



# Predlog plana za profesionalni višestranični portfolio sajt (digitalni CV)

**Opis:** Ovaj dokument predstavlja detaljan plan za izradu personalnog portfolio sajta (digitalnog CV-a) za programera-početnika. Plan obuhvata strukturu stranica, organizaciju HTML/CSS koda, potrebne sekcije i funkcionalnosti, kao i savete za pristupačnost, responzivnost i korisničko iskustvo.

## Struktura sajta – stranice i sekcije

Predloženi portfolio biće podeljen na više povezanih **HTML stranica**, što omogućava jasno razdvajanje sadržaja i jednostavnu navigaciju. Glavne stranice i njihove uloge su:

- **Home (Početna):** Uvodna stranica koja predstavlja vaš lični brending. Sadrži **kratak pozdrav ili slogan**, vaše ime i profesiju, uz možda *hero* sliku ili ilustraciju. Treba da ostavi dobar prvi utisak i može da uključi *call-to-action* (npr. dugme „Pogledaj moje projekte“ ili „Preuzmi CV“).
- **About (O meni):** Detaljnija biografija i pozadina. Ova sekcija uključuje ko ste, **vaše obrazovanje i iskustvo**, koje tehnologije poznajete i **vašu stručnost**. Idealno je navesti i lokaciju, zanimljive detalje o sebi i profesionalne ciljeve <sup>1</sup>. Ovaj deo može sadržati i vašu profilnu fotografiju.
- **Timeline (Vremenska linija)** vaše karijere/obrazovanja može biti uključen ovde ili kao posebna stranica – hronološki prikaz ključnih događaja (škola, kursevi, poslovi, projekti) radi preglednosti.
- **Skills (Veštine):** Lista vaših tehničkih veština i tehnologija koje poznajete. Preporučljivo je **grupisati veštine** po kategorijama (npr. *Programski jezici, Framework-i, Alati*) i uz svaku navesti nivo znanja ili iskustva. Možete dodati **ikone ili oznake** za svaku tehnologiju (npr. HTML5, CSS3, JavaScript, Python, Git itd.) radi vizuelnog naglašavanja <sup>2</sup>. Takođe je korisno pomenuti relevantne **sertifikate ili kurseve** koje ste završili radi kredibiliteta <sup>3</sup>.
- **Projects (Projekti):** Portfolij projekata koje ste radili. Prikažite **samo najbolje radove** – kvalitet je bitniji od kvantiteta <sup>4</sup>. Svaki projekat treba da bude predstavljen kratkim opisom, spiskom ključnih tehnologija (označenih ikonicama ili tagovima), slikom/screenshotom i linkovima ka živom demou ili GitHub repozitorijumu <sup>5</sup>. Ova stranica može biti organizovana kao mreža kartica (*cards*) ili lista projekata. Cilj je da posetilac brzo stekne uvid u vaš rad i dostignuća.
- **Timeline (Iskustvo):** (Opcionalno, ako nije uklapljeno u About) Posebna stranica sa hronološkim prikazom vaše edukacije i radnog iskustva. Navedite po godinama (ili periodima) ključne stavke: školovanje, diplome, poslove, stažiranja, volonterski rad i sl. Svaka stavka treba da sadrži **naziv pozicije ili programa**, naziv institucije/kompanije i vremenski period. Ovakva **sekcija iskustva** ističe šta ste radili i kada <sup>6</sup>. Može se dizajnirati kao vertikalna linija sa čvoristima (događajima) radi atraktivnosti.
- **Blog:** (Opcionalno) Stranica na kojoj možete objavljivati članke ili pisati o svojim projektima i znanju. Blog može doprineti demonstraciji vašeg stručnog znanja i strasti prema oblasti <sup>7</sup>. Na **Blog** stranici prikažite liste vaših postova sa naslovom, kratkim opisom i datumom. Svaki post bi bio poseban HTML (npr. `blog-post1.html`) ili link ka eksternom blogu ako ga imate. Ova sekcija je *nice-to-have* za lični brend, ali nije obavezna tek započinjete (možete je predvideti za kasniju fazu razvoja).
- **Contact (Kontakt):** Stranica koja omogućava posetiocima da vas lako kontaktiraju. Sadrži **kontakt formu** sa poljima kao što su ime, email i poruka, kao i dugme za slanje. Forma može biti funkcionalna (uz pomoć servisa ili skripte) ili slati na vaš email (`mailto:link`) u najjednostavnijem obliku. Pored forme, uključite i direktnе kontakte: **email adresu (kao link)**,

eventualno broj telefona, kao i **linkove ka vašim profilima na društvenim mrežama** (GitHub, LinkedIn, Twitter...) u vidu ikonica. Važno je da mogućnost kontaktiranja bude istaknuta – cilj portfolija je da olakša potencijalnim poslodavcima ili klijentima da stupe u vezu sa vama <sup>8</sup>. Formu postavite **istaknuto** (npr. na vrh kontakt stranice ili kao poseban deo homepage-a) <sup>9</sup>.

- **Dodatne sekcije:** Moguće je dodati i druge elemente po potrebi, npr. **Testimonials (Preporuke)** – kratke izjave mentora, kolega ili klijenata koje potvrđuju vaše veštine i rad (ovo daje *social proof* i povećava poverenje <sup>9</sup>). Testimonials se mogu ubaciti na početnu ili about stranicu. Takođe, možete imati i sekciju **Interests/Hobbies (Lična interesovanja)** za ličniji dodir, ali držite fokus na profesionalnim aspektima.

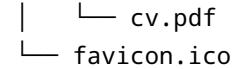
**Navigacija:** Predvideti **fiksni meni (navigacionu traku)** na vrhu svake stranice, kako bi korisnik mogao lako da prelazi između sekcija. Navigacija treba da sadrži linkove ka navedenim stranicama: Home, About, Skills, Projects, (Timeline), Blog, Contact. Poželjno je da link ka stranici na kojoj se korisnik trenutno nalazi bude vizuelno istaknut (npr. podvučen ili drugom bojom) radi orientacije. Na mobilnim uređajima, klasični horizontalni meni treba prilagoditi – često se koristi **"hamburger" ikona** koja otvara vertikalni meni. Svaka stranica će imati dosledan **header** (zaglavje sa navigacijom i vašim imenom/logo-om) i **footer** (podnožje). U footer možete staviti npr. autorska prava, kratku poruku ili ponoviti linkove ka društvenim mrežama. Time obezbeđujete konzistentnost dizajna na svim stranicama.

**Napomena:** Višestrajnični pristup znači da će se izvesni elementi (poput navigacije i footer-a) ponavljati na svakom HTML fajlu. To zahteva ručno održavanje istih elemenata na više mesta – npr. ako dodate novi link u navigaciju, morate ga dodati u svim stranicama <sup>10</sup>. Ovo je za manje sajtove prihvatljivo, ali imajte na umu da se kod većih projekata koristi templating ili frameworks da bi se ova duplikacija izbegla. Za početak, statički HTML pristup sa odvojenim stranicama je sasvim u redu <sup>10</sup>.

## Struktura fajlova i direktorijuma

Organizacija fajlova doprinosi preglednosti koda i lakšem održavanju. Preporučena struktura projekta na GitHub-u:

```
project-folder/
├── index.html      (Početna strana)
├── about.html      (O meni)
├── skills.html     (Veštine)
├── projects.html   (Projekti)
├── timeline.html   (Timeline iskustva)
├── blog.html        (Lista blog postova, ako postoji)
├── contact.html    (Kontakt)
├── css/
│   └── styles.css   (Glavni stilovi za ceo sajt)
└── js/
    └── scripts.js    (Glavni JavaScript – za buduće funkcionalnosti poput
teme, forma)
├── assets/
│   ├── images/       (Slike: npr. profile.jpg, projekti screenshots...)
│   ├── icons/        (Ikonice ili logo, ako ih čuvate lokalno)
│   └── fonts/        (Custom fontovi ako se ne koriste CDN/Google Fonts)
```



(Vaš CV u PDF formatu za preuzimanje)  
(Favicon ikonica za sajt)

Ovakva struktura grupiše sličan sadržaj zajedno: svi mediji (slike, ikonice, dokumenta) nalaze se u `assets/` folderu, stilovi u `css/`, skripte u `js/`, dok su HTML stranice u korenu projekta radi jednostavnih URL-ova <sup>11</sup> <sup>12</sup>. Centralizacijom resursa, olakšavate održavanje – npr. sve slike su na jednom mestu. Favicon datoteku možete smestiti u root (radi podrazumevane putanje) ili unutar assets (uz odgovarajuće `<link>` u HTML).

GitHub Pages hosting zahteva da početna strana bude dostupna kao `index.html` u korenskom direktorijumu ili odgovarajućem `gh-pages` ogranku. Prema tome, navedena struktura je kompatibilna – samo objavite čitav projekat na GitHub repozitorijum i omogućite Pages (najčešće sa `main branch / root` foldera). Nakon deploy-a, vaš sajt će biti dostupan na adresi tipa `korisnickoime.github.io/naziv-projekta` (ili `korisnickoime.github.io` ako koristite tzv. user/organization sajt).

## HTML – semantika i SEO dobre prakse

Struktura HTML koda treba da bude **čista, semantička i optimizovana za pretraživače**. To znači da koristite odgovarajuće HTML5 **semantičke elemente** umesto generičkih `<div>` gde god je to moguće. Na primer:

- `<header>` za zaglavlje stranice ili sekcije (obično sadrži navigaciju ili naslov).
- `<nav>` za blok navigacionih linkova (meni).
- `<main>` za glavni sadržaj stranice (npr. sadržaj koji je unikatan za tu stranicu).
- `<section>` za logičke celine sadržaja unutar glavnog dela (npr. odeljci poput About, Skills na jednoj stranici).
- `<article>` za nezavisne, samostalne komponente sadržaja, kao što su pojedinačni blog postovi ili stavke na listi projekata <sup>13</sup> <sup>14</sup>. (Svaki projekat se može posmatrati kao `article`, jer ima smisla i sam za sebe, dok je npr. About sekcija bolje da ostane `<section>`).
- `<footer>` za podnožje stranice ili sekcije (kontakt informacije, copyright i sl.).

Korišćenje semantičkih elemenata umesto samo `<div>` elemenata poboljšaće **čitljivost koda**, olakšati stilizovanje i značajno doprineti **pristupačnosti i SEO** rejtingu <sup>15</sup>. Pretraživači i alati za pristupačnost (čitači ekrana) bolje razumeju strukturu strane kada nailaze na ove elemente, a ne na bezimene kontejnere <sup>15</sup>. Na primer, `<main>` pomaže pretraživaču da shvati šta je glavni sadržaj, dok `<nav>` označava navigaciju.

Osnovni *template* svakog HTML fajla treba da uključi:

- **Deklaraciju doctype i jezik stranice:** `<!DOCTYPE html>` na vrhu i atribut `<html lang="en">` (ili odgovarajući jezik, npr. `lang="sr"` za srpski) u `<html>` tagu. Definisanje jezika je važno za pretraživače i čitače ekrana kako bi ispravno interpretirali sadržaj <sup>16</sup>.
- **Meta karakter set:** `<meta charset="UTF-8">` za podršku naših slova i ispravan prikaz teksta.
- **Meta viewport:** `<meta name="viewport" content="width=device-width, initial-scale=1.0">` radi pravilnog skaliranja stranice na mobilnim uređajima <sup>16</sup>. Ovo je ključno za **responzivnost** – bez ovog taga, stranice se mogu čudno prikazivati na telefonima.
- **Naslov stranice (title):** Unutar `<head>` elementa, postavite opisni `<title>` za svaku stranicu (npr. "Ime Prezime – About" ili "Ime – Projekti"). Title tag se prikazuje u kartici braузera i utiče na SEO, pa treba da bude kratak i jasan.

- **Opis (meta description):** Za svaku stranicu dodajte `<meta name="description" content="...>` sa kratkim opisom sadržaja (oko 150 znakova). Ovaj opis koriste pretraživači u rezultatima pretrage, pa treba da bude specifičan i privlačan <sup>17</sup>. (Primer: za projekte: "Portfolio projekti i dostignuća Ime Prezime, junior web developer-a").
- **(Opcionalno) Meta keywords:** Nekad se koriste ključne reči (`<meta name="keywords" content="portfolio, front-end, developer,...>`), ali moderni pretraživači uglavnom ignoriraju ovaj tag. Fokusirajte se više na kvalitet sadržaja i relevantnost.
- **Open Graph meta tagovi (socijalni preview):** Kao dopunu, možete ubaciti **OG tagove** da kontrolišete kako će se stranice prikazati kada se dele na društvenim mrežama (Facebook, LinkedIn) ili u chat aplikacijama. Na primer, za početnu stranicu:

```

<meta property="og:title" content="Ime Prezime - Web Developer Portfolio">
<meta property="og:description" content="Junior web programer, traži prilike za rad. Pogledajte moj online CV i projekte.">
<meta property="og:image" content="https://example.com/assets/preview.png">
<meta property="og:url" content="https://korisnickoime.github.io/">
<meta property="og:type" content="website">

```

Ovi tagovi poboljšavaju prikaz linka ka vašem sajtu (npr. ubaciće se naslov, opis i sličica kad neko podeli link) <sup>18</sup>. Slično, postoji i skup `<meta name="twitter:...>` tagova za Twitter kartice <sup>19</sup> – nisu neophodni, ali su **lep dodatak** za profesionalni utisak.

- **Link ka favicon-u:** Dodajte u head: `<link rel="shortcut icon" href="assets/favicon.ico" type="image/x-icon">`. Favicon (16x16 ili 32x32 ikonica) će se pojaviti na tabu brauzera pored naslova stranice <sup>20</sup>. Ovo doprinosi **prepoznatljivosti** vašeg sajta u moru otvorenih tabova.

Na početku `<body>` dela, obično se uključuje jedan **<header>** element koji sadrži logo/ime i navigaciju (meni lista `<ul>` sa `<a>` linkovima ka stranicama). Nakon toga, koristite **<main>** za glavni sadržaj te stranice podeljen u semantičke **<section>** elemente kako je ranije opisano. Na kraju `<body>` obično стоји **<footer>** sa kontaktnim informacijama ili kopirajtom.

**Primer skeleton strukture (HTML)** – za ilustraciju, struktura `about.html` stranice mogla bi izgledati ovako (skraćeno):

```

<body>
  <header>
    <nav>
      <ul>
        <li><a href="index.html">Home</a></li>
        <li><a href="about.html" class="active">About</a></li>
        <li><a href="projects.html">Projects</a></li>
        ... <!-- ostali linkovi -->
      </ul>
    </nav>
  </header>

  <main>

```

```

<section id="about-me">
    <h1>O meni</h1>
    <p>... kratka biografija ...</p>
</section>
<section id="timeline">
    <h2>Moje iskustvo</h2>
    <!-- stavke vremenske linije, npr. lista ili custom timeline markup -->
</section>
<section id="skills">
    <h2>Veštine</h2>
    <!-- lista ikonica ili stavki sa veštinama -->
</section>
<!-- druge sekcije po potrebi -->
</main>

<footer>
    <p>&copy; 2025 Ime Prezime</p>
    <div class="social-links">
        <!-- ikonice sa linkovima na društvene mreže -->
    </div>
</footer>
</body>

```

Obratite pažnju da svaki `<section>` ima **id** ako želimo navigaciju do sidro-tačke (pogotovo korisno za one-page scenario ili skip-linkove). U multi-page pristupu nije obavezno da sekcije imaju id, osim ako ne planirate scrollspy ili deljenje direktnih linkova ka sekciji. ID-ove koristite pažljivo – moraju biti jedinstveni na stranici <sup>21</sup>. Za stilizaciju i skripte bolje se oslanjati na **classe**, a ID samo gde je potrebno jedinstveno referenciranje.

Još par SEO saveta:

- Koristite ispravno hijerarhiju naslova – npr. **jedan** `<h1>` **po stranici** (najvažniji naslov, obično na početnoj to može biti vaše ime i tagline, na About stranici "O meni", itd.), zatim `<h2>` za glavne sekcije, `<h3>` za podsekcije unutar njih itd. Ne preskačite nivoe naslova radi stilizovanja – umesto toga koristite CSS za veličinu fonta. Pravilna struktura naslova takođe doprinosi dostupnosti. - **Alt opisi za slike:** Sve `<img>` elemente obavezno opremite atributom `alt="opis slike"`. Opis treba biti kratak ali smislen (npr. `alt="Fotografija Ime Prezime"` za profilnu sliku). Time sadržaj slike postaje dostupan korisnicima koji ne vide slike i poboljšava SEO <sup>22</sup>. Ako neka slika služi isključivo dekoraciji, alt može biti prazan (ali atribut mora postojati). - **Struktura linkova:** Eksterni linkovi (ka drugim sajtovima, npr. vaš GitHub profil ili live demo projekata) mogu se otvarati u novom tabu koristeći `target="_blank"`. Ako to radite, dodajte i `rel="noopener"` radi bezbednosti. Takođe, radi pristupačnosti, kada link čini samo ikonica (bez teksta), obavezno dodajte **atribut** `aria-label` sa tekstualnim opisom linka (npr. `<a href="https://github.com/username" target="_blank" rel="noopener" aria-label="GitHub profil">[ikonica]</a>`). Na taj način čitači ekrana mogu da najave svrhu linka <sup>23</sup> <sup>24</sup>. - **SEO optimizovan sadržaj:** Trudite se da tekstovi na sajtu uključuju ključne reči vezane za pozicije ili tehnologije koje vas zanimaju (npr. "front-end developer", "JavaScript", "React", itd.), ali prirodno utkane u rečenice. To pomaže da vas poslodavci lakše pronađu putem pretrage, iako je portfolio mali sajt.

## CSS stilovi – organizacija, responzivnost i teme

Sav CSS kod smestite u poseban fajl (npr. `css/styles.css`) i povežite ga u `<head>` svih HTML stranica ovim redosledom: `<link rel="stylesheet" href="css/styles.css">`<sup>25</sup>. Korišćenje **externog CSS fajla** olakšava održavanje i osigurava konzistentan izgled na svim stranicama – promenom u `styles.css` promeniće se stil svuda. Takođe, browser će keširati CSS, što poboljšava učitavanje.

**Organizacija CSS-a:** možete organizovati kod logički – npr. na početku definisati **CSS reset/normalize**, zatim osnovne promenljive i globalne stilove (body, tipografija), pa stilove za layout (npr. grid sistemi, flex containeri), i na kraju stilove specifičnih komponenti i sekcija. Za manje projekte nije nužno razbijati CSS u mnogo fajlova, ali možete koristiti komentare ili sekcije unutar jednog fajla za preglednost (`// ===== Navigation =====, // ===== Footer =====, itd.`). Bitno je da budete **dosledni u imenovanju klasa** za HTML elemente, kako bi se odmah iz imena videlo na šta se odnosi (npr. `.nav-bar`, `.project-card`, `.btn-primary` itd.). Popularna metodologija za imenovanje je **BEM** (Block-Element-Modifier), ali u malom projektu i jednostavna konvencija sa smislom će raditi posao.

**CSS promenljive (Custom Properties):** Koristite CSS promenljive za definisanje **paleta boja i drugih ponavljamajućih vrednosti**. Na vrhu CSS fajla (ili unutar `:root` selektora) definišite glavne boje, fontove i dimenzije, npr.:

```
:root {  
    --color-bg: #ffffff;  
    --color-text: #333333;  
    --color-primary: #4a90e2;  
    --color-secondary: #d94f5c;  
    --font-family-base: 'Inter', sans-serif;  
    --spacing: 16px;  
    /* ... */  
}
```

Promenljive omogućavaju da na jednom mestu promenite npr. primarnu boju sajta, a da se ta promena reflektuje na svim mestima gde je korišćena (header, dugmad, linkovi, itd.)<sup>26</sup>. Takođe, imena promenljivih mogu biti **semantička** (kao gore `--color-primary`) što olakšava razumevanje uloga boja za razliku od korišćenja heks kodova svuda<sup>26</sup>. U nastavku CSS-a, koristite ove promenljive sa `var()`, npr. `background-color: var(--color-bg); color: var(--color-text);`.

**Responzivni dizajn:** Dizajnirajte *mobile-first*, što znači da CSS pišete prvo za male ekrane (telefon), a zatim koristite **media queries** za veće širine ekranata da prilagodite izgled za tablet i desktop. Primer:

```
/* Mobile (default) */  
.nav ul {  
    flex-direction: column; /* vertikalni meni */  
}  
/* Desktop */  
@media (min-width: 768px) {  
.nav ul {  
    flex-direction: row; /* horizontalni meni za šire ekranate */  
}
```

```
}
```

Ovde je `.nav ul` podrazumevano kolona (što odgovara mobilnom prikazu sa hamburger menijem), a na `>=768px` prelazi u red (klasičan meni). Podesite **breakpoint** vrednosti prema sadržaju – npr. ako vidite da se layout “lomi” na ~600px, tu ubacite `@media (min-width: 600px)`.

Uz to, koristite fleksibilne jedinice: **procenat**, `em`, `rem`, `vh/vw` kada je moguće, umesto fiksnih piksela. Na primer, širinu kolona možete dati u %, slike neka imaju `max-width: 100%` da bi se skalirale u manjim okvirima <sup>27</sup>, fontove možete definisati relativno prema `rem` (gde je 1rem bazna veličina fonta, obično ~16px). Ovo omogućava lakšu prilagodbu različitim veličinama ekrana i korisničkim podešavanjima.

Ne zaboravite meta viewport (spomenut gore) bez koga responzivnost neće funkcionisati kako treba. Takođe, koristite **flexbox** i **CSS grid** za rasporede elemenata umesto starih tehnika (float, table, itd.). Flexbox je odličan za linearno poređane elemente (navigacije, liste kartica), dok je grid koristan za složenije mrežne rasporede (galerija projekata). Obe tehnologije su responzivne i ponašaju se predvidivo uz malo podešavanja.

**Primer responzivnog dizajna:** Projekti možete prikazati u grid-u od tri kolone na desktopu, dve kolone na tabletima i jedne kolone na mobilnom. CSS može izgledati ovako:

```
.projects-grid {
  display: grid;
  gap: 1rem;
  grid-template-columns: 1fr; /* 1 kolona mobilni */
}

@media (min-width: 600px) {
  .projects-grid { grid-template-columns: 1fr 1fr; } /* 2 kolone */
}

@media (min-width: 900px) {
  .projects-grid { grid-template-columns: 1fr 1fr 1fr; } /* 3 kolone */
}
```

Na ovaj način, layout se *fluidno* prilagođava širini ekrana.

**Light i Dark tema:** Danas je moderno da portfolio podržava tamni režim prikaza. Ovo možete ostvariti uz minimalan trud zahvaljujući CSS promenljivima. Već ste definisali skup boja u :root za podrazumevanu (recimo svetlu) temu. Potom dodajte CSS za tamnu temu – najjednostavnije je iskoristiti `prefers-color-scheme` medija upit, koji automatski primenjuje tamnu temu ako korisnik sistema to preferira:

```
@media (prefers-color-scheme: dark) {
  :root {
    --color-bg: #1e1e1e;
    --color-text: #e4e4e4;
    --color-primary: #5dade2;
    --color-secondary: #e59866;
```

```
    /* ... override drugih promenljivih za tamnu temu ... */  
}
```

Ovaj pristup će automatski zameniti vrednosti promenljivih kada je tamni mod aktivan na nivou operativnog sistema ili brauzera <sup>28</sup>. Alternativno ili dodatno, možete implementirati **ručni prekidač teme** na sajtu (ikonica ili dugme koje korisnik klikne da promeni temu). U tom slučaju, jedna od tehnika je dodavanje klase na <body> npr. `.dark-theme` i u CSS-u definisanje alternativnih vrednosti pod tom klasom, ili čak uključivanje posebnog CSS fajla za dark theme. Ručni prekidač zahteva JavaScript koji će na klik dodati/ukloniti klasu i eventualno sačuvati preferencu u *localStorage* da bi se zapamtila poseta. Mnoge moderne portfolio teme implementiraju ovu funkcionalnost zbog popularnosti <sup>29</sup>. U kontekstu ovog projekta, možete krenuti sa `prefers-color-scheme` implementacijom (koja daje instant rezultat bez ijedne linije JS), a kasnije dodati i toggle dugme za veću kontrolu posetioца.

**Tipografija i web fontovi:** Izbor fonta dosta utiče na utisak. Google Fonts je jednostavno rešenje – odaberite čitljiv **web font** (npr. Roboto, Lato, Poppins, Inter ili sl. – za CV/portfolio obično sans-serif porodica radi modernog izgleda). Google Fonts vam generiše `<link>` tag koji ubacite u <head> pre svog CSS-a, na primer:

```
<link href="https://fonts.googleapis.com/css2?  
family=Roboto:wght@400;700&display=swap" rel="stylesheet">
```

Nakon toga u CSS-u postavite `font-family: 'Roboto', sans-serif;` za body ili .root, i time će ceo sajt koristiti taj font. **Napomena:** `display=swap` parametar obezbeđuje da se prikaže fallback font dok se Google font ne učita, što sprečava *flash of invisible text*. Uvek navedite rezervnu porodicu (npr. sans-serif) posle glavnog fonta za slučaj da font ne može da se učita.

Koristite različite **težine fonta** (npr. 400 za normal, 700 za bold) u skladu sa dizajnom – npr. naslovi mogu biti bold, telo teksta regular. Nemojte preterivati sa previše različitih fontova; 1–2 fonta ukupno su dovoljna da sajt izgleda profesionalno i jedinstveno.

**Boje i kontrast:** Odaberite paletu boja koja je prijatna i ima dobar kontrast. Tipično, 2–3 osnovne boje su dovoljne (primarna za akcente – npr. naglašena dugmad, linkove; sekundarna za manje naglašene elemente ili kao boja isticanja u pozadini sekcija; neutralne za tekst i pozadinu). Vodite računa o **kontrastu teksta i pozadine** radi čitljivosti – npr. tamno siva slova na beloj pozadini ili svetlosiva na tamnoj su u redu, ali svetložuta na belo ne bi imala dobar kontrast. Preporučuje se kontrastni odnos minimum 4.5:1 za tekst vs pozadina (WCAG standard). Postoje online alati za proveru kontrasta boja.

**Ikonice:** Pored teksta, ikonice će vizuelno obogatiti sajt. Već smo spomenuli korišćenje ikona za veštine (logoi jezika i alata) i za linkove ka društvenim mrežama. Možete koristiti **ikon fontove** (kao što je *Font Awesome* ili *Bootstrap Icons*), koje uključite preko CDN `<script>` ili `<link>`. Alternativno, još bolje, koristite **SVG ikonice** jer su vektorske i prilagodljive. Mnogi logo-i programskih jezika dostupni su kao SVG (npr. preko servisa *Simple Icons* ili putem oficijelnih brend ikona). SVG možete direktno ubaciti u HTML kao `<svg>` kod ili linkovati kao ``.

Na primer, za GitHub ikonicu možete uključiti FontAwesome i onda u HTML staviti `<i class="fab fa-github" aria-hidden="true"></i>` uz odgovarajući aria-label na linku. Ili jednostavno dodati alt tekst na `<img>` ako koristite sliku ikonice. Ikone za tehnologije (HTML5, CSS3,

JS, itd.) možete rasporediti u sekciji veština, a takođe i uz svaki projekat navesti ikonice tehnologija koje su korišćene (to su pomenute "oznake/ikonice" uz projekte).

**Layout detalji:** Za kartice projekata i slične elemente, koristite **card stil** – to obično podrazumeva beli (ili tamni) blok sa blagom senkom ili obrubom, koji na hover može blago da se podigne (efekat senke) kako bi delovao klikabilno. U CSS-u to postižemo sa `box-shadow` i `transition`. Primer:

```
.project-card {  
    background: #fff;  
    border-radius: 8px;  
    padding: 16px;  
    transition: transform 0.3s, box-shadow 0.3s;  
}  
.project-card:hover {  
    transform: translateY(-4px);  
    box-shadow: 0 4px 15px rgba(0,0,0,0.1);  
}
```

Ovakvi mali detalji doprinose modernijem izgledu i boljem **korisničkom iskustvu**, jer korisnik dobija vizuelni feedback da je element interaktivan.

## Pristupačnost (Accessibility)

Pristupačan dizajn znači da sajt mogu ravnopravno koristiti i osobe sa hendikepom (npr. slabovidim osobama pomoću čitača ekrana), ali i bilo koji korisnik u različitim okolnostima (jak odsjaj na ekranu, spor internet, itd.). Mnoge pristupačne prakse su već pomenute, a u nastavku je rezime ključnih:

- **Semantički HTML:** Korišćenje `<header>`, `<nav>`, `<main>`, `<footer>`, `<section>`, `<article>` i ispravne hijerarhije naslova pomaže čitačima ekrana da razumeju strukturu stranice <sup>15</sup>. To smo implementirali u prethodnim sekcijama.
- **Tekst alternativan slikama:** Svaka slika ima `alt`. Ovo je **obavezno** za pristupačnost <sup>22</sup>. Ako slika nosi informaciju (npr. screenshot projekta), alt treba da prenese tu informaciju. Ako je dekorativna, alt="" (prazno) znači da je čitač ekrana može preskočiti.
- **Label-e za formu:** Forme na kontakt stranici treba da imaju `<label>` za svako polje vezan putem `for` atributa ili ugnježđen. Primer:

```
<label for="name">Ime:</label>  
<input type="text" id="name" name="name">
```

Na taj način, korisnik koji koristi čitač ekrana čuje naziv polja kada se fokusira na input. Dodajte i `type="email"` za email polje (što daje i mobilnim korisnicima odgovarajuću tastaturu), `required` atribut gde je potrebno (za HTML5 validaciju), i jasno istaknite `submit` dugme. Takođe, razmotrite poruke o grešci ili potvrdi slanja – može se JS dodati kasnije da prikazuje "Hvala na poruci" ili slično.

- **Fokus i tastatura:** Proverite da se kroz sve linkove i forme može kretati tabulatorom. Podrazumevano, browser omogućava focus na `<a>`, `<button>`, `<input>`. **Ne uklanjajte outline** na fokusiranim elementima *bez da obezbedite zamenu!* Outline (ona plava ili isprekidana linija) je

važna za korisnike koji navigiraju tastaturom. Možete je stilizovati umesto ukloniti. Na primer, globalno:

```
:focus { outline: 2px dashed #00ff19; outline-offset: 4px; }
```

(primer ističe fokusiranu stavku zelenom isprekidanom linijom <sup>30</sup>). Ovo je čak korisno i vidljivim korisnicima da lakše prate gde im je fokus.

- **ARIA oznake:** Već spomenuto korišćenje `aria-label` za linkove koji nemaju tekst (ikonice). Još jedan primer – ako imate *social icons bar* gde su samo ikonice GitHub, LinkedIn itd., dodajte svakoj `<a>` linku atribut `aria-label="LinkedIn profil"` itd. <sup>23</sup>. Tako čitač ekrana umesto "link, image" pročita smislenu informaciju.
- **Kontrast:** Visok kontrast teksta i pozadine (minimum 4.5:1 za regularan tekst). Koristite alate ili ekstenzije (poput Lighthouse, koji je ugrađen u Chrome devtools, ili webaim kontrast proveru) da testirate vaše kombinacije boja. Ovo nije bitno samo za osobe sa slabim vidom, već za sve korisnike pri jakom svetlu ili na prosečnim ekranima.
- **Responsive pristupačnost:** Prilagodljiv dizajn koji smo opisali takođe je deo pristupačnosti – npr. korisnici na mobilnim uređajima moći će lako da vide sadržaj. Takođe, koristite relativne veličine fonta (rem) tako da korisnikovo povećavanje teksta u browseru skalira vaš sajt lepo. Proverite izgled na 200% zoom – sve treba i dalje da funkcioniše (medija upiti se baziraju na širini viewporta, što pokriva i scenario zooma).
- **Preferencije korisnika:** Uvažite korisnička podešavanja poput **prefers-reduced-motion**. Ako uvodite animacije ili parallax efekte, obavezno dodajte CSS medija query: `@media (prefers-reduced-motion: reduce) { /* isključi animacije ili ih učini instant */ }`. Ovo smo videli i u jednom od primer koda – glatko skrolovanje se isključuje ako je uključen reduce motion <sup>31</sup>. Ovo je važan detalj za korisnike koji dobijaju vrtoglavicu od animacija.
- **Testirajte sa čitačem ekrana:** Probajte otvoriti svoj sajt i kretati se isključivo tastaturom (Tab, Shift+Tab, Enter) da vidite da li su sve interakcije moguće. Takođe, koristite neki besplatni čitač (VoiceOver na mac, NVDA na Windows) da proverite da li čita sadržaj logično (npr. da li preskače bitne informacije, kako izgovara navigaciju, itd.). Tako ćete uočiti pristupačnost probleme.

Zapamtite, **pristupačan sajt** je često i **SEO-friendly sajt**, jer tehnike koje pomažu čitačima ekrana obično pomažu i pretraživačima da razumeju vaš sadržaj.

## Korisničko iskustvo (UX) i dodatni saveti

Pored čisto tehničke ispravnosti, vaš portfolio treba da ostavi dobar utisak i bude priјatan za korišćenje. Evo nekoliko smernica za poboljšanje UX-a:

- **Jasna struktura i navigacija:** Kao što smo planirali, sve važne informacije su organizovane u logične celine (stranice/sekcije). Posetioc treba odmah da shvati *ko ste vi i šta nudite*. Na početnoj stranici izdvojite ukratko **ko ste (uloga)** i **šta tražite** (posao, freelance projekte, saradnju) – to je vaš *personal pitch*. Navigacija treba da je **vidljiva i intuitivna**. Koristite uobičajene nazive (About, Projects, Contact...) da se korisnik ne zbuni. Takođe, obezbedite da logo ili ime u header-u vodi na Home (ovo je standardno ponašanje).
- **Konzistentnost dizajna:** Svi delovi sajta treba da deluju kao celina. To znači konzistentna paleta boja, isti fontovi i stilovi dugmadi kroz ceo sajt <sup>32</sup>. Na primer, ako koristite zaobljene ivice kartica na jednoj stranici, koristite ih svuda. Navigacija i footer su isti na svim stranicama. Konzistentnost gradi **profesionalni utisak**.
- **Brzina učitavanja:** GitHub Pages će servirati statički sajt prilično brzo, ali vi možete doprinjeti tako što ćete **optimizovati slike** (npr. smanjite rezoluciju velikih fotografija jer u web verziji retko

treba više od 1200px širine, kompresujte JPEG/PNG datoteke, razmislite o modernim formatima kao WebP). Učitajte samo ono što je potrebno – ako imate 5 projekata, ne treba vam 10MB slika. Takođe, *lazy-load* tehnike (npr. `loading="lazy"` atribut na `<img>`) mogu pomoći da se slike koje nisu odmah u vidokrugu ne učitavaju momentalno.

- **Odzivnost interfejsa:** Podrazumeva se da linkovi i dugmad menjaju stil na hover/focus (npr. promene boju ili opacity) – to korisniku daje vizuelni signal da su interaktivni. Implementirajte CSS *hover* efekte na dugmad, kartice projekata, linkove u meniju. Takođe, obratite pažnju da veličina **klikabilnih elemenata** bude dovoljna (preporučeno barem 44px visine za dugme/link na mobilnom).
- **Istaknite poziv na akciju (CTA):** Razmislite šta želite od posetioca – verovatno da vas kontaktira ili preuzme vaš CV. Stoga, **dugme za preuzimanje CV-a ili link ka PDF-u** postavite na vidljivo mesto, recimo na Home (npr. "Download CV" dugme ispod vaše kratke biografije) i/ili na About stranicu <sup>33</sup>. To dugme treba vizuelno istaći primarnom bojom i ikonicom fajla. Slično, na stranici Projekata možete staviti CTA "Pogledaj moj kod na GitHub-u" ili "Kontaktiraj me za saradnju" itd.
- **Kontakt informacije uvek dostupne:** Neki posetoci će želeti direktno da vas kontaktiraju bez čitanja celog sajta. Zato je zgodno u futeru **na svakoj stranici** navesti email adresu i ikonice mreža. Tako su kontakti **uvek na dohvati ruke** <sup>34</sup>. Možete čak razmotriti *fiksni* element (poput malog dugmeta ili trake) sa kontakt emailom ili ikonicom poruke u čošku, ali to nije obavezno.
- **Testimonijali i rezultati:** Ako imate bilo kakve preporuke, pohvale (iz škole, sa prakse, od zadovoljnih korisnika aplikacije koju ste izradili), prikažite kratke citate na sajtu <sup>35</sup>. To gradi poverenje. Takođe, istaknite broj projekata, broj godina iskustva (ako značajno) ili druge metrike, u nekoj vizuelnoj formi (brojači, ikone) – npr. "3+ godine iskustva, 5 završenih projekata, 2 sertifikata" i slično.
- **Korisnički put (User journey):** Zamislite da posetilac dođe na vaš sajt – pokušajte da mu omogućite da u roku od 1–2 minuta pokupi sve bitne informacije: ko ste, koje veštine imate, kako da vas kontaktira, i možda da pogleda 1–2 projekta koja ga zanimaju. Struktura koju smo definisali upravo vodi kroz tu priču <sup>36</sup> <sup>37</sup>: kratko predstavljanje (personal brand), zatim prikaz veština i projekata (demonstracija umenja), potom radno iskustvo/edukacija (kredibilitet), i na kraju poziv da se jave (kontakt). Obezbedite i da se sa jednog dela lako može "skočiti" na drugi – npr. na kraju About sekcije može biti link "Pogledaj moje projekte →", ili na Home odmah postoji dugme koje vodi na Projects stranu.
- **Provera u različitim browser-ima:** Testirajte sajt u više pregledača (Chrome, Firefox, Safari, Edge, mobilni browseri). Moderni CSS uglavnom radi svuda, ali ipak proverite. Takođe testirajte na različitim veličinama ekrana manualno (developer tools vam omogućavaju *Device Toolbar* za simulaciju).
- **SEO optimizacija:** Pored meta tagova i čiste strukture, razmislite o sadržaju: uključite svoje ime u naslov početne strane (da Google asocira pretragu vašeg imena na sajt), dodajte eventualno strukturisane podatke ([schema.org](#)) ako ste napredni – npr. JSON-LD skript koji definiše *Person* i *WebSite* – nije neophodno, ali može pomoći. U najmanju ruku, **naziv fajla** CV-a (`cv.pdf`) može sadržati vaše ime, i title tag takođe.
- **GitHub Pages posebnosti:** Ako je repo ime `username.github.io`, sajt će biti dostupan na tom domenu root. Ako je običan repozitorijum, dobijete URL sa dodatkom repozitorijuma. U oba slučaja, možete podesiti **vlastiti domen** kasnije za profesionalniji izgled (nije obavezno, ali dobro za budućnost). Takođe, u README repozitorijuma stavite link ka sajtu, tako da ljudi koji vide kod mogu lako da posete sajt.

Konačno, **priprema za budućnost:** Planirano je naknadno dodavanje JavaScript funkcionalnosti. Ostavite u kodu "kukice" za to – npr. dodajte `id="theme-toggle"` na dugme za promenu teme pa kasnije lako dodate JS kod za klik događaj koji menja klasu na body. Ili spremite okvir za validaciju kontakt forme. Struktura koju ste postavili treba da bude **skalabilna**: da lako možete dodati novi

projekat, novi blog post ili novu sekciju bez refaktorovanja od nule. Korišćenjem dobre prakse i semantike, to će biti izvodljivo.

Uz ovakav pristup, rezultat će biti **profesionalno izgledajući portfolio sajt**, hostovan na GitHub Pages, koji sadrži sve ključne elemente modernog digitalnog CV-a: od pregledne prezentacije veština i projekata do responsivnog dizajna, tamne teme i SEO optimizacije. Pridržavajte se navedenog plana i prilagodite ga sopstvenom stilu – vaš lični pečat u dizajnu (boje, font, eventualno logo ili monogram) učiniće da sajt bude **autentičan**, a gore nabrojane stavke da bude istovremeno **funkcionalan i efektan**. Srećno kodiranje!

## Reference:

- Preporučene sekcije i funkcionalnosti portfolija 38 39 6 40
  - Primeri uspešnih portfolija i saveti (Hostinger) 3 9 41
  - SEO i dostupnost – meta tagovi, alt, aria-label (Dev.to) 16 17 23 20
  - Semantički HTML i struktura navigacije (FCC forum) 15 10
  - CSS promenljive i tamna tema – najbolja praksa 26 28
  - Struktura projekta i fajlova (freeCodeCamp) 11 12
  - Pristupačnost i UX saveti (roadmap.sh) 34 42
- 

1 5 6 7 8 32 33 34 36 37 38 39 40 42 Top 10 Web Developer Portfolio Templates - A Pro's Pick

<https://roadmap.sh/frontend/web-developer-portfolio>

2 10 best developer portfolio website templates and designs | Webflow Inspo  
<https://webflow.com/list/developer-portfolio>

3 4 9 29 35 41 25 Web Developer Portfolio Examples from Top Developers  
<https://www.hostinger.com/tutorials/web-developer-portfolio>

10 What is best practice when creating website with multiple webpages? : r/webdev  
[https://www.reddit.com/r/webdev/comments/zgt7z7/what\\_is\\_best\\_practice\\_when\\_creating\\_website\\_with/](https://www.reddit.com/r/webdev/comments/zgt7z7/what_is_best_practice_when_creating_website_with/)

11 12 25 27 30 31 How to Build a Simple Portfolio Website With HTML and CSS  
<https://www.freecodecamp.org/news/build-a-simple-portfolio-website-with-html-and-css/>

13 14 HTML Semantic Elements  
[https://www.w3schools.com/html/html5\\_semantic\\_elements.asp](https://www.w3schools.com/html/html5_semantic_elements.asp)

15 21 22 28 Personal Portfolio Webpage - Build a Personal Portfolio Webpage - HTML-CSS - The freeCodeCamp Forum  
<https://forum.freecodecamp.org/t/personal-portfolio-webpage-build-a-personal-portfolio-webpage/667774>

16 17 18 19 20 23 24 SEO tips for your developer portfolio - DEV Community  
<https://dev.to/rossellafer/seo-tips-for-your-developer-portfolio-26fm>

26 How I Utilized CSS Variables in My Portfolio Site | by Marcus Siegel | Medium  
<https://msiegel37.medium.com/how-i-utilized-css-variables-on-my-portfolio-site-53f097d88fff>