

# Implementacija smjernica za pristupačnost u web aplikacijama

---

**Milardović, Tomislav**

**Master's thesis / Diplomski rad**

**2024**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **University of Split, Faculty of Science / Sveučilište u Splitu, Prirodoslovno-matematički fakultet**

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:166:233022>

*Rights / Prava:* [In copyright](#)/[Zaštićeno autorskim pravom.](#)

*Download date / Datum preuzimanja:* **2025-01-04**

*Repository / Repozitorij:*

[Repository of Faculty of Science](#)



SVEUČILIŠTE U SPLITU  
PRIRODOSLOVNO MATEMATIČKI FAKULTET

DIPLOMSKI RAD

**IMPLEMENTACIJA SMJERNICA ZA  
PRISTUPAČNOST U WEB APLIKACIJAMA**

Tomislav Milardović

Split, rujan 2024.

# Temeljna dokumentacijska kartica

Diplomski rad

Sveučilište u Splitu  
Prirodoslovno-matematički fakultet  
Odjel za informatiku  
Ruđera Boškovića 33, 21000 Split, Hrvatska

## IMPLEMENTACIJA SMJERNICA ZA PRISTUPAČNOST U WEB APLIKACIJAMA

Tomislav Milardović

### SAŽETAK

Diplomski rad bavi se tematikom web pristupačnosti, s posebnim naglaskom na prilagodbe za korisnike s različitim vrstama poteškoća. U radu se identificiraju poteškoće s kojima se osobe mogu suočiti u digitalnom okruženju. Analizirani su izazovi koje ti korisnici susreću te alati poput čitača zaslona, povećavanja teksta i prepoznavanja glasa koji im olakšavaju pristup sadržaju. Rad daje uvid i u glavne smjernice za izradu pristupačnih web stranica prema WCAG standardima.

**Ključne riječi:** web pristupačnost

Rad je pohranjen u knjižnici Prirodoslovno-matematičkog fakulteta, Sveučilišta u Splitu

**Rad sadrži:** 47 stranica, 22 grafičkih prikaza i 22 literaturnih navoda. Izvornik je na hrvatskom jeziku.

**Mentor:** **dr. sc. Goran Zaharija**, *docent Prirodoslovno-matematičkog fakulteta, Sveučilišta u Splitu*

**Ocjenjivači:** **dr. sc. Goran Zaharija**, *docent Prirodoslovno-matematičkog fakulteta, Sveučilišta u Splitu*

**dr. sc. Divna Krpan**, *docent Prirodoslovno-matematičkog fakulteta, Sveučilišta u Splitu*

**Nika Jerković**, *asistent Prirodoslovno-matematičkog fakulteta, Sveučilišta u Splitu*

Rad prihvaćen: **rujan 2024.**

# Basic documentation card

Thesis

University of Split  
Faculty of Science  
Department of Computer Science  
Ruđera Boškovića 33, 21000 Split, Croatia

## IMPLEMENTATION OF ACCESSIBILITY GUIDELINES IN WEB APPLICATIONS

Tomislav Milardović

### ABSTRACT

The thesis addresses the topic of web accessibility, with a particular focus on adaptations for users with different types of disabilities. It identifies the challenges individuals may encounter in digital environments. The difficulties these users face are analyzed, along with tools such as screen readers, text enlargement, and voice recognition that facilitate their access to content. The thesis also provides an overview of the main guidelines for creating accessible websites according to WCAG standards.

**Key words:** web accessibility

Thesis deposited in library of Faculty of science, University of Split

**Thesis consists of:** 22 pages, 22 figures and 22 references

Original language: Croatian

**Mentor:** **Goran Zaharija, Ph.D.** *Assistant Professor of Faculty of Science, University of Split*

**Reviewers:** **Goran Zaharija, Ph.D.** *Assistant Professor of Faculty of Science, University of Split*

**Divna Krpan, Ph.D.** *Assistant Professor of Faculty of Science, University of Split*

**Nika Jerković,** *Assistant of Faculty of Science, University of Split*

Thesis accepted: **September 2024.**

# IZJAVA

kojom izjavljujem s punom materijalnom i moralnom odgovornošću da sam završni rad s naslovom **IMPLEMENTACIJA SMJERNICA ZA PRISTUPAČNOST U WEB APLIKACIJAMA** izradio samostalno pod voditeljstvom doc. dr. sc. Goran Zaharija. U radu sam primijenio metodologiju znanstvenoistraživačkog rada i koristio literaturu koja je navedena na kraju završnog rada. Tuđe spoznaje, stavove, zaključke, teorije i zakonitosti koje sam izravno ili parafrazirajući naveo u završnom radu na uobičajen, standardan način citirao sam i povezao s fusnotama s korištenim bibliografskim jedinicama. Rad je pisan u duhu hrvatskog jezika.

Student

Tomislav Milardović



## Sadržaj

Uvod .....	1
1. Pristupačnost.....	2
1.1. Digitalna pristupačnost.....	2
1.2. Pristupačnost i društvo .....	3
1.3. Povijest .....	4
1.4. Pristupačnost i zakoni.....	5
2. Smjernice za implementaciju.....	7
2.1. WCAG.....	7
2.2. WAI-ARIA.....	8
2.3. Prednosti korištenja pristupačnosti .....	10
3. Primjena pristupačnosti po ciljnoj skupini .....	13
3.1. Osobe s vidnim smetnjama.....	13
3.2. Osobe s oštećenjem sluha .....	15
3.3. Osobe s motoričkim poremećajima .....	15
3.4. Osobe s otežanim kognitivnim sposobnostima .....	16
3.5. Osobe s vestibularnim poremećajima i napadajima .....	17
4. Osiguravanje pristupačnosti .....	18
4.1. Alati za testiranje pristupačnosti.....	18
4.2. Struktura stranice.....	20
4.3. Pristupačnost i WAI-ARIA oznake .....	23
4.4. „Aria-label“ i „aria-labelledby“ .....	26
4.5. Prilagodba slabovidnosti .....	27
4.6. Prilagodba disleksiji .....	31
4.7. Prilagodba veličine fonta .....	32
4.8. Navigacija tipkovnicom.....	33

4.9.	Opisi i transkripti za multimediju.....	36
4.10.	Čitač zaslona.....	38
4.11.	Animacije i interaktivni elementi .....	40
4.12.	Pohrana korisnikovih preferencija.....	41
	Zaključak .....	43
	Literatura .....	44
	Tablica slika.....	46
	Privitak .....	47



# Uvod

U današnjem digitalnom dobu, Internet je postao glavni izvor informacija, usluga i interakcija koji je preoblikovao način života, učenja i povezivanja. Sveprisutnost interneta omogućila je brojne prilike za komunikaciju, trgovinu i obrazovanje, rušeći geografske barijere i omogućujući globalnu razmjenu ideja. Međutim, kako se ovo digitalno područje širi, osiguravanje jednakog pristupa za sve pojedince, bez obzira na njihove fizičke ili kognitivne sposobnosti, postaje imperativ koji se mora riješiti. Ovdje se koncept web pristupačnosti pojavljuje kao vodeći princip, nastojeći učiniti online okruženje pristupačnim za sve.

Digitalna pristupačnost podrazumijeva niz prilagodbi mrežnih stranica, aplikacija i ostalih digitalnih sadržaja kojima je cilj eliminirati prepreke koje bi mogle spriječiti osobe s invaliditetom u korištenju istih. Korištenje sadržaja bez potrebnih prilagodbi može biti prilično izazovno, od poteškoća prilikom samog čitanja sadržaja pa sve do složenijih akcija upravljanja sučeljem.

U radu je opisan pojam digitalne pristupačnosti, važnost, načela, prednosti i izazovi. Razumijevanjem važnosti dostupnosti weba i istraživanjem praksi koje ga podupiru ostvaruje se preduvjet za stvaranje digitalnog društva koji osigurava jednak pristup i sudjelovanje za sve. Kako internet nastavlja oblikovati način na koji komuniciramo, davanje prioriteta dostupnosti weba nije samo nužnost, već i dokaz naše predanosti izgradnji svijeta koji cijeni različitost i prihvaća svakog pojedinca, bez obzira na njegove sposobnosti. Također, dati će se uvid u dio rada koji obuhvaća primjenu prilagodbe web aplikacije.

# 1. Pristupačnost

Pristupačnost, u kontekstu tehnologije i dizajna, odnosi se na praksu stvaranja proizvoda, usluga i okruženja upotrebljivim osobama s invaliditetom. To uključuje osiguravanje da pojedinci s različitim vrstama invaliditeta, kao što su vizualna, slušna, motorička ili kognitivna oštećenja, mogu komunicirati s digitalnim sadržajem, web stranicama, softverskim aplikacijama i drugim tehnološkim proizvodima i imati koristi od njih.

Pristupačnost je ključna za osobe s poteškoćama, ali i općenito korisna za sve ljude u različitim situacijama. Iako je prilagođavanje proizvoda usmjereno za određenu skupinu ljudi, mnogi imaju korist od takvih praksi. Primjer kako je takav proizvod promijenio svijet može biti izum dizala koji su počeli koristiti svi, a ne samo osobe s invaliditetom. [1]

Pristupačnost proizvoda je temeljni princip inkluzivnog dizajna koji stavlja naglasak na stvaranje proizvoda koji su dostupni i korisni za sve. Ovaj se princip koristi u različitim sektorima, uključujući tehnologiju, građevinarstvo, obrazovanje i zabavu. To je pristup dizajniranju proizvoda ili usluga koji su pristupačni i korisni što većem broju ljudi bez potrebe posebnih adaptacija ili posebnog dizajna. Osim što osobama s poteškoćama olakšava korištenje, proizvodi koji su izrađeni inkluzivnim pristupom mogu donijeti i poslovne prilike jer ciljaju na mnogo širu publiku. [2]

Pristupačnost proizvoda može potaknuti i inovacije jer izaziva dizajnere i inženjere na kreativno razmišljanje kako bi riješili različite izazove s kojima se suočavaju različiti korisnici. Na taj se način otvara mogućnost za razvoj novih tehnologija, rješenja i pristupa.

## 1.1. Digitalna pristupačnost

Pristupačnost se kod web stranica ostvaruje dizajniranjem i pisanjem koda koje postaju dostupne većem broju ljudi. Podrazumijeva dizajniranje i razvoj web stranica na način koji se prilagođava korisnicima s invaliditetom. To može uključivati pružanje alternativnog teksta za slike, korištenje navigacije pomoću tipkovnice, osiguravanje odgovarajućeg kontrasta boja za čitljivost, pružanje titlova za videozapise i još mnogo toga.

Osim etičke važnosti, digitalna pristupačnost je često zakonski regulirana u mnogim zemljama kako bi se pristup informacijama omogućio svim građanima.

Digitalna pristupačnost polazi od četiri osnovna načela koja zahtijevaju da svi korisnici, bez obzira na svoje mogućnosti, mogu:

- percipirati informacije i sastavne dijelove korisničkog sučelja (mogućnost opažanja)
- upravljati korisničkim sučeljem (operativnost)
- razumjeti informacije i način rada korisničkog sučelja (razumljivost)
- pristupiti sadržaju bez obzira na tehnologiju (stabilnost). [3]

Kao što je spomenuto, benefite prilagodbi web stranica koriste i ljudi bez poteškoća, tj. ljudi koji se nađu u specifičnim okolnostima u kojima ne mogu koristiti digitalni sadržaj u punom opsegu. Primjerice, za ljude koji imaju određena ograničenja vezana za specifičnu situaciju (nemogućnost čitanja sa zaslona zbog jake sunčeve svjetlosti, formalno okruženje u kojem je neprikladno slušati glasovni zapis). Moguće su i privremene poteškoće (izgubljene naočale za vid) ili za ljude sa sporom ili ograničenom internetskom vezom. Web stranice bi trebale biti prilagođene i prikazu za različite vrste uređaja kao što su mobiteli, pametni satovi, televizije i drugi uređaji manjeg ili većeg zaslona te različitih uređaja koji na različite načine primaju naredbe korisnika. [4]

Pristupačan dizajn potreban je i starijim ljudima. Uz razvitak tehnologije, produljio se životni vijek ljudi, a stariji ljudi su sve više i više tehnički pismeniji. No, starenjem se pojavljuju razne poteškoće. Najčešći su oni koji se tiču vida, sluha i kognitivnih sposobnosti. [2]

## **1.2. Pristupačnost i društvo**

Web je sve važniji resurs u mnogim aspektima života: obrazovanju, zapošljavanju, trgovini, zdravstvu, rekreaciji. Pristup informacijskim i komunikacijskim tehnologijama, uključujući Web, definiran je kao osnovno ljudsko pravo u Konvenciji Ujedinjenih naroda o pravima osoba s invaliditetom. Prepreke dostupnosti tiskanih, audio i vizualnih medija mogu se puno lakše prevladati putem web tehnologija, a podržava se društvena uključenost starijih ljudi, ljudi u ruralnim područjima i ljudi u zemljama u razvoju. [4]

Postoji i snažan poslovni argument za pristupačnost. Pristupačan dizajn poboljšava cjelokupno korisničko iskustvo i zadovoljstvo, posebno u različitim situacijama i uređajima. Pristupačnost može unaprijediti marku, potaknuti inovacije i proširiti doseg na tržištu. [4]

## 1.3. Povijest

Ne postoji točan vremenska period početka razvoja digitalne pristupačnosti, no mogu se izdvojiti određeni događaji koji su usmjerili taj razvoj.

Sredinom 1990-ih počela je jačati svijest o dostupnosti weba, što je prvi spomenuo Tim Berners-Lee u svom govoru na World Wide Web konferenciji 1994. u Chicagu. Godine 1995. dr. Cynthia Waddell objavila je standard pristupačnosti web dizajna za Ured za osiguranje jednakosti grada San Josea. Uključivao je popis specifikacija kojih se dizajneri gradske web stranice trebaju strogo pridržavati. Popis je uključivao zahtjev da sve oznake slika budu popraćene alternativnim tekstom, da svi video i audio elementi budu upareni s tekstualnim transkripcijama i eksplicitno ograničenje od samo dva stupca po HTML tablici, kako bi se ograničio neprikladan prikaz tablica u lošijim preglednicima. Standard je bio godinama ispred svog vremena, prethodeći službenim Smjernicama za pristupačnost web sadržaja (WCAG) W3C-a gotovo pola desetljeća. Poznata je i po softveru za testiranje pristupačnosti – Cynthia Says. [5][6]

1997. W3C pokreće Inicijativu za pristupačnost weba – WAI čiji je cilj poboljšati pristupačnost World Wide Weba za osobe s invaliditetom.

1999. godine W3C-ova Inicijativa za pristupačnost weba (WAI) objavila je prvu verziju Smjernica za pristupačnost web sadržaja (WCAG 1.0). Ovo je bio značajan napredak u promicanju pristupačnog weba kojim je bilo moguće procijeniti pristupačnost na temelju standarda. Standard su koristili programeri pomoćne tehnologije (eng. *assistive technology*) kako bi bolje razumjeli kako ona treba komunicirati sa sadržajem na webu. [5]

2001. razvija se Wave, još jedan alat za testiranje pristupačnosti koji je i danas popularan među programerima. Trenutno ga održava WebAIM.

Mnoge slijepe osobe koriste čitač zaslona, alat koji pretvara informacije na zaslonu u govor. VoiceOver je čitač zaslona nastao 2005., a koriste ga Appleovi proizvodi. Iako nije bio prvi čitač zaslona, vrijedi ga spomenuti jer je on i danas jedan od najpopularnijih čitača zaslona za mobilne uređaje. Dvije godine kasnije razvija se NVDA, još jedan čitač zaslona koji se najčešće koristi kod stolnih i prijenosnih računala.

2008. objavljen je WCAG 2.0 kojim su ispravljeni nedostaci prethodnika. Jedna od promjena je i ta da se prepreke u razvoju pristupačnosti ne vežu isključivo za HTML nego općenito za sve tehnologije. WCAG 2.0 odnosi se na pristupačnost u čitavom nizu web tehnologija,

uključujući Flash, Java i JavaScript. Značajna promjena odnosila se i na Javascript i njegovo prihvaćanje kao legitimne web tehnologije. Uz WCAG 1.0, web stranica je morala pružati sve funkcionalnosti čak i ako je JavaScript isključen u korisnikovom pregledniku, što je općenito utjecalo na sporo usvajanje tog standarda. [5]

Danas postoje brojni načini ostvarivanja pristupačnosti, a jedan od njih je i WAI-ARIA. WAI-ARIA je ekstenzija HTML5 koja omogućuje programerima dodavanje informacija o ulogama, stanjima i svojstvima određenim značajkama koja se razvijaju koristeći JavaScript, a koja ranije nisu bila dostupna korisnicima pomoćnih tehnologija. [5]

Paralelno s objavom WCAG 2.1 2018. godine, koji donosi smjernice za razvoj mobilnog weba i responzivnog dizajna za web sadržaj, pokrenut je i projekt „Silver“. Fokus projekta je na integraciji standarda pristupačnosti u Internet stvari (IoT), što uključuje sve uređaje koji imaju mogućnost spajanja na Internet, od hladnjaka, preko sustava kontrole kućne klime do sigurnosnog nadzora. [5]

## **1.4. Pristupačnost i zakoni**

Digitalna tehnologija je u mnogočemu poboljšala svakodnevicu i učinila život jednostavnijim. Online kupnju je moguće izvršiti iz udobnosti svoga doma, čekanje u dugim redovima za usluge banaka je uvelike smanjeno zahvaljujući e-bankarstvu, a okuse raznih restorana je moguće doživjeti narudžbama uz aplikacije za dostavu hrane.

Pristup brojnim sličnim tehnologijama nije uvijek inkluzivan za sve osobe. Kako bi se spriječila diskriminacija i težilo ravnopravnijem pristupu, zemlje diljem svijeta objavljuju zakone o dostupnosti weba, iako su zakoni ti koji se bore da uhvate korak s tehnološkim razvojem.

U različitim literaturama se često spominju zakoni SAD-a, Australije, Japana i država Europske Unije, što i ne čudi s obzirom na tehnološki razvoj navedenih. Često su i same kompanije one koje postavljaju standarde u razvoju i primjeni pristupačnosti. Microsoft i Apple često objavljuju proizvode koji ne samo da slijede zakonske okvire već i na neki način daju smjernice za daljnji razvoj pristupačnosti.

Europska unija je direktivom o pristupačnosti internetskih stranica i mobilnih aplikacija tijela javnog sektora obvezala sve članice na donošenje nacionalnih propisa, pa je Hrvatska 2019. donijela „Zakon o pristupačnosti mrežnih stranica i programskih rješenja za pokretne

uređaje tijela javnog sektora“. Zakon utvrđuje da se mjere trebaju poduzeti kako bi se mrežne stranice javnog sektora učinile pristupačnijima korisnicima. Tijela javnog sektora su dužna osigurati opažanje, operabilnost, razumljivost i stabilnost sadržaja. Zakon navodi iznimke od primjene za određene mrežne stranice i sadržaje. Također, zakon propisuje načine usklađenosti sa zahtjevima pristupačnosti i postupke praćenja te izvješćivanja o provedbi zakona. [7]

## 2. Smjernice za implementaciju

### 2.1. WCAG

Smjernice za pristupačnost web sadržaja (WCAG 2) razvijene su od strane W3C u suradnji s pojedincima i organizacijama diljem svijeta, s ciljem pružanja jedinstvenog zajedničkog standarda za pristupačnost web sadržaja koji zadovoljava potrebe pojedinaca, organizacija i vlada na međunarodnoj razini. Dokumenti objašnjavaju kako web sadržaj učiniti pristupačnijim osobama s invaliditetom. "Sadržaj" se općenito odnosi na informacije na web stranici ili web aplikaciji, uključujući tekst, slike i zvukovi, kod ili oznake koje definiraju strukturu, prezentaciju itd. Dokumenti su prvenstveno namijenjeni web programerima, programerima alata za izradu web stranica i alata za procjenu pristupačnosti. [8]

Smjernice se temelje na četiri glavna principa koji opisuju zahtjeve koje web-sadržaj mora imati kako bi se smatrao pristupačnim:

- percepcija – korisnici moraju biti u mogućnosti percipirati sadržaj koristeći jedno ili više osjetila
- upotrebljivost – korisnici moraju imati mogućnost upravljanja elementima korisničkog sučelja (npr. pritisak na tipku mora biti moguć uz pomoć miša, tipkovnice, glasovnim upravljanjem i sl.)
- razumljivost – sadržaj mora biti razumljiv korisnicima
- robusnost – sadržaj mora biti razvijen uz pomoć dobro usvojenih web standarda koji će raditi na različitim preglednicima. [9]

WCAG se sastoji od 13 smjernica koje se temelje na četiri navedena principa. Za svaku smjernicu postoje provjerljivi kriteriji uspješnosti koji se dijele na tri razine: A, AA i AAA. Kriteriji uspješnosti određuju usklađenost sa WCAG smjernicama.

Razina A je minimalna razina usklađenosti. Sadrži osnovne kriterije uspjeha za uklanjanje ozbiljnih prepreka pristupačnosti koje utječu na širok raspon korisnika. Odnosi se na zahtjeve kao što su tekstualne alternative, opisi, upotreba boja, redoslijed sadržaja, upravljanje tipkovnicom i upravljanje pokretnim elementima na sučelju kao što su pauziranje, zaustavljanje ili sakrivanje.

Razina AA uklanja dodatne prepreke i uspostavlja razinu pristupačnosti koja funkcionira za većinu uređaja i pomoćnih tehnologija. Uključuje zahtjeve koji se odnose na zvučne opise, kontraste boja, veličinu teksta, naslove i oznake, navigaciju, povratne informacije o pogreškama u stvarnom vremenu.

Razina AAA je najviša razina usklađenosti. Sadrži dodatne kriterije uspjeha za uspostavljanje najviše moguće razine pristupačnosti. Podrazumijeva poboljšane kontraste boja, znakovni jezik za unaprijed snimljeni sadržaj, jasno značenje poveznica, dodatne informacije za kratice, korisnička kontrola nad vizualnim prikazom teksta i nad prekidima kao što su skočni prozori ili automatski reproducirani oglasi.

## 2.2. WAI-ARIA

WAI-ARIA (Accessible Rich Internet Applications) definira način na koji je moguće ostvariti pristupačniji web sadržaj i web aplikacije osobama s invaliditetom. Pomaže kod dinamičkog sadržaja i naprednih kontrola korisničkog sučelja razvijenih s HTML-om, JavaScriptom i sličnim tehnologijama. Bez upotrebe WAI-ARIA određene funkcionalnosti koje se koriste na web stranicama nisu dostupne nekim korisnicima s invaliditetom, posebno osobama koje se oslanjaju na čitače zaslona i osobama koje ne mogu koristiti miš. WAI-ARIA definira način na koji se određene funkcionalnosti mogu pružiti pomoćnoj tehnologiji. [10]

WAI-ARIA pruža okvir za dodavanje atributa za identifikaciju značajki za korisničku interakciju, njihove međusobne odnose i njihovo trenutno stanje. WAI-ARIA opisuje tehnike navigacije za označavanje regija i razne web strukture kao što su izbornici, primarni sadržaj, sekundarni sadržaj, podnožje stranice i slično. Omogućava autorima opisivanje različitih elemenata sučelja kao što su izbornik (menu), element stabla (treeitem), klizač (slider), traka napretka (progressbar) te svojstva kojima se opisuje stanje pojedinih elemenata, primjerice stanje „označeno“ (checked) kod korištenja potvrdnog okvira. [10]

ARIA pruža nove attribute koda i vrijednosti koje proširuju vokabular i semantiku HTML-a. Kao i HTML, ARIA se razvija tijekom vremena i općenito je dobro podržan u modernim preglednicima i čitačima zaslona. Razlikuju se uloge (roles), stanja (states) i svojstva (properties). Ulogama se definira vrsta ili svrha elemenata na web stranici. Stanje pruža dodatne informacije o trenutnom stanju ili svojstvu elementa, a služe za prijenos dinamičkih promjena ili korisničkih interakcija pomoćnim tehnologijama. Svojstva za razliku od stanja



daju statičke informacije o elementu, tj. karakteristike koje se ne mijenjaju tijekom interakcije s korisnikom.

ARIA specifikacija definira pet pravila korištenja ARIA-e za podršku optimalne pristupačnosti:

- Ukoliko je određena funkcionalnost dostupna koristeći HTML, potrebno je izbjegavati korištenje ARIA atributa. HTML često sam pruža dovoljnu semantiku za razvoj pristupačnih aplikacija, pa upotreba ARIA atributa i njegovog pogrešnog korištenja može dovesti do određenih prepreka u korištenju sadržaja.
- Izvorna semantika HTML se ne smije mijenjati, a ARIA može mijenjati semantiku. Ovo se odnosi na određene oznake HTML. Primjerice ukoliko želimo da naslovu s oznakom `<h1>` dodijelimo ulogu (role) „tab“ koji može poslužiti za otkrivanje nekog dijela sučelja, potrebno je naslov ugraditi u oznaku `<div>`, koji se najčešće koristi za označavanje određenih sekcija, te toj oznaci dodijeliti ulogu „tab“:

```
<div role=tab>
```

```
    <h2>Naslov</h2>
```

```
</div>
```

- Sve interaktivne kontrole razvijene korištenjem ARIA atributa moraju se moći koristiti s tipkovnicom. Takvi elementi sučelja moraju reagirati na standardne pritiske tipki ili njihove kombinacije. Ovo također omogućava usklađenost uputa čitača zaslona sa stvarnom funkcijom na stranici.
- Svi elementi na koje se moguće pozicionirati koristeći tipkovnicu, moraju imati odgovarajuću semantiku kako bi se identificirali kao link, tipka ili razni elementi obrazaca. Potrebno je kod takvih elemenata izbjegavati uporabu `role="presentation"` jer uklanja semantiku HTML elementa na kojeg se primjenjuje. Sličan problem je i sa `aria-hidden="true"`. Postavljanjem ovoga na HTML oznaku uklanja se mogućnost pozicioniranja na taj element.
- Interaktivne kontrole je potrebno opisati kako bi korisnici čitača zaslona znali što se očekuje ili kakva akcija se događa pozicioniranjem na određenu kontrolu. Primjerice, tipka mora imati opisni tekst „Registracija“ ako se njime stvara određeni korisnički račun na nekoj aplikaciji, ili kontrola koja pruža unos teksta koja treba imati pripadajuću oznaku, ovisno o tome što se očekuje kao unos, npr. „Ime“.

Dodavanje ARIA uloge nadjačava izvornu semantiku HTML oznaka, pa stoga ARIA utječe na ono što se prijavljuje čitaču zaslona ili drugoj pomoćnoj tehnologiji. S druge strane, dodavanje ARIA uloge neće učiniti da element izgleda ili djeluje drugačije za osobe koje ne koriste pomoćnu tehnologiju. Ne mijenja se ponašanje, stanje niti svojstva glavnog elementa, već samo izvorna semantika. [11]

## **2.3. Prednosti korištenja pristupačnosti**

Najočitija prednost web pristupačnosti je da pomaže osobama s invaliditetom da koriste web sadržaj, proizvode i usluge. Međutim, prednosti web pristupačnosti nisu ograničene samo na njihov učinak na osobe s invaliditetom. Postoje brojne druge prednosti web pristupačnosti kako za korisnike tako i za organizaciju ili autora koji stoji iza web sadržaja.

Pružanje tekstualnih alternativa vizualnom sadržaju koristi svima koji nemaju trenutni pristup grafičkim elementima stranice, što osim slijepih, uključuje i korisnike računala koji pretražuju internetom pomoću tekstualnih preglednika koji pružaju brz i učinkovit pristup web sadržaju. Također, korisnici se mogu nalaziti u područjima koja su slabije pokrivena signalom. Korisnicima s lošijom internet vezom se grafika možda uopće neće prikazati što ukazuje na potrebu korištenja alternativnih opisa. Slično tome, korisnici mobilnih uređaja često onemogućuju grafiku u svojim preglednicima kako bi omogućili brža preuzimanja i sačuvali memoriju. [12]

Kontrast je jedna od najčešćih prilagodbi ljudima koji imaju poteškoća s vidom. Mogućnost promjene kontrasta pomaže i starijim osobama čija osjetljivost na boje prirodno opada s godinama ili ljudima u različitim uvjetima osvjetljenja koji primjerice doživljavaju odsjaj na zaslonu pri jakoj sunčevoj svjetlosti. Vrijedi spomenuti i kako je posljednjih godina sve češća upotreba tamnih pozadina web stranica i aplikacija koje su se isprva činile kao prilagodba uvjetima slabijeg osvjetljenja, ali dodatne prednosti su i smanjen negativan utjecaj zaslonskog svjetla na oči te manja potrošnja energije zbog čega baterije imaju duže trajanje.

Pružanje tekstualnih alternativa audio sadržaju omogućuje pristup osobama s ograničenim ili nikakvim pristupom zvučnom izlazu. Ovo uključuje osobe koje su gluhe ili nagluhe, ali dodatno uključuje svakoga tko pokušava pristupiti sadržaju u bučnom okruženju te osobe kojima nedostaje, kojima je pokvaren ili na bilo koji drugi način neadekvatan zvučni hardver.

Multimedija s titlovima također omogućuje indeksiranje i arhiviranje sadržaja tako da ga je moguće u potpunosti pretraživati. [12]

Tehnologija ostvarivanja interakcije s računalima ispočetka je bila pomoć slijepima ili ljudima s poteškoćama u kretanju. No, s vremenom se ovaj način interakcije proširio i među onima koji nemaju takve izazove. Mogućnost unosa teksta ili pretraživanja web stranica govorom je sveprisutna, od različitih preglednika i web stranica koji imaju opciju pretraživanja glasom, raznih aplikacija za razmjenu poruka do virtualnih asistenata. Microsoft je u čitavom nizu aplikacija Office 365 paketa omogućio diktiranje sadržaja i glasovno upravljanje naredbama.

Ranije su spomenuti zakoni koji obvezuju različite organizacije na primjenu pristupačnosti. S obzirom na sve veću prisutnost interneta i općenito digitalnih tehnologija, dostupnost web sadržaja postaje sve više građansko pravo te suočavanje s pravnim pritužbama organizacijama kod kojih to pravo nije ostvarivo.

Invaliditeti se javljaju u mnogim različitim oblicima. Američki centri za kontrolu i prevenciju bolesti (CDC) procjenjuju da 1 od 4 odrasle osobe u SAD-u živi s invaliditetom. Poduzeća i organizacije stoga ne bi smjele ignorirati ovaj dio populacije. Prema istraživanju „Click-Away Pound“, trgovci u Velikoj Britaniji izgubili su procijenjenih 11,75 milijardi funti u 2016. jer osobe s invaliditetom nisu mogle kupiti njihove proizvode online. Isto istraživanje je provedeno i 2019. godine, a rezultati su pokazali porast propuštenih „poslovnih prilika“ na 17,1 milijardu funti. Poboljšanje pristupačnosti web stranice se očito čini kao sjajna poslovna odluka. [13]

Pristupačnost weba je važna za osobe s invaliditetom, njihove bližnje i zagovornike prava osoba s invaliditetom. Općenito, mnogi potrošači žele podržati tvrtke koje dijele njihova uvjerenja, ideale i vrijednosti. Osobe s invaliditetom koje imaju pozitivnu interakciju s pristupačnim sadržajem vjerojatnije će isto preporučiti svojoj obitelji, prijateljima i poznanicima. Iskustvo korištenja, pozitivno ili ne, utječe na to hoće li se osoba uopće više vratiti na stranicu.

Stvaranjem pristupačnije stranice povećava se vidljivost web stranice. Pristupačnost doprinosi SEO optimizaciji i boljoj poziciji na različitim tražilicama, što u konačnici donosi veći broj posjeta.

U konačnici, korist mogu imati i programeri. Pravilno implementirana pristupačnost može smanjiti tehničke probleme i greške u kodiranju. To olakšava održavanje i ažuriranje web stranica i aplikacija.

## 3. Primjena pristupačnosti po ciljnoj skupini

Različite vrste invaliditeta zahtijevaju različite pomoćne tehnologije kako bi omogućile pristup tehnološkim resursima. Pomoćna tehnologija obuhvaća razne hardverske i softverske alate koji su dizajnirani da olakšaju život osobama s invaliditetom. Svaka vrsta invaliditeta zahtijeva svoja specifična tehnološka rješenja. U nastavku su navedene različite vrste invaliditeta i odgovarajuće pomoćne tehnologije, uz opis pristupačnog dizajna web stranica kako bi se osiguralo da svi korisnici mogu bez prepreka koristiti web sadržaj.

### 3.1. Osobe s vidnim smetnjama

Pomoćna tehnologija osnažuje osobe s invaliditetom da obavljaju zadatke koji bi inače bili teški ili nemogući. Malo ljudi koji nose naočale smatraju se osobama s invaliditetom, no mnogi bi bez njih imali poteškoća s korištenjem interneta.

Slijepi osobe nisu u mogućnosti koristiti web na uobičajen način, a monitor i miš su ustvari beskorisni. Vizualne informacije kao što su različite grafike, raspored elemenata na sučelju i znakovi ne mogu se uočiti, a za upravljanje i navigaciju web sadržajem najčešće se koristi tipkovnica.

Slijepi osobe mogu slušati, kretati se i komunicirati s web sadržajem putem softvera za čitanje zaslona koji web sadržaj pretvara u govor. Korisnik može dopustiti čitaču zaslona da čita sve od vrha do dna, red po red, ili korisnik može koristiti tipku „tab“ za navigaciju kroz veze i forme. Korisnik se također može kretati putem naslova (ako web-sadržaj ima naslove), s jedne regije stranice na drugu (ako postoje definirane regije) ili raznim drugim metodama. Pametni telefoni i tableti dolaze s ugrađenim čitačima zaslona koji omogućuju pokrete dodira i prelaska prstom za navigaciju kroz sadržaj. Najčešće korišteni čitači zaslona su JAWS, NVDA i VoiceOver. [14]

Čitače zaslona koriste i oni koji su i gluhi i slijepi koristeći uređaj koji pretvara sadržaj web stranice u Brailleove znakove. Takvi uređaji imaju male mehaničke igle koje se mogu podići ili spustiti kako bi oblikovali Brailleove znakove koje osoba može osjetiti.

Korisnici koji nemaju problema s vidom mogu brzo pregledati stranicu, razumjeti njenu strukturu i stvarati mentalne asocijacije ili prepoznati sadržaj na temelju rasporeda ili vizualnog dizajna. Kako bi prenijeli barem osnovne informacije ili svrhu slika osobama s

poteškoćama, programeri dodaju alternativni tekst na njih. Osim teksta, bitno je pravilno strukturirati stranice i koristiti semantički ispravan kod kako bi bilo moguće preskakati i istraživati sadržaj uz pomoć čitača zaslona. [14]

S obzorom da se korisnici čitača zaslona obično kreću webom pomoću tipkovnice, sve funkcije moraju biti dostupne putem tipkovnice. Potrebno je izbjegavati dodavanje bilo kakvih elemenata koji podrazumijevaju korištenje miša ili zaslona osjetljivog na dodir. Neki korisnici koji nemaju problema s vidom također se oslanjaju na tipkovnicu zbog motoričkih poteškoća ili drugih razloga.

Još jedna od poteškoća je slabovidnost. Češća je među starijim osobama, ali može se pojaviti u bilo kojoj dobi. Nastaje kao posljedica određenih bolesti koje su vezane za oči. Svako od tih stanja različito utječe na vid osobe pa se tako razlikuju i poteškoće koje korisnici mogu imati sa web sadržajem, od onih koje su vezane za veličinu teksta pa do kontrasta boja. Osobe imaju problema središnjim vidnim područjem koje je zatamnjeno ili zamućeno, koje pokušavaju nadomjestiti perifernim vidom (makularna degeneracija), gubitak perifernog vida (glaukom), pojava tamnih „mrlja“ u vidnom polju (dijabetička retinopatija) ili zamagljen učinak (siva mrena). [14]

Za prilagodbu sadržaja ovoj skupini ljudi najčešće se koristi povećavanje cijelog ili dijela zaslona te korištenje visokog kontrasta koji omogućuju korisnicima da jasnije vide sadržaj. Povećavanje omogućuje korisnicima da podese svoje preglednike da automatski povećaju sadržaj ili veličinu teksta. Povećavanjem zaslona ne smije doći do gubitka sadržaja i funkcionalnosti, a implementacijom responzivnog dizajna dodatno se osigurava prilagodba sadržaja veličini zaslona. Omogućavanje promjena kontrasta i boja je još jedna poželjna prilagodba, iako je nju moguće podesiti kod velikog broja preglednika.

Daltonizam je nemogućnost razlikovanja određenih nijansi boja što čini određene kombinacije boja teškim za razlikovanje. Boje s kojima osobe imaju poteškoća u razlikovanju ovise o njihovoj vrsti daltonizma, ali nedostaci crveno-zelene su najčešći, što znači da se određene nijanse crvene i zelene boje mogu činiti vrlo sličnim. Prilikom dizajniranja web sadržaja, najbitnije za ovu skupinu ljudi je da označavanje bojama nije jedini način prikazivanja bitnih informacija. Takve informacije potrebno je nadopuniti tekstom ukoliko se radi o nekom grafičkom prikazu, ili primjenom različitog kontrasta boja.

## 3.2. Osobe s oštećenjem sluha

Postoje različiti stupnjevi gubitka sluha, od blagog gubitka kod kojeg pojedinci imaju problema u razumijevanju normalnog govora, pa sve do teškog gubitka sluha kod kojeg pojedinci neće primijetiti niti glasan govor i buku.

Kod osoba s oštećenjem sluha, audio sadržaj je potrebno predstaviti na drugačiji način. U tu svrhu se najčešće koristi tekstualni opis. Ukoliko se radi o multimedijском zapisu, sadržaj je potrebno popratiti titlovima, ili ako se radi samo o audio zapisu potrebno je osigurati prijepis.

Osobe koje su i gluhe i slijepe često komuniciraju znakovnim jezikom stavljanjem ruke druge osobe u svoju i osjećaju znak. Kada pristupaju web sadržaju, uglavnom koriste uređaje koji koriste Brailleove znakove koji im omogućuju pristup tekstualnom sadržaju web stranice, uključujući alternativni tekst za slike.

Audio sadržaj popraćen znakovnim jezikom je još jedan način razvoja pristupačnosti za gluhe osobe, ali se rijetko koristi. Razlog je što postoji preko 130 različitih znakovnih jezika diljem svijeta. Ukoliko web sadržaj nije namijenjen određenoj demografskoj skupini, prikaz web sadržaja znakovnim jezikom nije najbolja opcija. [14]

## 3.3. Osobe s motoričkim poremećajima

Ključan princip kojeg se potrebno pridržavati za ljude s motoričkim poremećajima je operabilnost s obzirom da nije svatko u mogućnosti koristiti miš, koristiti ga za navigaciju kroz web sadržaj ili upravljati dinamičkim elementima.

Postoje različite vrste poremećaja vezanih uz motoričke sposobnosti koje više ili manje utječu na način kojim se ova skupina ljudi koristi webom.

Ozlijede leđne moždine mogu paralizirati udove. Osobe koje imaju paralizu nogu općenito nemaju poteškoća s pristupom web sadržaju, dok oni s paralizom ruku mogu imati značajnih poteškoća, ovisno o vrsti i težini ozlijede koje mogu otežati ili onemogućiti korištenje miša ili tipkovnice. Najčešći način za postizanje pristupačnosti je softver za prepoznavanje i upravljanje glasom.

Ozljede mozga često uzrokuju smanjenu kontrolu mišića što uključuje stezanje ili grčenje mišića, nekontrolirane pokrete ili poremećaje u govoru. Kod korištenja računala, ovakve poteškoće ponajviše ometaju spretnost u korištenju miša ili tipkovnice.

Ključ kod razvoja pristupačnog web sadržaja za programere je ne oslanjati se na određenu vrstu ulaznog uređaja, s obzirom da većina pomoćnih tehnologija oponaša funkcionalnost tipkovnice, miša ili neke dodirne podloge. Tako postoje različite tehnologije, od uređaja u obliku štapića kojim se upravlja ustima za osobe koje nisu u mogućnosti koristiti ruke, uređaj s trakama za glavu pomoću kojih se upravlja tipkovnicom, te različite vrste prilagodljivih tipkovnica. Postoje i uređaji za praćenje očiju koji mogu biti alternativa za osobe s ograničenom ili nikakvom kontrolom nad pokretima ruke. Uređaj omogućuje osobi da se kreće kroz web samo pokretima očiju. Uz odgovarajući softver, osoba je u mogućnosti tipkati, a može uključivati i tehnologiju dovršavanja riječi kako bi se ubrzao proces. [14]

### **3.4. Osobe s otežanim kognitivnim sposobnostima**

Kognitivni invaliditet opisuje osobu koja ima veće poteškoće s mentalnim zadacima od prosječne osobe. Kognitivni poremećaji daleko su najčešći tip invaliditeta, a većina ih je vezana uz biologiju ili fiziologiju. Osobe s teškim kognitivnim poteškoćama trebaju pomoć u gotovo svakom aspektu svakodnevnog života. Osobe s manjim poteškoćama, primjerice u učenju, mogu biti u stanju funkcionirati normalno, čak i do te mjere da se poteškoća nikada ne dijagnosticira. [14]

Teži oblici kognitivnih poteškoća obuhvaćaju bolesti kao što su autizam ili Downov sindrom, a manje ozbiljna kognitivna stanja uključuju stanja kao što su poremećaj nedostatka pažnje, disleksiju (poteškoće s čitanjem) ili opće poteškoće u učenju.

Neki korisnici imaju poteškoća u pamćenju, što vodi do problema kod obrazaca s mogućim višestrukim porukama o pogreškama što može izazvati zaborav i frustraciju. Održavanje dosljednosti u dizajnu pomaže smanjiti memoriranje, te problem pamćenja se donekle može riješiti postavljanjem svojevrstnih podsjetnika na web stranicu s podacima o napretku, prethodnim i sljedećim koracima, stranicama i slično. Također, kod osoba s kognitivnim poteškoćama vezanih za izazove u rješavanju problema, niska tolerancija na frustraciju može ih natjerati da napuste zadatak ili web mjesto. To se može dogoditi s teškim CAPTCHA zagonetkama, kompliciranim porukama o pogrešci i nepredvidivim poveznicama. Takve



processe provjere autentičnosti je potrebno zamijeniti drugim metodama, primjerice dvofaktorska provjera autentičnosti. [14]

Mnogi pojedinci imaju poteškoća u zadržavanju fokusa na zadatku koji im predstoji. Smetnje kao što su različite animacije, skočni prozori ili razni dijaloški okviri mogu otežati ili čak onemogućiti dovršavanje zadatka. Osobe s poremećajem pažnje i hiperaktivnosti (ADHD) često nisu sposobni završiti određene zadatke, ne zbog nemogućnosti obrade informacija, već zbog raznih distrakcija zbog kojih gube pozornost. Iz tog razloga je bitno izbjegavati sve što bi odvrátilo pozornost osobe od glavnog sadržaja ili funkcionalnosti te dodatno naglasiti bitne dijelove sadržaja.

Osim toga, smanjene kognitivne sposobnosti se očituju i kroz slabije razumijevanje teksta, nerazumijevanje ili doslovno shvaćanje sarkazma, slabije zaključivanje te nemogućnost razumijevanja značenja i konteksta. U ovim situacijama autor jednostavno mora imati na umu kome prezentira određeni sadržaj, pisati razumljivo i jasno bez skrivenih značenja.

Kao i kod ostalih vrsta poteškoća, programeri jednostavno trebaju razumjeti i uzeti u obzir raspon korisnikovih sposobnosti pri dizajniranju web sadržaja.

### **3.5. Osobe s vestibularnim poremećajima i napadajima**

Neke osobe su osjetljive na napadaje uzrokovane raznim dramatičnim efektima kao što su treperenje ili bljeskanje. Napadaji su uzrokovani izmjenama svjetlosti koji utječu na neurone oka koji primaju svjetlost i središnji živčani sustav. U vestibularne poremećaje ubrajaju se i mučnina i vrtoglavica. Iako je većina web sadržaja uključujući razne animacije i pokretne elemente potpuno bezopasna, postoje određeni autori koji koriste dramatične izmjene svjetline u videozapisima ili određenim grafičkim formatima s ciljem privlačenja pažnje korisnika.

U smjernicama za pristupačnost za web definirani su učestalost bljeska, intenzitet, kontrast i boje koji se smatraju neprikladnim te se trebaju izbjegavati.

## 4. Osiguravanje pristupačnosti

U ovom radu će biti predstavljene osnovne smjernice za izradu aplikacija s prilagodbama za web pristupačnost, uz praktičan primjer implementacije određenih smjernica u konkretnoj aplikaciji razvijenoj u okviru kolegija "Okviri i alati za razvoj web aplikacija". Aplikacija je dostupna za preuzimanje putem priloženog GitHub linka.

Aplikacija je namijenjena praćenju troškova i budžeta korisnika, a razvijena je koristeći React JS okvir. Korisnicima omogućava kreiranje, evidenciju i pregled troškova te doprinosa, uz opcije filtriranja podataka prema određenim vremenskim intervalima. Pored toga, nudi vizualizaciju financijskih podataka, olakšavajući korisnicima praćenje njihovih troškova i raspodjelu budžeta. Prilikom unosa svakog troška, potrebno je odabrati kategoriju, poput vrste troška ili pripadajućeg bankovnog računa, čime se olakšava kasnija analiza podataka.

Važno je istaknuti da implementacija smjernica web pristupačnosti varira ovisno o vrsti aplikacije i ciljanih korisnika. Općenito su brojne prilagodbe već dostupne kroz funkcionalnosti samog preglednika, pa ih nije potrebno posebno implementirati, već je dovoljno testirati ih i prilagoditi postojećim elementima.

Također, pravilno korištenje HTML semantičkih oznaka može automatski osigurati mnoge aspekte pristupačnosti, zbog čega je ključno obratiti pozornost na semantičko značenje oznaka tijekom programiranja. Kao i općenito, ključno je rano prepoznati važnost pristupačnosti, jer se time olakšava kasnija implementacija i smanjuju potencijalni troškovi razvoja. Usvajanje smjernica za pristupačnost od samog početka omogućava lakšu integraciju ovih važnih elemenata u konačni proizvod.

### 4.1. Alati za testiranje pristupačnosti

Alati za testiranje i procjenu pristupačnosti weba su softverski programi ili mrežne usluge koje vam pomažu utvrditi zadovoljava li web sadržaj smjernice za pristupačnost. Prema trenutnom popisu kojeg W3C redovito ažurira, postoji čak 166 alata za testiranje pristupačnosti. [15]

Alati za procjenu pristupačnosti weba mogu pomoći brzom pronalasku potencijalnih problema s pristupačnošću, a mogu se koristiti u svim fazama procesa web dizajna i razvoja.

Alati mogu pružiti potpuno automatizirane provjere, međutim nije moguće provjeriti sve aspekte pristupačnosti na taj način već je potrebna ljudska prosudba. Ponekad alati za procjenu mogu proizvesti čak lažne ili pogrešne rezultate. [16]

Korištenje alata za procjenu dostupnosti weba uključuje sve one koji sudjeluju u kreiranju web sadržaja, od dizajnera, programera, autore sadržaja, testere, a ponekad i krajnje korisnike. Alati nude različite značajke i funkcionalnosti koje mogu pomoći korisnicima u usporedbi i procjeni alata za procjenu pristupačnosti weba za njihove specifične potrebe. Neki od njih se mogu pronaći kao dodaci u raznim web preglednicima, nekima se upravlja uz pomoć naredbenog retka, a najčešće su razvijeni kao zasebne aplikacije za računala. Ovisno o dizajnu i procesu razvoja moguće je, čak i poželjno, kombinirati više različitih alata. [16]

Jedan od poznatijih alata za provjeru pristupačnosti je „Accessibility Checker“. S obzirom da su smjernice WCAG duge i sklone čestim promjenama, alat je razvijen s namjerom premošćivanja jaza između smjernica i njegovih korisnika (programera, dizajnera). Alat je dostupan kao online aplikacija, a korištenje je jednostavno s obzirom da je potrebno upisati samo web stranicu koja se želi testirati. Alat izrađuje detaljno izvješće o mogućim problemima te ih klasificira s obzirom na važnost te skupinu ljudi kod kojih može uzrokovati poteškoće.

Još jedan popularan alat je „WAVE“, besplatni alat za procjenu pristupačnosti weba koji identificira načine kako web stranicu učiniti pristupačnijom osobama s invaliditetom. Proširenja za Chrome, Firefox i Edge omogućuju procjenu problema s pristupačnošću web-sadržaja izravno unutar preglednika. Također je praktičan za provjeru lokalno pohranjenih stranica. Za procjenu mrežnih stranica potrebno je unijeti URL stranice u polje nakon čega se pojavljuju rezultati procjene. Pruža izvrstan početni pregled web pristupačnosti, ali kao i većina drugih automatiziranih alata za testiranje pristupačnosti, ne može odrediti je li web sadržaj u potpunosti dostupan. Na primjer, ovi alati mogu provjeriti ima li oznaka `<img>` alternativni atribut, ali ne mogu provjeriti opisuje li alternativni atribut ispravno sliku. To je potrebno ručno potvrditi od strane nekog stručnjaka u razvojnom timu. Potpuno i isključivo oslanjanje na ovaj alat vjerojatno nije dobra ideja.

„WCAG Color Contrast Checker“ je još jedan popularan alat, ali za razliku od WAVE-a i Accessibility Checker-a koji pružaju čitav niz preporuka, ovaj alat je usmjeren samo na korištenje boja i kontrasta na stranici. Ipak, mnogo je elemenata koje ovo proširenje ne može

provjeriti, poput teksta u pozadinskim slikama. U tim je slučajevima obavezna ručna provjera.

Postoje alati koji mijenjaju izgled web stranice kako bi lakše identificirali probleme s dizajnom. Na primjer, alat može prikazati stranicu bez grafičkih elemenata koristeći samo tekst ili potpuno ukloniti boje.

## 4.2. Struktura stranice

Dobra struktura pomaže korisnicima da brzo pronađu ono što traže. To je posebno važno za osobe koje koriste čitače ekrana (npr. slijepi ili slabovidni korisnici), jer im jasna hijerarhija naslova, linkova i sekcija omogućava lakše razumijevanje sadržaja. Naslovi (od <h1> do <h6>) definiraju hijerarhiju informacija na stranici. Kada su naslovi pravilno označeni, korisnici mogu jednostavno preskakati s jedne sekcije na drugu korištenjem pomoćnih tehnologija.

Struktura stranice također osigurava da je sadržaj prezentiran na logičan način. To je važno za osobe s kognitivnim poteškoćama koje se oslanjaju na logičan slijed informacija. Ako je struktura kaotična ili nelogična, korisnici se mogu zbuniti i ne uspjeti ispravno konzumirati sadržaj. Navigacijski elementi, poput izbornika i *breadcrumbs*<sup>1</sup>, kada su pravilno označeni i postavljeni unutar strukture, pomažu korisnicima da razumiju gdje se nalaze na stranici i kako doći do drugih dijelova stranice. Ovo je posebno važno za osobe koje koriste tipkovnicu ili druge alternativne načine navigacije.

Struktura koja se temelji na standardima osigurava da će stranica biti kompatibilna s različitim pomoćnim tehnologijama poput čitača ekrana ili prepoznavanja govora. To omogućuje osobama s raznim vrstama invaliditeta da pristupe sadržaju na način koji im najviše odgovara. Web preglednici često imaju opciju „čitač“ čijom aktivacijom se pokazuje samo glavni sadržaj stranice, ukoliko je pravilno označen HTML oznakama. Ova opcija se češće koristi u preglednicima na mobilnim uređajima. Konzistentna struktura pomaže korisnicima da brzo nauče kako koristiti web stranicu i predviđaju gdje će se određeni

---

<sup>1</sup> breadcrumbs - navigacijski element na web stranicama koji korisnicima prikazuje njihovu trenutnu lokaciju unutar hijerarhije stranice

elementi nalaziti, što je općenito važno za sve korisnike, a posebno za osobe s kognitivnim poteškoćama ili one koje imaju teškoće s učenjem.

Web stranice bi stoga trebale biti podijeljene na nekoliko osnovnih regija. To su najčešće zaglavlje, podnožje, izbornik, glavni dio stranice i sporedni sadržaj.

Većina web-mjesta ima područje na vrhu svake stranice koje sadrži informacije o web-mjestu, poput logotipa web-mjesta, funkcije pretraživanja i mogućnosti navigacije. Za definiranje takve regije koristi se element `<header>`. Slično zaglavlju stranice, većina web-mjesta također ima područje na dnu svake stranice koje sadrži informacije o cijelom web-mjestu, kao što su informacije o autorskim pravima, izjave o privatnosti ili odricanje od odgovornosti. Element `<footer>` se koristi za definiranje takve regije.

Element `<nav>` se može koristiti za identifikaciju navigacijskog izbornika. Web stranica može imati neograničeni broj navigacijskih izbornika te za svaki od njih je potrebno koristiti oznaku.

Element `<main>` se koristi za definiranje glavne regije sadržaja web stranice ili aplikacije.

Za definiranje regija koje su u vezi sa glavnim sadržajem, a ipak su zasebne i smislene sekcije same za sebe koristi se element `<aside>`. Tim elementom se može označiti popratna bilješka koja objašnjava ili komentira glavni sadržaj.



Slika 1. Regije web stranice

Na manjim zaslonima takve regije ili komponente mogu biti sažete ili čak skrivene, ali komponente koje se prikazuju trebale bi se pojaviti istim redoslijedom za određeni prikaz.

Ako je moguće, struktura stranice trebala bi biti dosljedna u različitim veličinama i orijentacijama zaslona.

Većina modernih web preglednika podržava navedene HTML5 elemente i prenosi informacije pomoćnim tehnologijama putem API-ja za pristupačnost. No, kako bi se osigurala maksimalna kompatibilnost s preglednicima i pomoćnim tehnologijama koje podržavaju WAI-ARIA, ali još ne podržavaju HTML5, preporučuje se korištenje i HTML5 elemenata i odgovarajućih WAI-ARIA uloga. Ako HTML5 nije dostupan ili se HTML4 stranica prilagođava kako bi bila pristupačnija, potrebno je dodati WAI-ARIA uloge div elementima koji zamjenjuju HTML5 elemente.

```
<body>
  <header>
    <h1>Web pristupačnost</h1>
  </header>
  <nav aria-labelledby="izbornik">
    <h2 id="izbornik">Izbornik</h2>...
  </nav>
  <main>
    <article>
      <h2>Čitač zaslona</h2>
      <p>
        Za dodavanje botuna koji čita tekst naglas u React aplikaciji,
        možete koristiti Web Speech API, točnije SpeechSynthesis objekt.
      </p>
      <h3>SpeechSynthesis</h3>
      <p>
        SpeechSynthesis je dio Web Speech API-ja koji omogućuje konverziju
        teksta u govor.
      </p>
    </article>
    <aside aria-labelledby="povezano">
      <h3 id="povezano">Povezani članci</h3>
      <ul>
        <li>
          <a href="#">Povezani članak 1</a>
        </li>
        <li>
          <a href="#">Povezani članak 2</a>
        </li>
      </ul>
    </aside>
  </main>
  <footer>
    <p>Podnožje web stranice.</p>
  </footer>
</body>
```

Slika 2. Primjer pravilno strukturirane web stranice

### 4.3. Pristupačnost i WAI-ARIA oznake

Korištenjem WAI-ARIA (Web Accessibility Initiative – Accessible Rich Internet Applications) atributa poboljšava se pristupačnost dinamičkih elemenata na stranici koji inače ne bi bili pristupačni korisnicima čitača zaslona. Primjerice, interaktivni element kao što je botun bez upotrebe standardnog HTML <button> elementa može biti neupotrebljiv bez WAI-ARIA atributa.

Ovaj primjer koristi `div` kao botun, ali čitači zaslona ne bi prepoznali ovaj element kao botun jer element `<div>` nema semantičko značenje:

```

<div onclick="alert('Kliknuli ste!')" style="background-color: #007bff; color: white; cursor: pointer;">
  Klik
</div>

```

Slika 3. Element bez WAI-ARIA atributa

Iako će ovaj kod dati izgled botuna, korisnici koji se oslanjaju na čitače zaslona ili tipkovnicu neće moći koristiti ovaj element jer nedostaju potrebne informacije o ulozi elementa.

Koristeći WAI-ARIA attribute i dodatne prilagodbe, moguće je učiniti `<div>` element pristupačnijim. Dodavanjem ARIA atributa i upravljanjem događajima tipkovnice, korisnici čitača zaslona mogu prepoznati da je ovo botun.

```

<div role="button" tabindex="0"
  onclick="alert('Kliknuli ste!')"
  onkeydown="if (event.key === 'Enter' || event.key === ' ') { this.click(); }"
  style="background-color: #007bff; color: white; cursor: pointer;">
  Klik
</div>

```

Slika 4. Element `<div>` u ulozi botuna

Ključne promjene uključuju korištenje atributa `role="button"` koji obavještava čitač zaslona da se `div` element ponaša kao botun. Zatim, dodavanje `tabindex="0"` omogućuje navigaciju pomoću tipkovnice, što znači da se element može fokusirati pritiskom na tipku `Tab`. Konačno, atribut `onkeydown` osigurava da element reagira na pritisak tipki `Enter` ili razmaknice, replicirajući standardno ponašanje HTML botuna.

Bez WAI-ARIA atributa i prilagodbi, čitači zaslona ne bi prepoznali da se radi o botunu, a navigacija tipkovnicom bi bila nemoguća. Ovaj primjer jasno prikazuje kako WAI-ARIA poboljšava pristupačnost interaktivnih elemenata na webu.

Iako HTML `<button>` element sam po sebi pruža veliku pristupačnost (jer ga čitači zaslona prepoznaju i može se navigirati tipkovnicom), postoje situacije kada WAI-ARIA može



pružiti dodatnu funkcionalnost ili kontekst koji HTML sam po sebi ne nudi. Primjerice, kod botuna s funkcijom prebacivanja stanja (*toggle*<sup>2</sup>), ARIA atributi mogu jasno označiti stanje botuna, što je važno za korisnike čitača zaslona.

```
<button aria-pressed="false" id="toggleButton" onclick="toggleState()">
  Ukljuci
</button>

<script>
  function toggleState() {
    const button = document.getElementById('toggleButton');
    const isPressed = button.getAttribute('aria-pressed') === 'true';

    // Promijeniti aria-pressed stanje
    button.setAttribute('aria-pressed', !isPressed);
  }
</script>
```

Slika 5. Korištenje WAI-ARIA kod toggle botuna

Primjer prikazuje botun koji uključuje ili isključuje neki način rada, a ARIA atributi poboljšavaju korisničko iskustvo za osobe koje koriste čitače zaslona, jer jasno označavaju je li način rada trenutno uključen ili isključen.

ARIA atribut *aria-pressed="false"* označava da botun trenutno nije pritisnut i da nije aktiviran. Kada korisnik koristi čitač zaslona, čut će da je botun u stanju "isključeno" ili "nije pritisnut". Funkcija *toggleState()* omogućuje promjenu stanja atributa *aria-pressed* između "true" i "false" svaki put kada se botun klikne, čime se prikazuje njegovo trenutno stanje, bilo uključeno ili isključeno.

Prednosti korištenja WAI-ARIA u ovom primjeru odnose se na jasnu indikaciju stanja botuna. Atribut *aria-pressed* omogućuje čitačima zaslona da korisnicima jasnu predodžbu je li opcija omogućena ili onemogućena, što ne bi bilo moguće samo s HTML botunom.

---

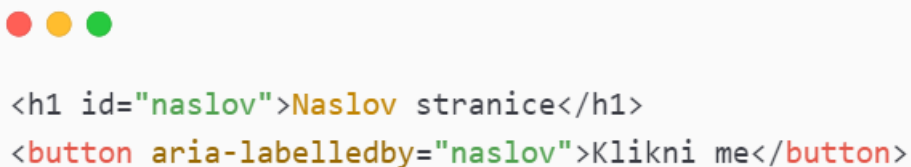
<sup>2</sup> Botun toggle - element na web stranici koji omogućuje prebacivanje između dva stanja, obično "uključeno" i "isključeno"

Korisnici čitača zaslona dobivaju dodatne informacije preko ARIA atributa koji označava je li botun "pritisnut" ili "nije pritisnut".

## 4.4. „Aria-label“ i „aria-labelledby“

*Aria-labelledby* i *aria-label* dio su ARIA specifikacije čija je uloga je da pružaju dodatne informacije o elementima na stranici kako bi korisnicima bilo lakše da razumiju funkciju ili sadržaj tih elemenata.

Atribut *aria-labelledby* koristi se za povezivanje elementa sa drugim elementom koji ga opisuje ili daje naziv. Ovaj atribut se postavlja na element koji treba opisati i kao vrijednost uzima ID elementa koji služi kao opis.



```
<h1 id="naslov">Naslov stranice</h1>
<button aria-labelledby="naslov">Klikni me</button>
```

Slika 6. Primjer korištenja "aria-labelledby"

U ovom slučaju, čitač ekrana će za dugme pročitati sadržaj elementa sa ID-jem naslov, tj. "Naslov stranice", umjesto teksta unutar botuna, što može pomoći korisniku da bolje razume njegovu funkciju.

Atribut *aria-label* koristi se za direktno definiranje tekstualnog opisa elementa. To je korisno kada nema odgovarajućeg vidljivog teksta koji bi element mogao opisati ili kada je tekst koji je vidljiv prekratak ili ne daje dovoljno informacija.



```
<button aria-label="Zatvori prozor">X</button>
```

Slika 7. Primjer korištenja "aria-label"

U ovom primjeru, botun s ikonom vizualno označava funkciju zatvaranja, no taj simbol nije dovoljno jasan za korisnike čitača ekrana. Korištenjem atributa *aria-label="Zatvori prozor"*, čitač zaslona može umjesto ikone pročitati tekst "Zatvori prozor".

## 4.5. Prilagodba slabovidnosti

Tamni način rada ili noćni način rada postavka je zaslona za digitalna korisnička sučelja. To znači da, umjesto zadanog tamnog teksta na svijetloj pozadini, korisnik vidi svijetli tekst na tamnoj ili crnoj pozadini. Osnovna ideja je smanjenje svjetlosti koju emitiraju zasloni uređaja, a istovremeno održava minimalne omjere kontrasta boja potrebne za čitljivost.

Primjer implementacije dodavanja tamnog načina rada prikazan je u aplikaciji spomenutoj ranije. Potrebno je osigurati korisnicima mogućnost promjene sa svijetlog na tamni način rada i obratno. Ova prilagodba pomaže osobama sa slabim vidom, fotofobijom (osjetljivost na svjetlost) ili daltonizmom.

U aplikaciju je potrebno dodati element, poput ikone ili botuna, koji će omogućiti promjenu kontrasta između svijetlog i tamnog načina rada. To je jedan od primjera kako prilagodbe mogu koristiti svima, ne samo osobama s poteškoćama, jer smanjuju naprezanje očiju, mogu smanjiti potrošnju baterije i poboljšavaju cjelokupno korisničko iskustvo. Danas je gotovo nezamislivo da popularna aplikacija ne nudi tamni način rada, bilo putem postavki unutar aplikacije, bilo sinkronizacijom s temom uređaja.

Najbolje rješenje je dodavanje CSS klase koja će definirati stilove za visoki kontrast. Budući da će odabir svijetlog ili tamnog načina rada utjecati na cijelu aplikaciju, važno je postaviti logiku za upravljanje temom u glavnu komponentu, *App()*.

Prvi korak je definiranje *hooka*<sup>3</sup> koji će sadržavati trenutno odabranu temu te funkciju za promjenu teme.

---

<sup>3</sup> hook - posebna funkcija koja omogućuje korištenje stanja i drugih značajki Reacta unutar funkcionalnih komponenti

```
● ● ●

const [isDarkTheme, setIsDarkTheme] = useState(false);

const toggleTheme = () => {
  setIsDarkTheme(!isDarkTheme);
};
```

Slika 8. React hook i funkcija za promjenu teme

Nakon toga je potrebno dodati ikonu ili tekst na čiji klik će se dogoditi promjena:

```
● ● ●

<div className="iconWrapper" onClick={toggleTheme}>
  {isDarkTheme ? '☀️ Svijetla tema' : '🌙 Tamna tema'}
</div>
```

Slika 9. Primjena funkcije za promjenu teme

Nakon toga je potrebno dodati klase u pripadajuću CSS datoteku te iskoristiti u kodu:

```
● ● ●

.light-theme {
  background-color: white;
  color: black;
}

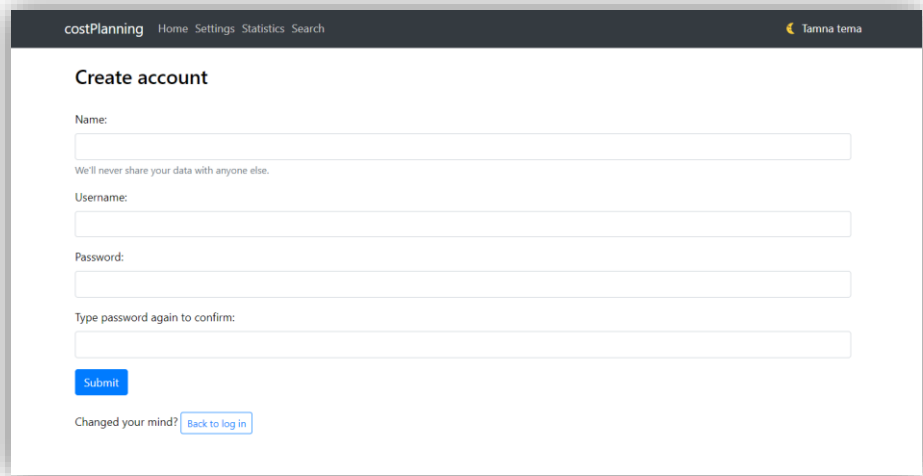
.dark-theme {
  background-color: black;
  color: white;
}
```

Slika 10. CSS klase i stilovi

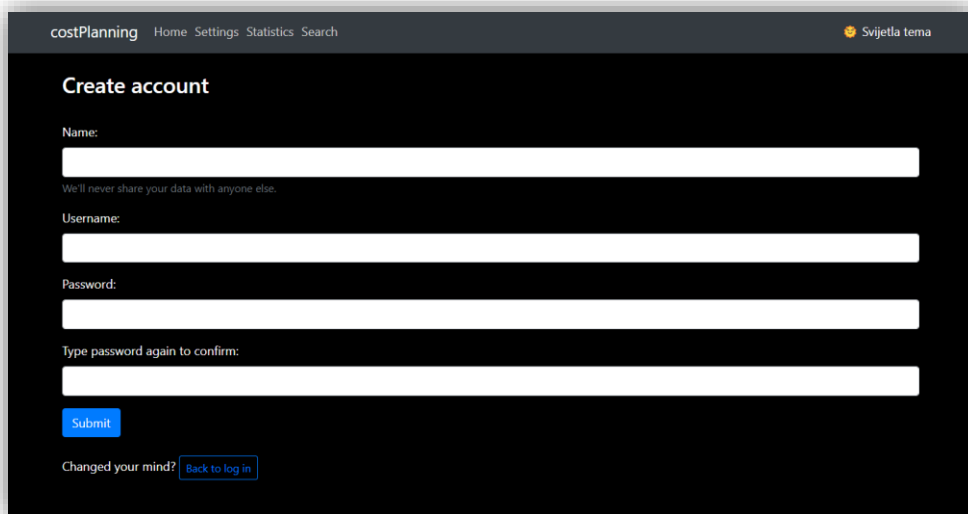
```
⋯  
<div className={isDarkTheme ? 'dark-theme' : 'light-theme'}>  
  /* ostatak koda */  
</div>
```

Slika 11. Primjena CSS klase

Napravljene izmjene omogućile su korisniku izmjenu između tamnog i svijetlog načina rada.



Slika 12. Svijetli način rada



Slika 13. Tamni način rada

Koristeći različite CSS selektore moguće je definirati izgled specifičnih HTML elemenata. Na primjer, moguće je definirati izgled botuna u tamnom načinu rada:

A screenshot of a code editor window with a white background and a grey border. At the top left, there are three colored circles: red, yellow, and green. The code is as follows:

```
.dark-theme button {  
  background-color: red;  
}
```

Slika 14. Izgled botuna u tamnom načinu rada

Osim promjene boja za tamnu i svijetlu temu, postoje brojne druge prilagodbe korisničkog sučelja koje je moguće napraviti na sličan način u CSS-u, koristeći klase i JavaScript za dinamičku primjenu stilova. To mogu biti promjena veličine teksta, dodavanje različitih razina kontrasta, promjena fonta, prilagodba razmaka između teksta, smanjivanje i uklanjanje animacija, dodavanje posebnih paleta boja za osobe sa specifičnim poteškoćama vida (npr. deuteranopija, protanopija), prilagodba širine sadržaja i sl.

Postoje brojne prednosti ovakvog načina rada aplikacije zbog kojih je i postao trend u dizajniranju web sučelja, ali ima i neke nedostatke. Tamni način rada smanjuje umor očiju i suhoću oka. Koristi slabovidnim osobama te osobama s osjetljivošću na svjetlo (fotofobija). Mobilni ili računalo također emitiraju manje plave svjetlosti, što prema nekim istraživanjima pomaže kod problema sa snom. Bijeli tekst na tamnoj pozadini podrazumijeva jak kontrast teksta i pozadine. Većim kontrastom postiže se bolja čitljivost, što sadržaj čini pristupačnijim. Konačno, tamni način rada za neke osobe s ADHD-om pomaže minimizirati smetnje koje stvaraju svijetli zasloni, a istovremeno smanjuje vizualni šum, čime pomaže zadržati fokus na primarnom sadržaju. [17]

S druge strane, visok kontrast također može stvoriti probleme s vidljivošću za određene alate za pristupačnost. Na primjer, tamni način rada može spriječiti da indikatori fokusa budu vidljivi korisniku koji se kreće samo pomoću tipkovnice. S obzirom na mogućnost implementacije korištenja tamnog načina rada na raznim sučeljima postoje razlike u kompatibilnosti koja se razlikuje među različitim uređajima, sustavima i platformama. Kao rezultat toga, ono što bi moglo ispravno funkcionirati na jednom uređaju ili pomoćnoj tehnologiji može naići na probleme na drugom. Ovime se stvara potreba za usvajanje

određene standardizacije jer trenutno ne postoje određene prakse prihvaćene u cijeloj industriji. [17]

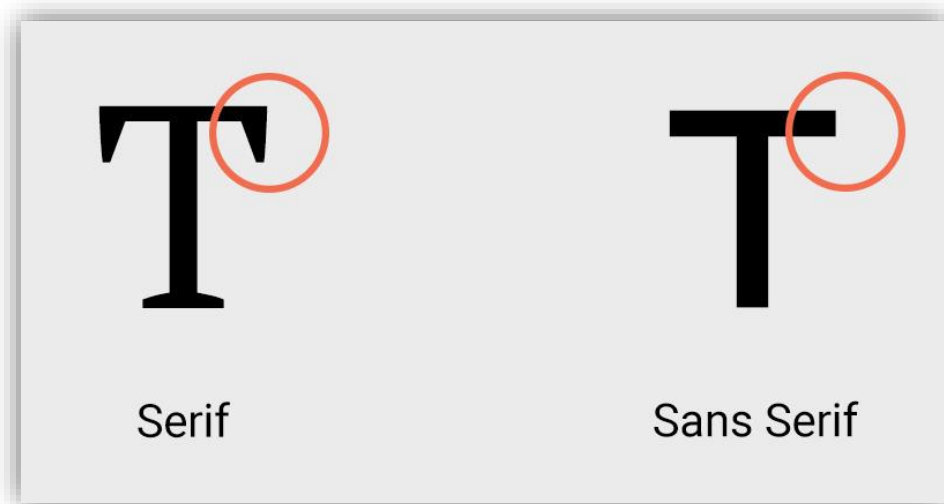
## 4.6. Prilagodba disleksiji

Disleksija je stanje kod kojeg osobe mogu doživjeti vrlo različite izazove, iako imaju istu dijagnozu, jer se manifestira različito kod svakog pojedinca. Na primjer, neki ljudi imaju problema s brzinom čitanja, drugi s razumijevanjem ili pisanjem, a treći možda imaju poteškoće s kratkoročnim pamćenjem. Kako bi se osiguralo da nove funkcionalnosti u aplikacijama i na web stranicama doista odgovaraju potrebama raznolike korisničke baze nužno je temeljito testiranje.

Prilagodba za disleksiju na web stranici često uključuje promjenu vrste fonta, razmake između redaka, paragrafa i slova kako bi tekst bio lakši za čitanje osobama s disleksijom. Preporučuje se korištenje fontova koji su dizajnirani s obzirom na disleksiju ili koji imaju određene karakteristike koje olakšavaju čitanje.

Općenito se fontovi dijele na 5 vrsta – *serif* (imaju poteze na rubovima slova, stvaraju osjećaj formalnosti), *sans-serif* (imaju jasne linije, moderan i minimalistički izgled), *monospace* (sva slova imaju fiksnu širinu), *cursive* (oponašaju ljudski rukopis) i *fantasy* (dekorativni font).

Za ljude s disleksijom koriste se *sans-serif* fontovi, a najčešći su „OpenDyslexic“ (specijalno dizajniran font za osobe s disleksijom), „Helvetica“, „Verdana“ i „Arial“, a dostupni su u svakom većem operacijskom sustavu, pregledniku i aplikaciji.



Slika 15. Usporedba "serif" i "sans serif" fonta

Implementacija mogućnosti izmjene fonta ne razlikuje se mnogo od implementacije za izmjenu kontrasta aplikacije. Nakon dodavanja hooka, funkcije i botuna za izmjenu, potrebno je definirati samo font u CSS datoteci:

```
.dyslexia-font {  
  font-family: 'OpenDyslexic', Arial, sans-serif;  
}
```

Slika 16. Primjena specifičnog fonta za ljude s poteškoćama

Osim fonta i razmaka između elemenata, prilagodbe mogu uključivati i grupiranje elemenata na stranici na osnovu konteksta ili sadržaja, kontrast uz izbjegavanje kombinacije boja koje mogu biti zbunjujuće (npr. crvena i zelena zbog daltonizma), razdvajanje velikih blokova teksta na manje dijelove, korištenje lista ili numeriranih elemenata čime se poboljšava jasnoća, izbjegavanje kurziva i podcrtavanja koji mijenjaju oblik slova.

## 4.7. Prilagodba veličine fonta

Fleksibilnost u prilagodbi veličine fonta i osiguravanje čitljivosti ključni su aspekti pristupačnosti na webu. Jedan od osnovnih principa pristupačnosti je omogućiti korisnicima da prilagode veličinu teksta. Prema smjernicama za pristupačnost web sadržaja (WCAG)



korisnici bi trebali moći povećati veličinu teksta do 200% bez gubitka sadržaja ili funkcionalnosti. Iako nije nužno pružiti posebne kontrole za promjenu veličine fonta na stranici, važno je da korisnici mogu lako napraviti te prilagodbe putem funkcije zumiranja u pregledniku. Mnogi korisnici, osobito oni sa slabijim vidom ili oni koji pristupaju sadržaju na manjim ekranima, oslanjaju se na ovu mogućnost povećavanja teksta kako bi ga učinili čitljivijim. Web preglednici, kao korisnički agenti, moraju podržavati ovu značajku prema Smjernicama za pristupačnost korisničkih agenata (UAAG), omogućujući korisnicima slobodu prilagodbe veličine teksta.

Jedna od ključnih preporuka za skaliranje teksta je izbjegavanje upotrebe apsolutnih veličina fonta, kao što su inči (in). To može ograničiti sposobnost korisnika da učinkovito povećaju sadržaj na različitim uređajima. Umjesto toga, korištenje relativnih jedinica poput "em" ili postotaka omogućuje tekstu da se proporcionalno povećava u odnosu na ostale elemente na stranici, što osigurava bolje iskustvo kada korisnici trebaju povećati ekran. Relativne veličine su znatno prilagodljivije, pomažući da sadržaj ostane pristupačan bez obzira na preglednik, platformu ili uređaj koji korisnici koriste.

Iako WCAG ne propisuje određenu minimalnu veličinu fonta, pridržavanje najboljih praksi za čitljivost teksta je ključno. Tekst tijela trebao bi idealno biti najmanje 16 piksela, a nijedan tekst ne bi trebao biti manji od 12 piksela.

Jednostavan način za procjenu pristupačnosti web stranice je povećanje preglednika na 200%. Ovaj test simulira kako bi korisnici s oslabljenim vidom ili oni koji preferiraju veći tekst mogli doživjeti vaš sadržaj. Bitno je da se tekst pravilno skalira u odnosu na druge elemente na stranici te da se sadržaj ne preklapa niti nestaje, te da stranica osigurava logičan redoslijed čitanja i da korisnici ne moraju pretjerano pomicati u više smjerova kako bi pročitali sav sadržaj.

## **4.8. Navigacija tipkovnicom**

Navigacija tipkovnicom jedan je od najvažnijih aspekata pristupačnosti weba. Mnogi korisnici s motoričkim poteškoćama oslanjaju se na tipkovnicu. Uz tradicionalne tipkovnice, neki korisnici mogu koristiti modificirane tipkovnice ili drugi hardver koji oponaša funkcionalnost tipkovnice. Tipkovnicu za navigaciju često koriste i ljudi bez invaliditeta zbog preferencija ili učinkovitosti.

Implementacija navigacije tipkovnicom je uglavnom jednostavna, a postiže se pravilnim korištenjem semantički ispravnih HTML elemenata koji, osim što utječu na funkcionalnost, interaktivnost, SEO i stilizaciju web stranice, prirodno podržavaju pristup putem tipkovnice. Na primer, `<input>`, `<button>`, i `<a>` tagovi imaju ugrađene događaje za tipkovnicu kao što su *tab*, *enter* i *space* tipke. Ako se element može fokusirati pomoću tipkovnice, onda bi trebao biti interaktivan. Korisnik bi trebao biti u mogućnosti učiniti nešto s njim i proizvesti neku vrstu promjene (na primjer, aktivirati vezu ili promijeniti opciju).

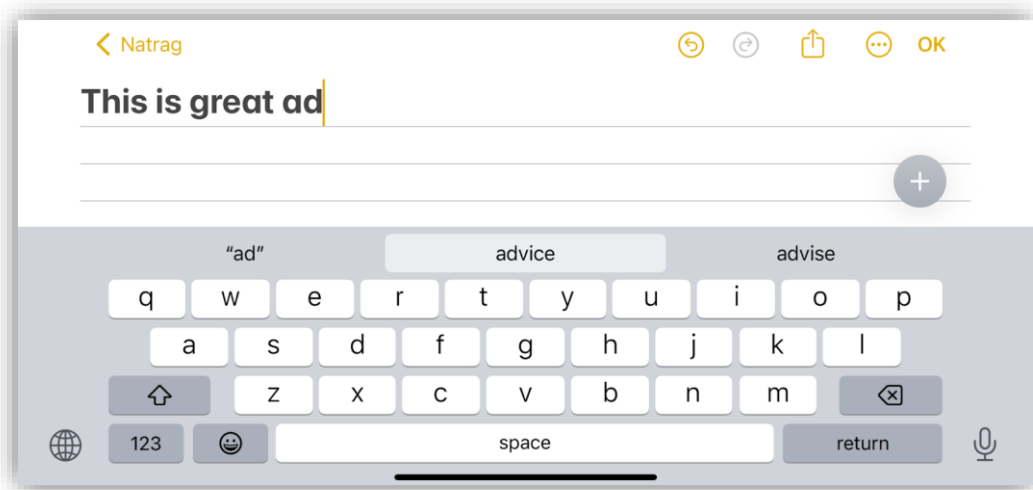
Zbog specifičnih zahtjeva, ponekad je potrebno implementirati određene interaktivne komponente koje ne koriste HTML elemente koji se automatski fokusiraju korištenjem tipkovnice. Element je moguće učiniti fokusiranim dodavanjem vrijednosti atributa *tabindex=0*. Vrijednost 0 označava da je element dio zadanog redoslijeda fokusa, koji se temelji na redoslijedu elemenata u HTML dokumentu. Pozitivna vrijednost stavlja element ispred onih u zadanom redoslijedu; elementi s pozitivnim vrijednostima fokusirani su redoslijedom njihovih vrijednosti *tabindexa* (1, zatim 2, zatim 3, itd.). To stvara zabunu za korisnike koji koriste samo tipkovnicu kada se redoslijed fokusa razlikuje od logičnog redoslijeda stranice. Stoga je bolje strukturirati HTML dokument tako da elementi koji se mogu fokusirati budu u logičnom redoslijedu, bez potrebe za mijenjanjem s pozitivnim vrijednostima *tabindexa*. Zadani redoslijed kretanja tipkovnicom mora biti logičan i intuitivan. To općenito znači da slijedi vizualni tijek stranice - slijeva nadesno, odozgo prema dolje. Za većinu stranica to znači prvo zaglavlje, zatim glavna navigacija, zatim navigacija po stranici i na kraju podnožje. S druge strane, ukoliko se s nekog elementa želi ukloniti opcija upravljanja tipkovnicom potrebno je dodati atribut *tabindex = -1*. [18] [19]

Također je bitno osigurati vizualnu povratnu informaciju prilikom prelaska fokusa na određene elemente putem tipkovnice. Ovo se postiže putem CSS pravila koja stiliziraju stanje elementa kada je fokusiran (pomoću pseudo-klase *:focus* i *:focus-visible*). Na primjer, može se primijetiti promjena boje ili okvira kada element dobije fokus, što pomaže korisnicima da jasno vide koji element je trenutno aktivan. Standardnim elementima koji se mogu fokusirati, kao što su veze i polja za unos, preglednik prema zadanim postavkama daje poseban stil, tako da često nije niti potrebno navesti stil fokusa za takve elemente, osim ako je potrebno promijeniti stil fokusa da bude karakterističniji. Kod izrade vlastitih komponenata koje se mogu fokusirati, svakako je potrebno definirati stil fokusa za njih. Bitno je i da elementi koji se mogu fokusirati, bilo da su ugrađeni ili ne, uistinu i obavljaju

neku akciju. Omogućavanje navigacije neinteraktivnim elementima tipkovnicom izazvat će zabunu.

Kod složenijih aplikacija često je potrebno koristiti JavaScript za unapređenje pristupačnosti komponenti kao što su modalni prozori, padajući izbornici i „*tooltips*“<sup>4</sup>. JavaScript osigurava da se korisnici mogu kretati kroz ove komponente pomoću tipke *Tab*, kao i da mogu koristiti tipke poput *Escape* za zatvaranje modala ili menija.

Predviđanje teksta dodatna je funkcionalnost koja poboljšava učinkovitost i pristupačnost tipkovnica, posebno za korisnike sa smanjenom mobilnošću ili one koji preferiraju bržu interakciju. Ova značajka omogućuje zaslonskoj tipkovnici da predlaže riječi dok korisnik piše, čime se skraćuje vrijeme potrebno za unos teksta jer nije potrebno tipkati cijelu riječ. Korisnik može odabrati željenu predloženu riječ, čime se ubrzava proces pisanja. Nedostatak je često manjkavost izbora jezika koji se mogu koristiti za predviđanje pa su najčešće uključeni oni najzastupljeniji jezici poput engleskog, španjolskog, talijanskog i sl.



Slika 17. Predviđanje teksta na sustavu iOS

Napredne tipkovnice sve češće koriste umjetnu inteligenciju (AI) za personaliziranje i unapređenje korisničkog iskustva. AI modeli analiziraju navike pisanja korisnika te s vremenom prilagođavaju prijedloge kako bi bili precizniji i relevantniji. Također, AI omogućuje predviđanje cijelih fraza, prepoznavanje čestih tipfelera i njihovu automatsku ispravku, čime se dodatno poboljšava brzina i točnost unosa.

---

<sup>4</sup> tooltip - mali, obično privremeni, vizualni element koji se pojavljuje kada korisnik ostvari interakciju s određenim dijelom sučelja, pruža dodatne informacije vezan za određeni element

## 4.9. Opisi i transkripti za multimediju

Osim za ljude s poteškoćama, prilagodbe mogu biti korisne svima. Dodavanje opisa i transkripata za videozapise pomaže ljudima u glasnom okruženju gdje se ne može čuti zvuk, u okruženjima gdje nije prikladno pojačati zvuk te za ljude kojima je lakše čitati tekst jer nisu izvorni govornici tog jezika. Osim toga, transkripti se mogu preuzeti i prevesti na drugi jezik, među ostalima i na znakovni jezik. Pristupačnost na web stranici može povećati broj korisnika te može poboljšati indeksiranje od strane tražilica. [20]

Titlovi pružaju sadržaj gluhim i nagluhim osobama. Titlovi su tekstualna verzija govornih i negovornih audio informacija potrebnih za razumijevanje sadržaja. Sinkronizirani su sa zvukom i obično se prikazuju u *medijskom playeru*<sup>5</sup> kada ih korisnici uključe. Koriste ih i ljudi koji bolje obrađuju pisane informacije nego audio. Mogu biti dio videozapisa, ali najčešće se mogu zasebno uključiti ili isključiti u prikazu videa. Često se koriste i u svrhu prikazivanja prijevoda videozapisa.

Dodavanje transkripta za audio datoteke prilično je teško i oduzima dosta vremena ljudima koji nemaju softver i vještine za to. Format datoteke za titlove je jednostavan, ali je zamorno dodavati vremenske oznake, pogotovo bez prikladnog softvera. Stvaranje titlova zahtijeva znanje o tome koje negovorne audio informacije treba uključiti u titlove jer nije uvijek jasno koje negovorne audio informacije uključiti i kako ih prenijeti tekstem. Automatski generirani titlovi često ne zadovoljavaju potrebe korisnika ili zahtjeve pristupačnosti pa je obično potrebno značajno uređivanje. Postoje alati koji koriste tehnologiju prepoznavanja govora za pretvaranje zvučnog zapisa u datoteku s vremenskim titlovima. Međutim, često je automatski tekst opisa pogrešan i ne odgovara izgovorenom zvuku, a ponekad i na način koji potpuno mijenja značenje. [21]

Dodavanje znakovnog jezika ima ogroman značaj za gluhe osobe. Znakovni jezici koriste pokrete ruku, izraze lica i položaje tijela kako bi prenijeli značenje. Znakovni jezik je materinji jezik mnogih osoba koje su gluhe. Neki ljudi imaju poteškoća s čitanjem ili ne razumiju dobro pisani jezik, osobito pri brzini većine titlova, a neki žele imati znakovni jezik i titlove u isto vrijeme. Ne znaju svi koji su gluhi znakovni jezik, pogotovo ako su kasnije u životu postali gluhi ili nagluhi. Znakovni jezici razlikuju se po regijama i zemljama. Na

---

<sup>5</sup> medijski player - softverski alat koji omogućuje reprodukciju različitih vrsta medijskog sadržaja, poput video i audio datoteka

primjer, postoje razlike između američkog znakovnog jezika (ASL), britanskog znakovnog jezika (BSL) i australskog znakovnog jezika (Auslan). Postoje određeni napori da se omogući automatski znakovni jezik iz teksta, međutim, avatari koji simuliraju tumačenje znakovnog jezika nisu dovoljno robusni da bi bili primjereni.

Slike moraju biti predstavljene i tekstualno na način da opisuju informacije koje prikazuju. Slike koje grafički predstavljaju koncepte i informacije, slike, fotografije i ilustracije trebaju imati tekst koji barem ukratko opisuju bitne informacije prikazane slikom. Kod slika kojima je jedina svrha vizualno poboljšanje stranice, a ne prijenos važnih informacija dovoljno je iskoristiti atribut *alt*:

```

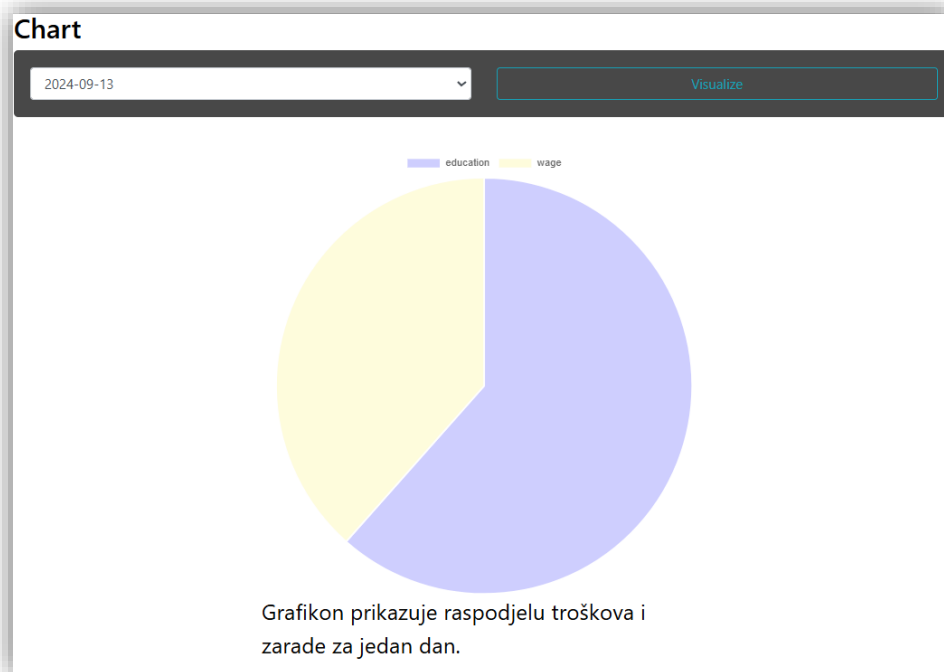
```

Slika 18. Primjer korištenja atributa "alt"

Kod složenijih slika kao što su grafikoni i dijagrami bitno je navesti potpuni tekstualni ekvivalent podataka ili informacija navedenih na slici kao tekstualnu alternativu. WAI-ARIA atribut „*aria-describedby*“ može se koristiti za povezivanje s opisom slike koji se nalazi bilo gdje na istoj web stranici. Vrijednost atributa je ID elementa koji daje opis. Primjer dodavanja tekstualnog opisa se može vidjeti u aplikaciji, u slučaju dodavanja opisa za element `<canvas>` koji je iskorišten za prikaz grafikona:

```
<canvas id="chart-area" aria-describedby="chart-description" />  
  
<p id="chart-description">  
  Grafikon prikazuje raspodjelu troškova i zarade za jedan dan.  
</p>
```

Slika 19. Primjer dodavanja tekstualnog opisa za grafikon

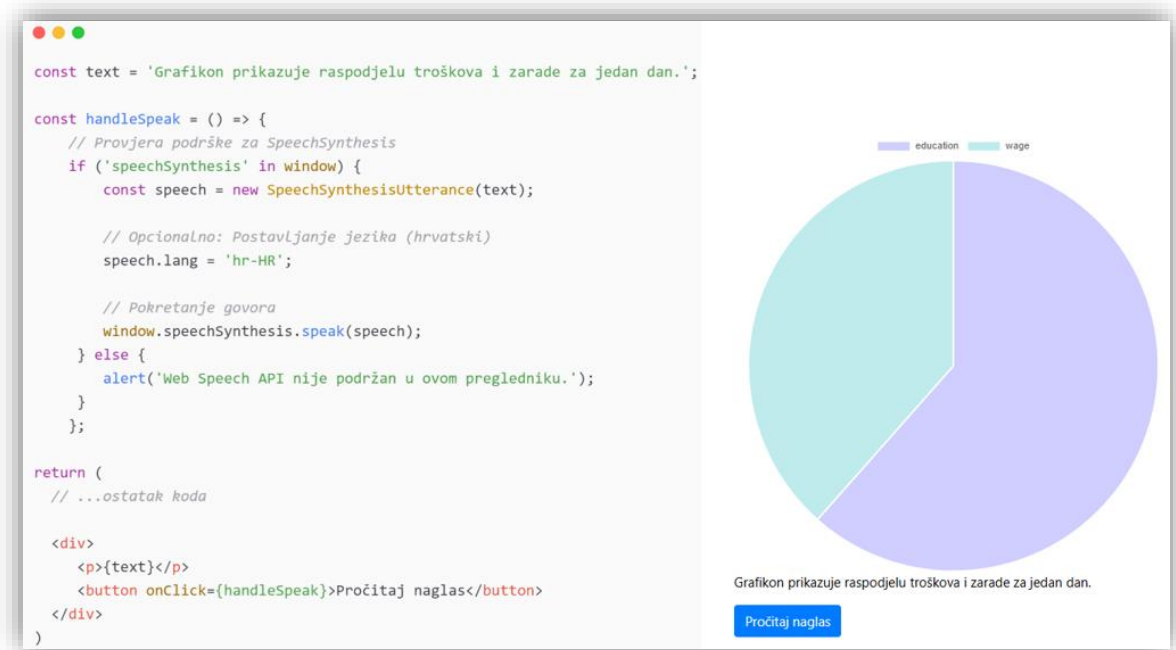


Slika 20. Izgled grafikona i alternativnog teksta u aplikaciji

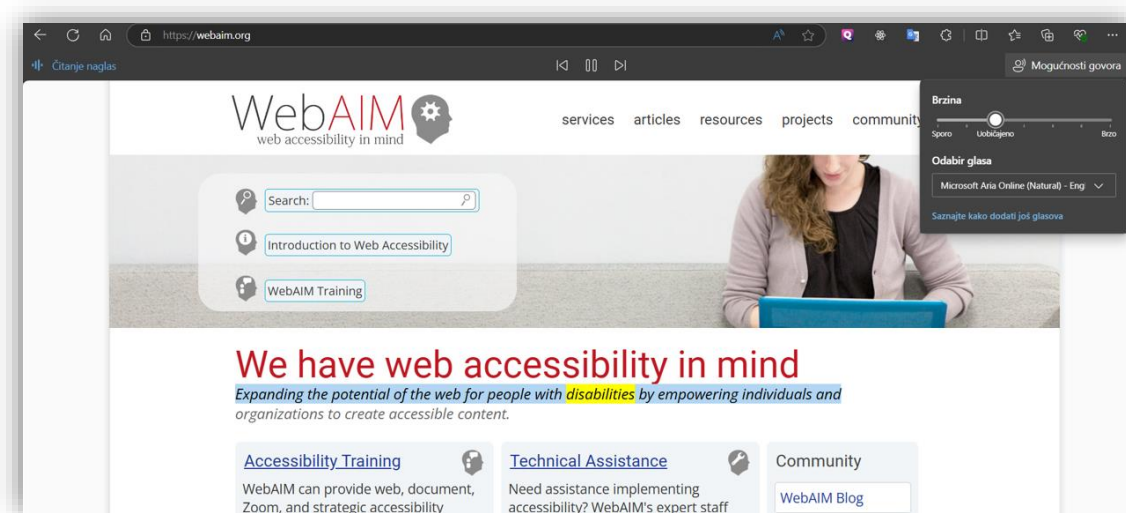
## 4.10. Čitač zaslona

Mnoga računala i mobilni uređaji imaju ugrađeni softver za pretvaranje teksta u govor koji se zove čitači zaslona. Čitači zaslona pružaju važne funkcije kao što je navigacija kroz naslove, izgovorene slikovne alternative i prepoznavanje poveznica. Mogu istaknuti tekst dok se čita naglas kako bi ljudi mogli vidjeti i čuti sadržaj u isto vrijeme. Sadržaj mora biti ispravno kodiran tako da sve funkcije softvera za pretvaranje teksta u govor rade sa sadržajem.

Iako moderni preglednici i operativni sustavi često dolaze s ugrađenim čitačima zaslona i drugim pomoćnim tehnologijama, programeri mogu jednostavno implementirati osnovnu funkcionalnost čitanja teksta naglas putem *Web Speech API*-ja, točnije korištenjem „*SpeechSynthesis*“. Ovaj API omogućuje web stranicama da sintetiziraju govor iz teksta, pružajući korisnicima mogućnost da čuju sadržaj bez potrebe za dodatnim softverom. Ova tehnika može biti korisna kao nadopuna ugrađenim čitačima zaslona, omogućujući prilagođeno čitanje određenih dijelova stranice ili specifičnih akcija na zahtjev korisnika, što dodatno poboljšava pristupačnost web sadržaja.



Slika 21. Primjer implementacije čitača zaslona



Slika 22. Čitač zaslona u pregledniku Microsoft Edge

Dodavanje čitača zaslona direktno na web stranicu može imati smisla, ali zavisi od ciljne publike i funkcionalnosti koje se žele postići. Postoje različiti razlozi za i protiv takvog pristupa. Prednosti uključuju veću pristupačnost za sve korisnike, jer ne koriste svi isti preglednik ili možda nisu tehnički potkovani da omoguće funkcije čitača zaslona. Integrirani čitač zaslona osigurava da svi korisnici, bez obzira na tehničke vještine, mogu pristupiti sadržaju. Također, omogućava prilagođavanje korisničkog iskustva te kontrolu nad načinom

na koji se sadržaj prikazuje osobama s oštećenjem vida. Ovo pruža i mogućnost brendiranja, jer neki brendovi preferiraju potpuno integrirano iskustvo koje odražava njihov vizualni identitet.

Međutim, postoje i nedostaci. Kao što je spomenuto, većina preglednika već ima ugrađene čitače zaslona, a korisnici koji se oslanjaju na te alate često koriste specijalizirane softvere kao što su JAWS ili NVDA, pa dodavanje dodatnog čitača može biti suvišno. Također, održavanje takvog čitača zahtijeva stalno prilagođavanje, testiranje i ažuriranje kako bi bio kompatibilan sa svim uređajima i platformama, što može biti skupo i zahtjevno. Performanse web stranice mogu biti narušene dodavanjem dodatnog softverskog sloja, što negativno utiče na korisnike koji ne koriste te funkcije. Umjesto integracije posebnog čitača, bolji pristup može biti fokus na usklađivanje sa standardima pristupačnosti kako bi stranica bila kompatibilna sa postojećim alatima za pristupačnost.

## 4.11. Animacije i interaktivni elementi

Web animacije mogu biti pristupačne, ali ponekad zahtijevaju dodatan trud programera i dizajnera. Postoje koraci kojima se može osigurati da animacije pozitivno utječu na pristupačnost te njihovog doprinosa cjelokupnom korisničkom iskustvu.

WCAG daje preporuke o tome kada treba omogućiti kontrole za pauziranje i reprodukciju, ograničiti treptanje ekrana, te ponuditi opcije sa smanjenim pokretom za korisnike osjetljive na pokret.

Jedna od glavnih preporuka je dodavanje opcija pauziranja, zaustavljanja i sakrivanja za automatski pokretane animacije koje traju dulje od pet sekundi. Preporučuje se dodavanje kontrola za pauziranje i reprodukciju kod dugotrajnijih animacija poput automatski animiranih *carousela*<sup>6</sup> ili animiranih pozadina. Poželjno je zaustaviti animaciju kada pokazivač miša prelazi iznad takvog elementa ili kada dobije fokus tipkovnice. Pauziranje pri prijelazu mišem korisno je za ljude kojima treba više vremena za čitanje sadržaja i olakšava odabir linkova. [22]

---

<sup>6</sup> carousel - interaktivni element na web stranicama koji omogućuje prikazivanje i pregledavanje niza slika, sadržaja ili informacija u obliku rotirajuće galerije



Potrebno je izbjegavati upotrebu treperećih elemenata, jer mogu izazvati epileptičke napade kod osoba osjetljivih na svjetlosne podražaje. Takvi elementi bi trebali biti zamijenjeni statičkim vizualima ili koristiti druge oblike animacija sa smirenijim prijelazima.

## 4.12. Pohrana korisnikovih preferencija

Sve navedene prilagodbe omogućuju bolje korisničko iskustvo i osiguravaju da web stranice budu dostupne svima, bez obzira na njihove potrebe. Međutim, uz sve navedene prilagodbe postavlja se i pitanje kako pohraniti te postavke na način da budu lako dostupne i trajne za korisnika. Jedno od mogućih rješenja je korištenje „*local storage*“, tehnologije pohrane unutar preglednika, koja ima i svoje prednosti i nedostatke.

*Local storage* je tehnologija pohrane podataka unutar preglednika, koja omogućava web aplikacijama da pohrane podatke lokalno na korisničkom uređaju. Koristi se za pohranu relativno malih količina podataka, a podaci pohranjeni ovim putem ostaju dostupni i nakon zatvaranja preglednika, dok ih korisnik ručno ne izbriše ili ne očisti „*cache*“<sup>7</sup>.

Jedna od ključnih prednosti pohrane postavki pristupačnosti u *local storage* je trajnost. Korisnici koji, primjerice, imaju oštećenje vida, mogu prilagoditi veličinu fonta na web stranici kako bi lakše čitali sadržaj. Kada su te postavke spremljene u memoriju, korisnik neće morati ponovno podešavati te postavke svaki put kada posjeti stranicu ili osvježi preglednik. Na taj način, pruža se konzistentno i personalizirano iskustvo, što doprinosi boljoj pristupačnosti i jednostavnijem korištenju web stranice. Osim toga, pohrana podataka se obavlja direktno na korisničkom uređaju, što znači da su postavke dostupne lokalno, bez potrebe za komunikacijom sa poslužiteljem svaki put kada se stranica ponovo učitava. To rezultira bržim učitavanjem stranice i boljim performansama.

Još jedna prednost pohrane postavki pristupačnosti u *local storage* jest ta što korisnik ne mora biti prijavljen na web stranicu kako bi mu postavke bile sačuvane. Dok bi pohranjivanje tih podataka na poslužitelju zahtijevalo korisnika da bude prijavljen, na ovaj se način omogućuje spremanje postavki za sve posjetitelje stranice, bez obzira na to jesu li registrirani korisnici ili ne. Ovo je posebno korisno za web stranice koje ne zahtijevaju korisničke račune, ali ipak žele omogućiti prilagodbu pristupačnosti svim posjetiteljima.

---

<sup>7</sup> cache - brza, privremena vrsta memorije koja se koristi za pohranu često korištenih podataka kako bi se ubrzao pristup tim podacima, smanjuje potrebu za sporijim pristupom glavnoj memoriji

S tehničke strane, ovo je relativno jednostavan način za pohranu i dohvaćanje postavki pristupačnosti. Tehnologija je dobro podržana u modernim preglednicima i lako se integrira putem JavaScripta.

Nedostatak pohrane u *local storage* je taj što su podaci vezani isključivo uz određeni preglednik i uređaj čime korisnik, ako promijeni uređaj ili koristi drugi preglednik, gubi svoje prilagođene postavke. Primjerice, korisnik može podešavati veličinu fonta na desktop računalu, no te postavke neće biti primijenjene kada pristupi istoj web stranici s mobilnog uređaja. Ovo ograničenje može frustrirati korisnike koji očekuju konzistentno iskustvo na svim uređajima.

Osim toga, ima ograničen kapacitet, koji se obično kreće između 5 i 10 MB po domeni, ovisno o pregledniku. Iako je to dovoljno za pohranu postavki pristupačnosti kao što su veličina fonta i kontrast, ovo ograničenje može postati problematično ako web stranica pohranjuje veću količinu podataka lokalno. U slučaju da se kapacitet premaši, web stranica neće moći pohraniti nove podatke, što može dovesti do nepravilnog funkcioniranja postavki pristupačnosti.

U konačnici, odluka o korištenju *local storagea* ovisi o specifičnim potrebama web stranice i njezinih korisnika. Ako je cilj pružiti jednostavno i brzo rješenje za personalizaciju pristupačnosti, *local storage* može biti prikladan alat. No, za korisnike koji očekuju konzistentno iskustvo na više uređaja mogla bi biti potrebna složenija rješenja, poput pohrane podataka na serveru.

# Zaključak

Web pristupačnost je ključna komponentna suvremenog razvoja web aplikacija i stranica. Povijesno gledano, interes za pristupačnost potječe iz 1990-ih kada su doneseni zakoni poput Američkog zakona o osobama s invaliditetom (ADA) kada se počelo razmišljati o tome kako tehnologija može osigurati jednak pristup informacijama svim korisnicima. Pristupačnost web platformi osigurava da ljudima s različitim vrstama izazova - bilo da su fizičke prirode ili se odražavaju na kognitivnu sposobnost - omogućuje jednak pristup digitalnim sadržajima u istoj mjeri kao i ostalim korisnicima. Razvoj web stranica koje su prilagođene svima nije samo moralna obveza već također i zakonska obaveza u mnogim zemljama.

Prvi dio rada bavi se općenito o ljudima s poteškoćama, identificira različite vrste poteškoća i kako se one mogu manifestirati u digitalnom okruženju. Poteškoće mogu biti vizualne (sljepoća, slabovidnost, daltonizam), auditivne, motoričke (problemi s kretanjem ili upravljanjem mišem) i kognitivne (problemi s koncentracijom, pamćenjem ili učenjem). Kroz ovaj pregled, ukazano je na specifične izazove s kojima se ovi korisnici suočavaju te su analizirani postojeći alati i tehnologije koji im olakšavaju pristup sadržajima, poput čitača ekrana, uvećanja teksta ili prepoznavanja glasa.

Drugi dio rada sadrži glavne smjernice za izradu pristupačne web stranice. Ove smjernice su često definirane kroz WCAG (Web Content Accessibility Guidelines), koje uključuju preporuke o prilagodbi boja, fonta, kontrasta, navigacije i multimedije. Kroz demo aplikaciju su implementirani neki od ovih principa – poput čitača zaslona i mogućnosti prilagodbe kontrasta te mogućnost prilagodbe veličine fonta. Također su dodani i primjeri za neke česte slučajeve koji nisu izravno implementirani, ali su objašnjeni, poput pravilne upotrebe HTML oznaka s ciljem ostvarivanja mogućnosti navigacije s tipkovnicom.

U konačnici, važno je istaknuti da stvaranje pristupačnih web stranica ima za cilj uključivanje svih korisnika, treba imati na umu i ciljnu publiku aplikacije ili stranice. Ne postoji univerzalno rješenje će odgovarati svima, ali razvijanje sa sviješću o potrebama različitih skupina korisnika može znatno poboljšati kvalitetu i dostupnost digitalnih sadržaja. Stoga, web pristupačnost treba biti sastavni dio svakog dizajnerskog i razvojnog procesa, kako bi svi, bez obzira na njihove sposobnosti, imali jednak pristup informacijama i uslugama.

# Literatura

- [1] Medium, *Web Accessibility – Make your web application more accessible* – Ankita Zaveri, 20.08.2023., <https://medium.com/geekculture/web-accessibility-make-your-web-application-more-accessible-af24b4292db0>
- [2] M. Gradiški, V. Cviljušac, T. Skala, *Prilagodba korisničkih sučelja primjenom principa inkluzivnog dizajna radi bolje pristupačnosti osobama starije životne dobi*, Zbornik radova, Printing&Design 2022, 86.
- [3] Središnji državni ured za razvoj digitalnog društva, *Digitalna pristupačnost*, 21.08.2023., <https://rdd.gov.hr/digitalna-pristupacnost>
- [4] W3C - Web Accessibility Initiative, *Introduction to Web Accessibility*, <https://www.w3.org/WAI/fundamentals/accessibility-intro/>, 23.08.2023.
- [5] Toronto Metropolitan University Pressbooks, *The Evolution of Web Accessibility*, <https://pressbooks.library.torontomu.ca/pwaa/chapter/the-evolution-of-web-accessibility/>
- [6] The History of the Web, *An Early History of Web Accessibility* – Jay Hoffmann, <https://thehistoryoftheweb.com/accessibility-tools/>
- [7] Narodne novine, [https://narodne-novine.nn.hr/clanci/sluzbeni/2019\\_02\\_17\\_358.html](https://narodne-novine.nn.hr/clanci/sluzbeni/2019_02_17_358.html)
- [8] WAI, *WCAG 2 Overview*, <https://www.w3.org/WAI/standards-guidelines/wcag/>
- [9] MDN, *Understanding of Web Content Accessibility Guidelines*, [https://developer.mozilla.org/en-US/docs/Web/Accessibility/Understanding\\_WCAG](https://developer.mozilla.org/en-US/docs/Web/Accessibility/Understanding_WCAG)
- [10] WAI, *WAI-ARIA Overview*, <https://www.w3.org/WAI/standards-guidelines/aria/>
- [11] WAI, *Using ARIA*, <https://www.w3.org/TR/using-aria/>
- [12] University of Washington, *How does accessible web design benefit all web users*, <https://www.washington.edu/doit/how-does-accessible-web-design-benefit-all-web-users>
- [13] Click-Away Pound, <https://www.clickawaypound.com/>
- [14] WebAIM, <https://webaim.org/articles/>
- [15] WAI, *Web Accessibility Evaluation Tools List*, <https://www.w3.org/WAI/ER/tools/>
- [16] WAI, *Selecting Web Accessibility Evaluation Tools*, <https://www.w3.org/WAI/test-evaluate/tools/selecting/>
- [17] Dubbot, *Dark Mode in Web Apps: AllY Pros and Cons* – Maggie Vaughan, <https://dubbot.com/dubblog/2023/dark-mode-in-apps.html>
- [18] MDN, *Keyboard*, [https://developer.mozilla.org/en-US/docs/Web/Accessibility/Understanding\\_WCAG/Keyboard](https://developer.mozilla.org/en-US/docs/Web/Accessibility/Understanding_WCAG/Keyboard)
- [19] WebAIM, *Keyboard accessibility*, <https://webaim.org/techniques/keyboard/>
- [20] W3C, *User Experiences and Benefits to Organizations*, <https://w3.org/WAI/media/av/users-orgs>
- [21] W3C, *Captions/Subtitles*, <https://w3.org/WAI/media/av/captions>

[22] W3C, *Animations*, <https://www.w3.org/WAI/tutorials/carousels/animations/>

## Tablica slika

Slika 1. Regije web stranice .....	21
Slika 2. Primjer pravilno strukturirane web stranice .....	23
Slika 3. Element bez WAI-ARIA atributa.....	24
Slika 4. Element <div> u ulozi botuna .....	24
Slika 5. Korištenje WAI-ARIA kod toggle botuna .....	25
Slika 6. Primjer korištenja "aria-labelledby" .....	26
Slika 7. Primjer korištenja "aria-label" .....	26
Slika 8. React hook i funkcija za promjenu teme .....	28
Slika 9. Primjena funkcije za promjenu teme.....	28
Slika 10. CSS klase i stilovi .....	28
Slika 11. Primjena CSS klase .....	29
Slika 12. Svijetli način rada.....	29
Slika 13. Tamni način rada.....	29
Slika 14. Izgled botuna u tamnom načinu rada .....	30
Slika 15. Usporedba "serif" i "sans serif" fonta.....	32
Slika 16. Primjena specifičnog fonta za ljude s poteškoćama.....	32
Slika 17. Predviđanje teksta na sustavu iOS.....	35
Slika 18. Primjer korištenja atributa "alt" .....	37
Slika 19. Primjer dodavanja tekstualnog opisa za grafikon.....	37
Slika 20. Izgled grafikona i alternativnog teksta u aplikaciji .....	38
Slika 21. Primjer implementacije čitača zaslona .....	39
Slika 22. Čitač zaslona u pregledniku Microsoft Edge .....	39

# Privitak

## Instalacija programske podrške

Aplikacija dostupna za pruzimanje na <https://github.com/tmildardovich/costPlanning>

Nakon preuzimanja naredbom *npm install* instalirati potrebne pakete projekta.

Nakon instalacije, backend dio se pokrenuti naredbom *npm run dev*, a frontend dio naredbom *npm start*.