: 20px;
  height: 20px;
  border-radius: 50%;
  background: var(--primary-color);
  cursor: pointer;
}

output {
  display: block;
  text-align: center;
  margin-top: 0.5rem;
  font-weight: 600;
  color: var(--primary-color);
}

/* Checkbox group */
.checkbox-group {
  display: grid;
  grid-template-columns: repeat(auto-fill, minmax(150px, 1fr));
  gap: 0.75rem;
}

.checkbox-label {
  display: flex;
  align-items: center;
  gap: 0.5rem;
  cursor: pointer;
  font-weight: normal;
}
```

```
.checkbox-label input[type="checkbox"] {
  width: 18px;
  height: 18px;
  accent-color: var(--primary-color);
}

/* Form success */
.form-success {
  text-align: center;
  padding: 2rem;
  background-color: #ecfdf5;
  border-radius: var(--radius);
  border: 2px solid var(--success-color);
}

.form-success h3 {
  color: var(--success-color);
  margin-bottom: 1rem;
}

.form-success-icon {
  font-size: 3rem;
  margin-bottom: 1rem;
}

.submitted-data {
  text-align: left;
  margin-top: 1.5rem;
  padding: 1rem;
  background-color: var(--white);
  border-radius: var(--radius);
}

.submitted-data h4 {
  margin-bottom: 0.75rem;
  color: var(--secondary-color);
}

.submitted-data p {
  margin: 0.5rem 0;
  font-size: 0.9rem;
}

.submitted-data strong {
  color: var(--secondary-color);
```

```
}

/* ===== FOOTER ===== */
#footer {
  background-color: var(--secondary-color);
  color: var(--white);
  padding: 3rem 0 1rem;
}

.footer-content {
  display: grid;
  grid-template-columns: repeat(auto-fit, minmax(200px, 1fr));
  gap: 2rem;
  margin-bottom: 2rem;
}

.footer-section h3 {
  color: var(--white);
  margin-bottom: 1rem;
  font-size: 1.1rem;
}

.footer-section ul {
  list-style: none;
}

.footer-section ul li {
  margin-bottom: 0.5rem;
}

.footer-section ul li a {
  color: rgba(255, 255, 255, 0.7);
  transition: var(--transition);
}

.footer-section ul li a:hover {
  color: var(--accent-color);
}

.footer-section p {
  color: rgba(255, 255, 255, 0.7);
  font-size: 0.9rem;
}

.social-links {
```

```
display: flex;
gap: 1rem;
margin-top: 1rem;
}

.social-link {
width: 40px;
height: 40px;
background-color: rgba(255, 255, 255, 0.1);
border-radius: 50%;
display: flex;
align-items: center;
justify-content: center;
transition: var(--transition);
}

.social-link:hover {
background-color: var(--primary-color);
}

.footer-bottom {
text-align: center;
padding-top: 1.5rem;
border-top: 1px solid rgba(255, 255, 255, 0.1);
color: rgba(255, 255, 255, 0.5);
font-size: 0.9rem;
}

/* ===== MODAL ===== */
.modal {
position: fixed;
top: 0;
left: 0;
width: 100%;
height: 100%;
background-color: rgba(0, 0, 0, 0.5);
display: flex;
align-items: center;
justify-content: center;
z-index: 1000;
opacity: 1;
transition: var(--transition);
}

.modal.hidden {
```

```
    opacity: 0;
    pointer-events: none;
}

.modal-content {
    background-color: var(--white);
    border-radius: var(--radius);
    max-width: 500px;
    width: 90%;
    max-height: 80vh;
    overflow-y: auto;
    position: relative;
    transform: scale(1);
    transition: var(--transition);
}

.modal.hidden .modal-content {
    transform: scale(0.9);
}

.modal-close {
    position: absolute;
    top: 1rem;
    right: 1rem;
    font-size: 1.5rem;
    cursor: pointer;
    color: var(--text-light);
    transition: var(--transition);
}

.modal-close:hover {
    color: var(--error-color);
}

#modal-body {
    padding: 2rem;
}

.modal-product-image {
    text-align: center;
    font-size: 5rem;
    padding: 2rem;
    background-color: var(--background);
    border-radius: var(--radius);
    margin-bottom: 1.5rem;
```

```
}

.modal-product-info h2 {
  margin-bottom: 0.5rem;
}

.modal-product-info .product-brand {
  margin-bottom: 1rem;
}

.modal-specs {
  margin: 1.5rem 0;
}

.modal-specs h4 {
  margin-bottom: 0.75rem;
}

.modal-specs ul {
  list-style: none;
}

.modal-specs ul li {
  padding: 0.5rem 0;
  border-bottom: 1px solid var(--border-color);
  display: flex;
  justify-content: space-between;
}

/* ===== UTILITY KLASÉ ===== */
.hidden {
  display: none !important;
}

.text-center {
  text-align: center;
}

.section-subtitle {
  color: var(--text-light);
  margin-bottom: 2rem;
}

/* ===== RESPONSIVE ===== */
@media (max-width: 768px) {
```

```
h1 {
  font-size: 2rem;
}

h2 {
  font-size: 1.5rem;
}

.header-info {
  display: none;
}

.mobile-menu-toggle {
  display: block;
}

.nav-menu {
  position: absolute;
  top: 100%;
  left: 0;
  right: 0;
  background-color: var(--white);
  flex-direction: column;
  box-shadow: var(--shadow-lg);
  max-height: 0;
  overflow: hidden;
  transition: var(--transition);
}

.nav-menu.active {
  max-height: 400px;
}

.nav-menu li a {
  padding: 1rem 1.5rem;
  border-bottom: 1px solid var(--border-color);
}

.nav-search {
  display: none;
}

.hero-content {
  flex-direction: column;
  text-align: center;
```

```
}

.hero-text h1 {
  font-size: 2rem;
}

.hero-buttons {
  justify-content: center;
  flex-wrap: wrap;
}

.phone-mockup {
  width: 180px;
  height: 360px;
}

.form-row {
  grid-template-columns: 1fr;
}
}

/* toast div */
#toast {
  position: fixed;
  bottom: 20px;
  right: 20px;
  background: #333;
  color: #fff;
  padding: 1rem 1.5rem;
  border-radius: 8px;
  opacity: 0;
  transition: opacity 0.3s, transform 0.3s;
  z-index: 9999;
}

#toast.show {
  opacity: 1;
  transform: translateY(0);
}

#toast.hidden {
  opacity: 0;
  transform: translateY(20px);
}

/* Autor */
.author-container {
  max-width: 600px;
```

```
margin: 5rem auto;
background: #fff;
padding: 2rem;
text-align: center;
border-radius: 12px;
box-shadow: 0 5px 15px rgba(0,0,0,0.1);
}

.author-image {
  width: 150px;
  height: 150px;
  border-radius: 50%;
  object-fit: cover;
  margin-bottom: 1rem;
  border: 3px solid #007bff;
}

.author-name {
  font-size: 1.8rem;
  font-weight: bold;
  margin-bottom: 0.5rem;
}

.author-index {
  font-size: 1rem;
  color: #555;
  margin-bottom: 1.5rem;
}

.author-text {
  font-size: 1rem;
  line-height: 1.6;
  color: #333;
}

.btn-back {
  display: inline-block;
  margin-top: 2rem;
  padding: 0.6rem 1.2rem;
  background-color: #007bff;
  color: #fff;
  text-decoration: none;
  border-radius: 6px;
  transition: 0.3s;
}
```

```
.btn-back:hover {
    background-color: #0056b3;
}

/* ===== ANIMACIJE ===== */
@keyframes fadeIn {
    from {
        opacity: 0;
        transform: translateY(20px);
    }
    to {
        opacity: 1;
        transform: translateY(0);
    }
}

.animate-in {
    animation: fadeIn 0.5s ease forwards;
}

@keyframes pulse {
    0%,
    100% {
        transform: scale(1);
    }
    50% {
        transform: scale(1.05);
    }
}

.pulse {
    animation: pulse 2s infinite;
}
```