**SVEUČILIŠTE U SPLITU**

**FAKULTET ELEKTROTEHNIKE, STROJARSTVA**

**I BRODOGRADNJE**

**Nail Studio Flex**

Marija Bikić i Domina Amižić

**Kolegij:**

Korisnička sučelja

**Mentor:**

Mario Čagalj

Split, kolovoz 2020.

Sadržaj

[Uvod 4](#_Toc49197794)

[Očekivani tip korisnika 4](#_Toc49197795)

[Ograničenja sustava 4](#_Toc49197796)

[Zahtjevi sustava 4](#_Toc49197797)

[Low-Fidelity grafički prototip 5](#_Toc49197798)

[High-fidelity prototip 7](#_Toc49197799)

[Evaluacija prototipa 8](#_Toc49197800)

[Heuristike iskoristivosti 8](#_Toc49197801)

[CRAP principi 15](#_Toc49197802)

[Implementacija prototipa 18](#_Toc49197803)

[PageSpeed Insights 21](#_Toc49197804)

[Zaključak 22](#_Toc49197805)

# Uvod

Nail Studio Flex je stranica pomoću koje se može rezervirati termin za nokte, ovisno o potrebi pedikura ili manikura. Na stranici se također mogu vidjeti osnovne informacije o salonu kao što su adresa, kontakt broj, email i lokacija te mogućnost uvida u cjenik i uslugu se nalazi u ponudi studia Flex.

Cilj ove stranice je omogućiti klijentima brzo i lako rezervirati svoj termin za nokte. Na taj način se izbjegne direktno zvanje na telefon salona, čekanje da se netko javi i sam taj razgovor dok se provjeri kada ima slobodan termin koji bi odgovarao klijentu. Ovako se na vrlo jednostavan način klikne na rezerviraj, odabere željeni slobodni datum i vrijeme.

## Očekivani tip korisnika

Ova stranica namijenjena je za osobe kojima je potrebna rezervacija termina za usluge manikure ili pedikure. Korisnik treba poznavati hrvatski jezik. Pretpostavlja da će prevladavati ženska populacija u dobi od 15 do 50 godina.

## Ograničenja sustava

Za korištenje ovog sustava korisniku je potreban pristup internetu.

## Zahtjevi sustava

Zahtjevi i zadaće:

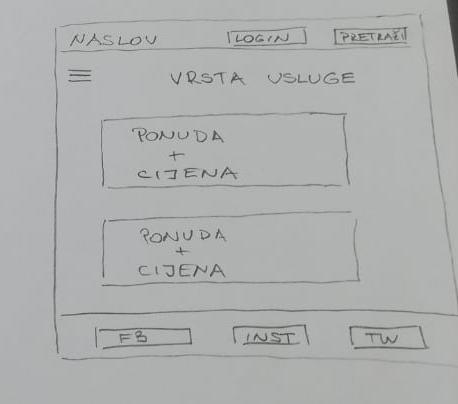
* Aplikacija će se koristiti iz web preglednika
* Aplikacija će se koristiti na uređajima različite veličine (smartphone, tableti, laptopi i desktop računala)
* Korisnik može pretraživati/filtrirati produkte ili usluge
* Broj produkata/usluga koje se nude putem web aplikacije veći je od 20
* Korisnik će koristiti aplikaciju kao gosti ili će se logirati u vlastiti profil
* Broj javnih stranica (one stranice koje ne zahtjevaju logiranje u sustav) je 5
* Jedna od javnih stranica je blog (blog mora sadržavati minimalno 20 postova koji mogu uključivati slike, videa, code snippet-e)

# Low-Fidelity grafički prototip

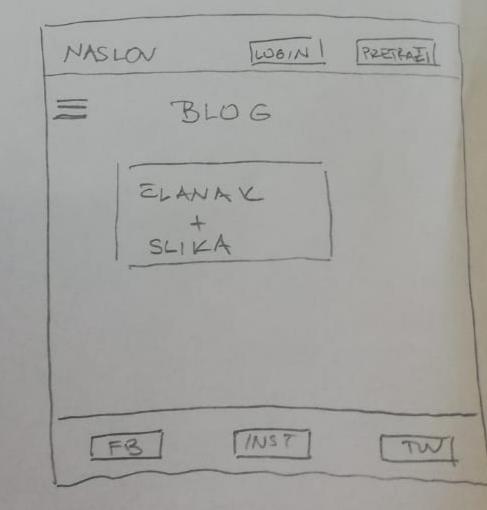
Prvi korak pri izradi projekta je napraviti low-fidelity prototipa. Cilj low-fidelity prototipa je predočiti pojednostavljeni koncept dizajna u svrhu početnog testiranja.



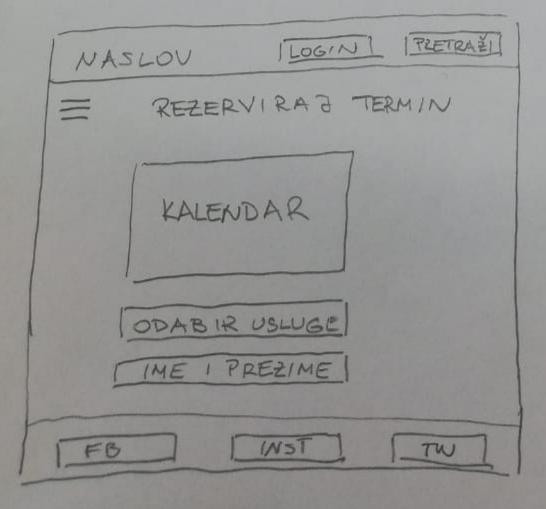
*Slika 1 - Low-fidelity prototip za naslovnu stranicu*



*Slika 2 - Low-fidelity prototip za desktop verziju za usluge*



*Slika 3 - Low-fidelity prototip za desktop verziju za blog*



*Slika 4 - Low-fidelity prototip za desktop verziju za rezervaciju*

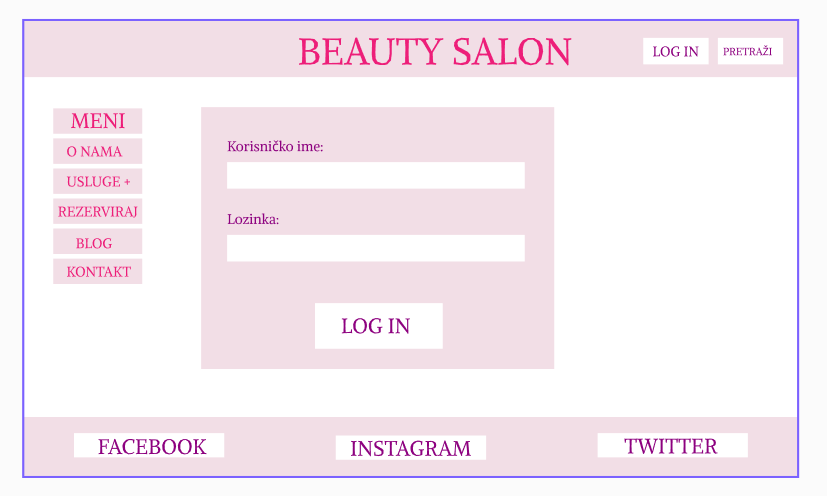
# High-fidelity prototip

Nakon izrade low-fidelity prototipa, drugi korak je izrada high-fidelity prototipa. Prototip je izrađen u programu Figma.

High-fidelity prototip za desktop:



*Slika 5 - High-fidelity prototip desktop verzija naslovne stranice*



*Slika 6 - High-fidelity prototip desktop verzija za prijavu*

# Evaluacija prototipa

Na početku semestra imali smo zadatak izraditi low-fidelity i high-fidelity prototip. Tijekom trajanja semestra naučili smo nove principe o pravilnom načinu izrade web stranica te se zbog toga krajnja verzija stranice razlikuje od high-fidelity prototipa. Na temelju usvojenog znanja o principima i heuristici podignuli smo stranicu na veću razinu.

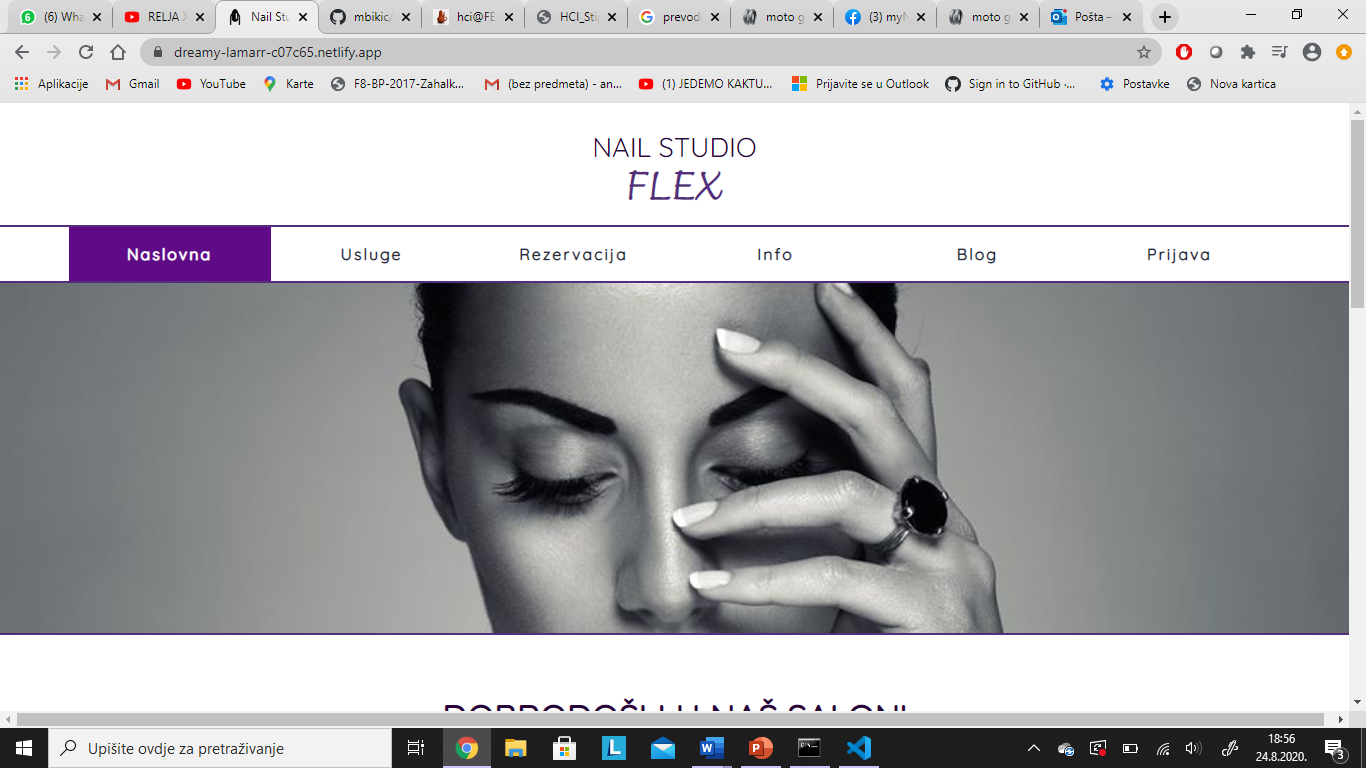
# Heuristike iskoristivosti

Jacob Nielsen definirao je deset heuristika iskoristivosti za korisnička sučelja. Pomoću usvojenih heuristika u ovom ćemo poglavlju opisati na koji način su one implementirane u našem projektu.

**Podudaranost između stvarnog svijeta i sustava**

Stranica mora biti dizajnirana jezikom koji je poznat korisnicima te izbjegavati stručnu terminologiju nepoznatu prosječnom korisniku. Ova stranica namijenjena je korisnicima koji pričaju hrvatski jezik te je sve pisano na istom.

Primjer ove heurističke metode može se prepoznati već u zaglavlju stranice u kojem korisnik može odabrati koju akciju želi izvršiti.

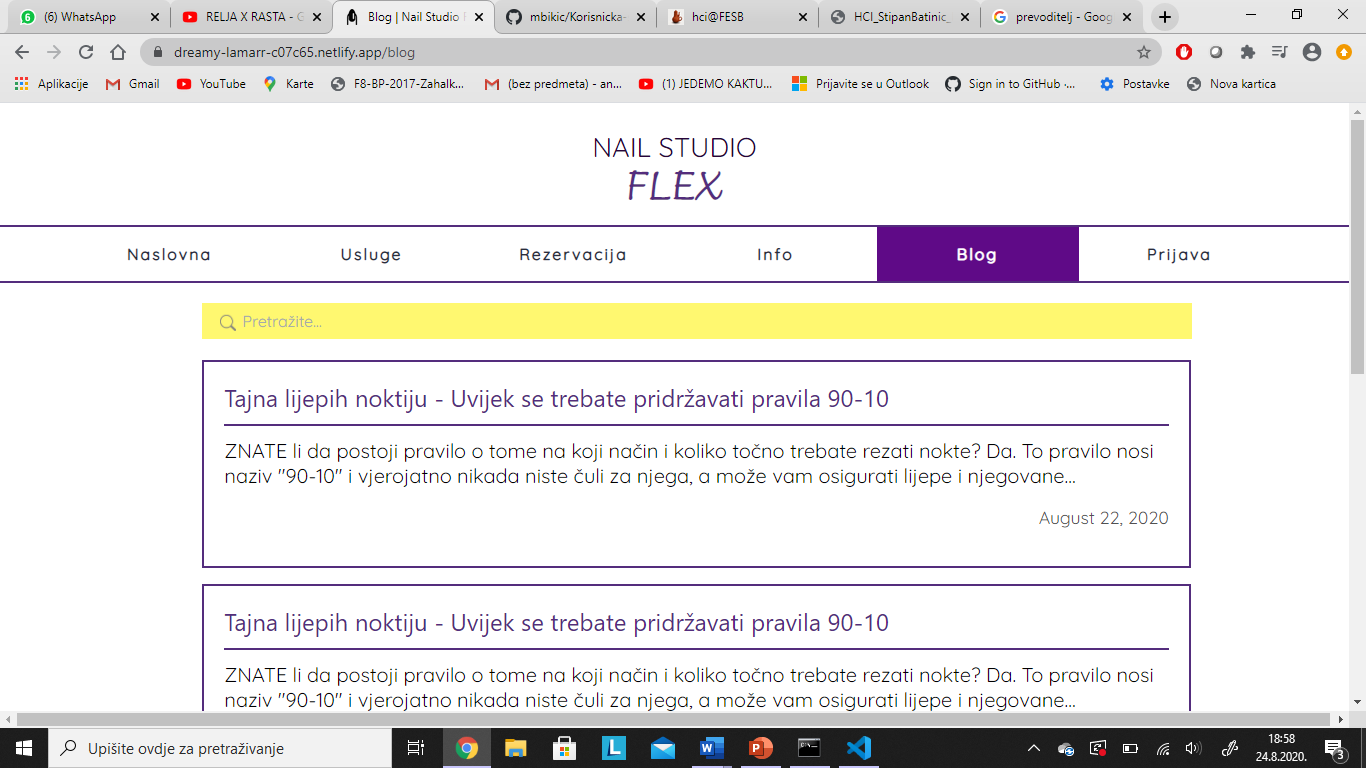


*Slika 7 - Podudaranost između stvarnog svijeta i sustava*

**Konzistentnost i standardi**

Korisnici ne bi trebali razmišljati znače li pojedine akcije, riječi ili situacije isto. Slične stvari bi u pravilu trebale funkcionirati i izgledati slično, a različite stvari različito.

Primjer ove heurističke metode: Koristili smo standarde specifične za prikaz podataka na webu kao što su navigacijska traka na vrhu stranice, footer na dnu stranice, tražilicu itd. Također smo koristili normativne simbole koji su korisnicima prepoznatljivi za određene akcije kao što je znak povećala za pretraživanje.

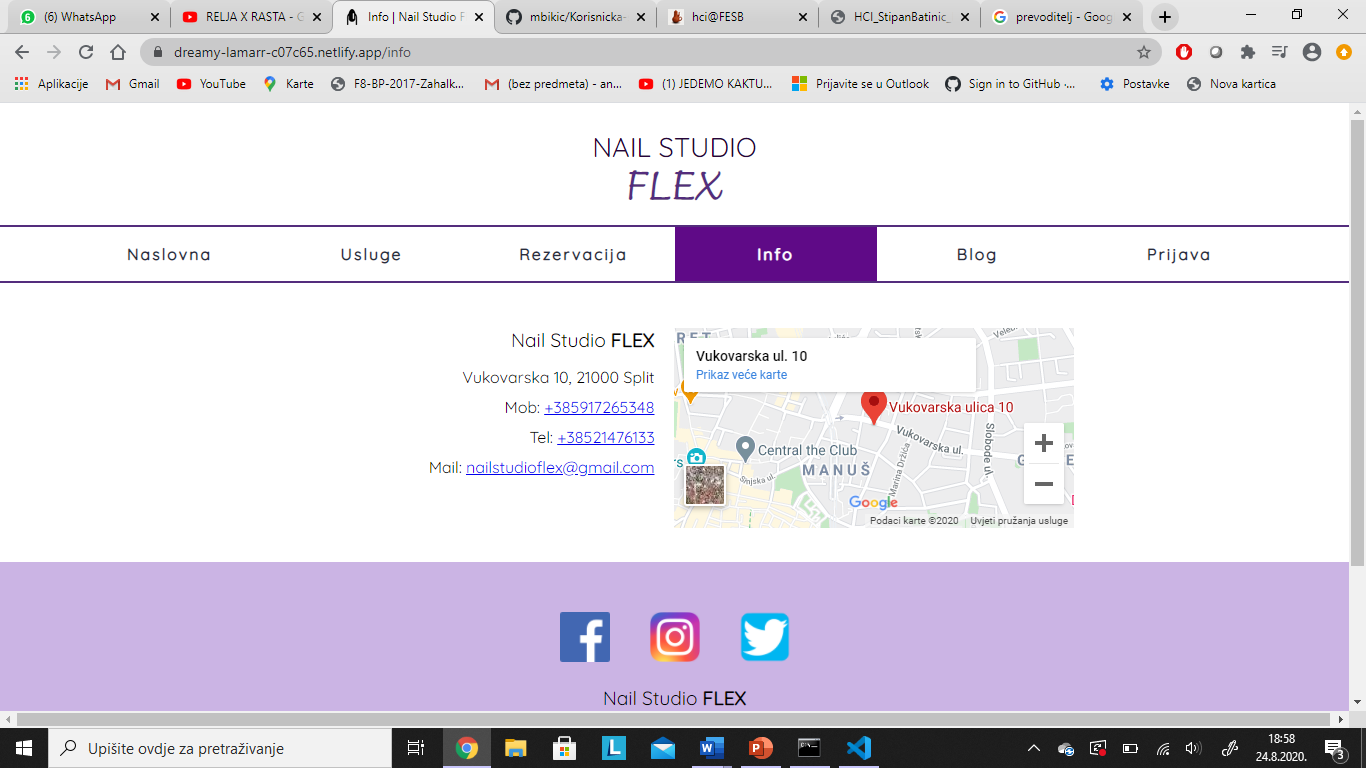


*Slika 8 - Konzistentnost i standardi*

**Vidljivost statusa sustava**

Sustav bi morao u svakom trenutku obavještavati korisnika o tome što obavlja u datom trenutku kroz određeni feedback.

Na primjer, na slici korisnik točno zna gdje je kliknuo zato što kad s mišem dođeš do želje stranice (Naslovna, Usluge, Rezervacija, Info, Blog ili Prijava) mali pravokutnik bude ljubičaste boje. To je upravo taj feedback koji korisniku vraća povratnu informaciju odnosno obavještava ga gdje se nalazi u datom trenutku.

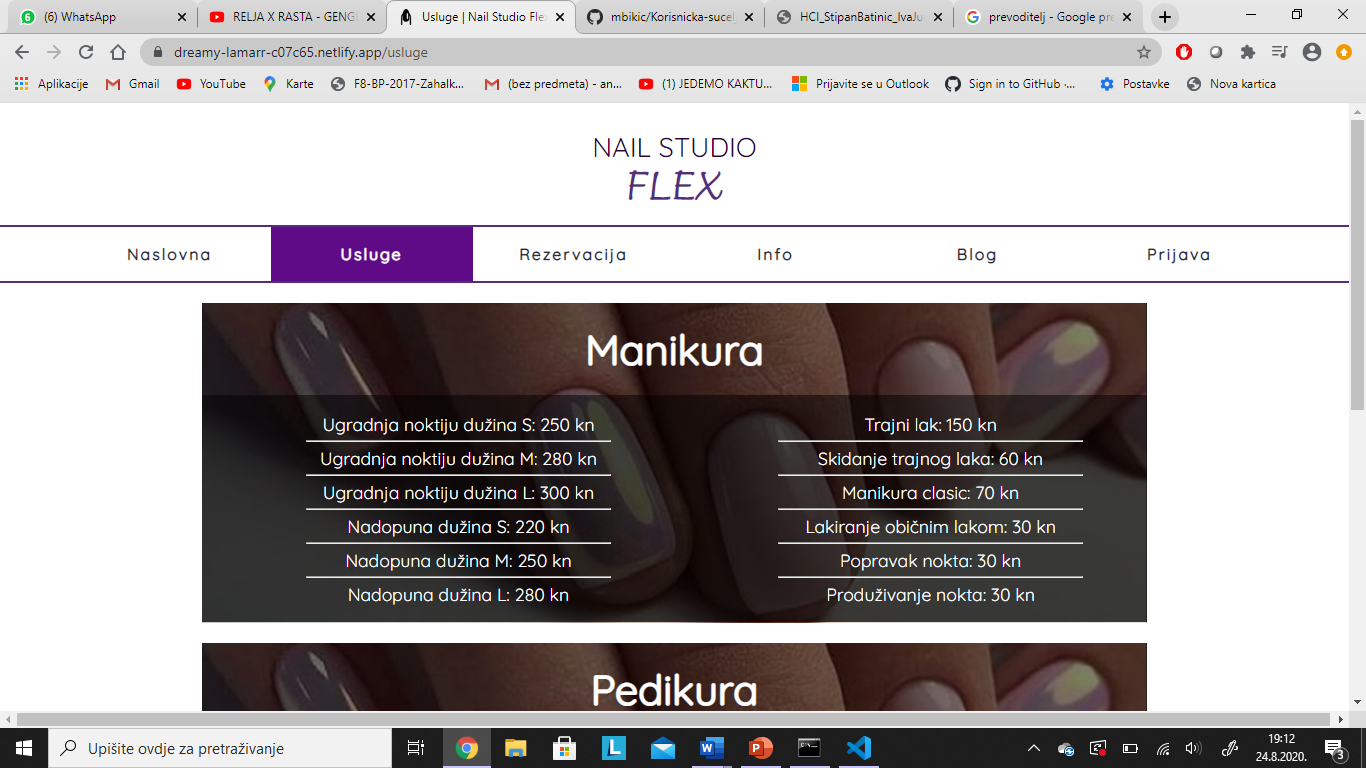


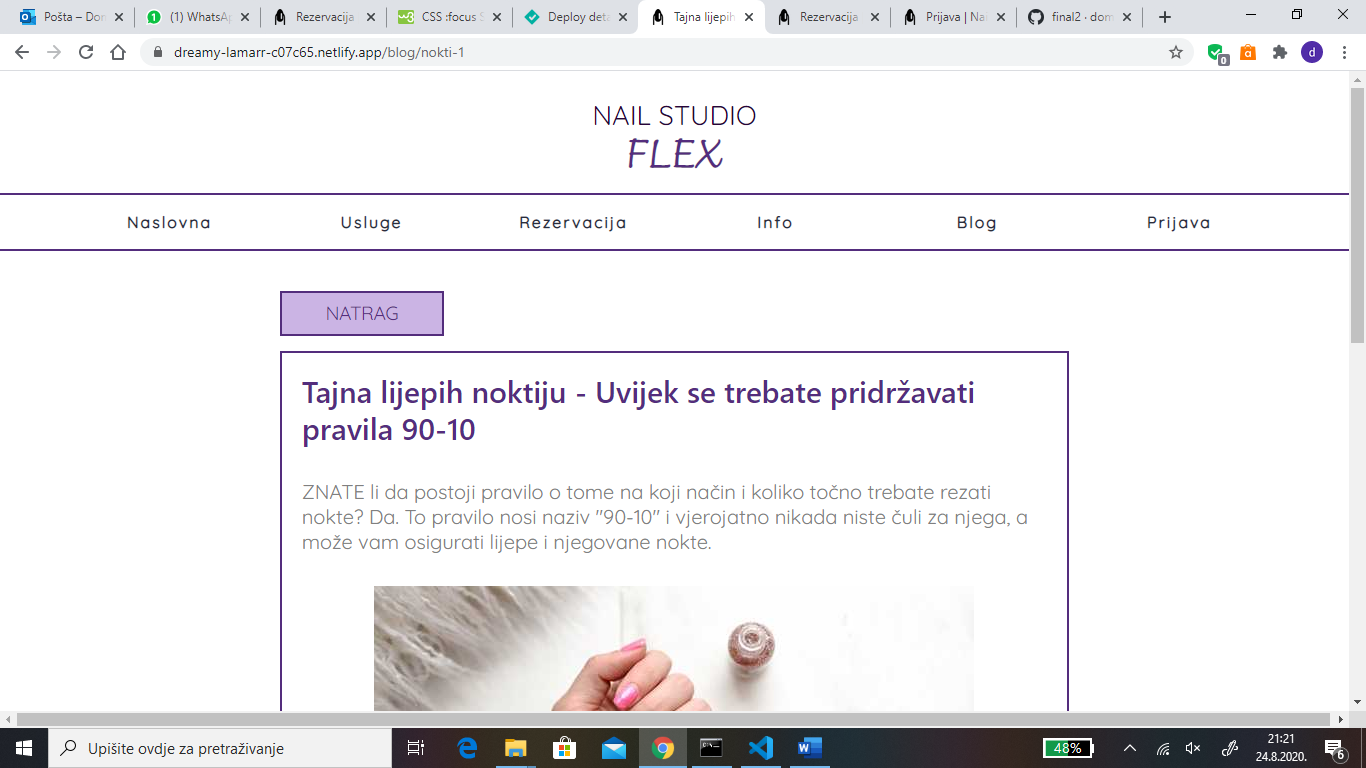
*Slika 9 - Vidljivost statusa sustava*

**Sloboda korisnika i korisnička kontrola**

Korisnicima se često dogodi da slučajno odaberu opciju koju nisu htjeli pa im je potrebno omogućiti „izlaz u slučaju nužde“ pomoću kojeg mogu napustiti neželjene opcije i vratiti na prethodno stanje stranice na jednostavan način.

Primjer ove heuristike je da korisnik ima mogućnost odabira druge stranice u svakom trenutku jer je zaglavlje uvijek vidljivo i dostupno korisniku, kao i dugme Natrag kod bloga.



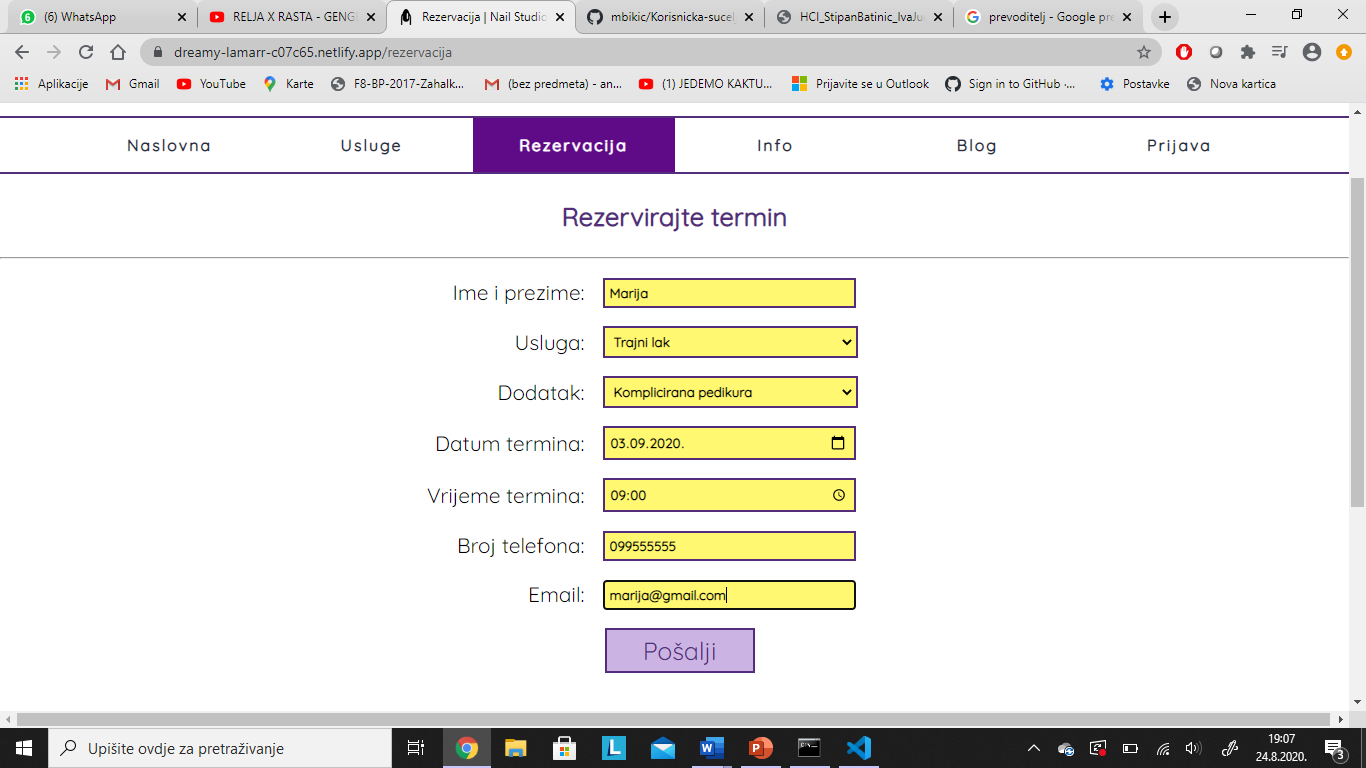


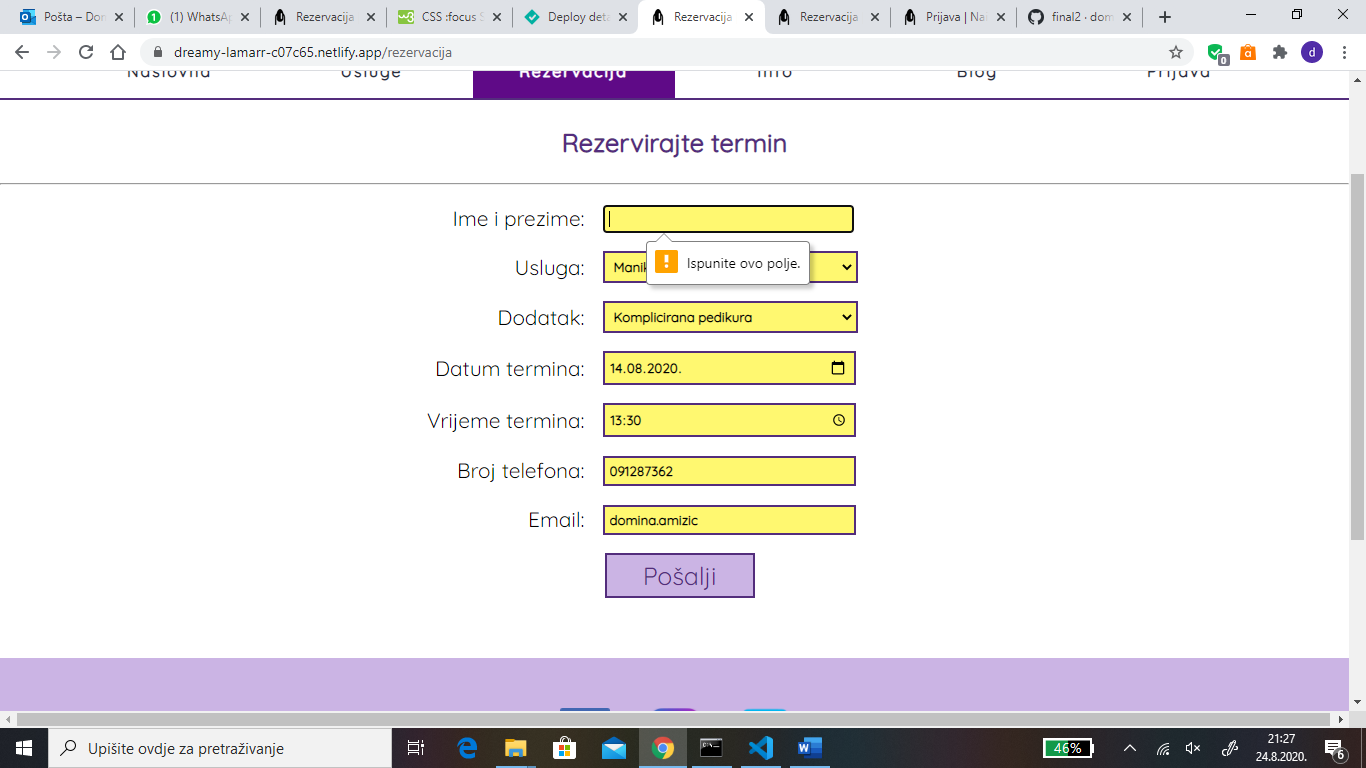
*Slika 10 - Sloboda korisnika i korisnička kontrola*

**Sprječavanje grešaka**

Sprječavanje grešaka bolje je od bilo kakve poruke o grešci. Sustav je potrebno dizajnirati na način da se eliminiraju moguće greške ili da se korisnika upozori o mogućim greškama.

Primjer ove heuristike vidljiv je na stranici za rezervaciju gdje korisnik nakon što odabere željenu odluku i unese podatke ima mogućnost da još jedan put provjeri da li su svi podatci ispravni.



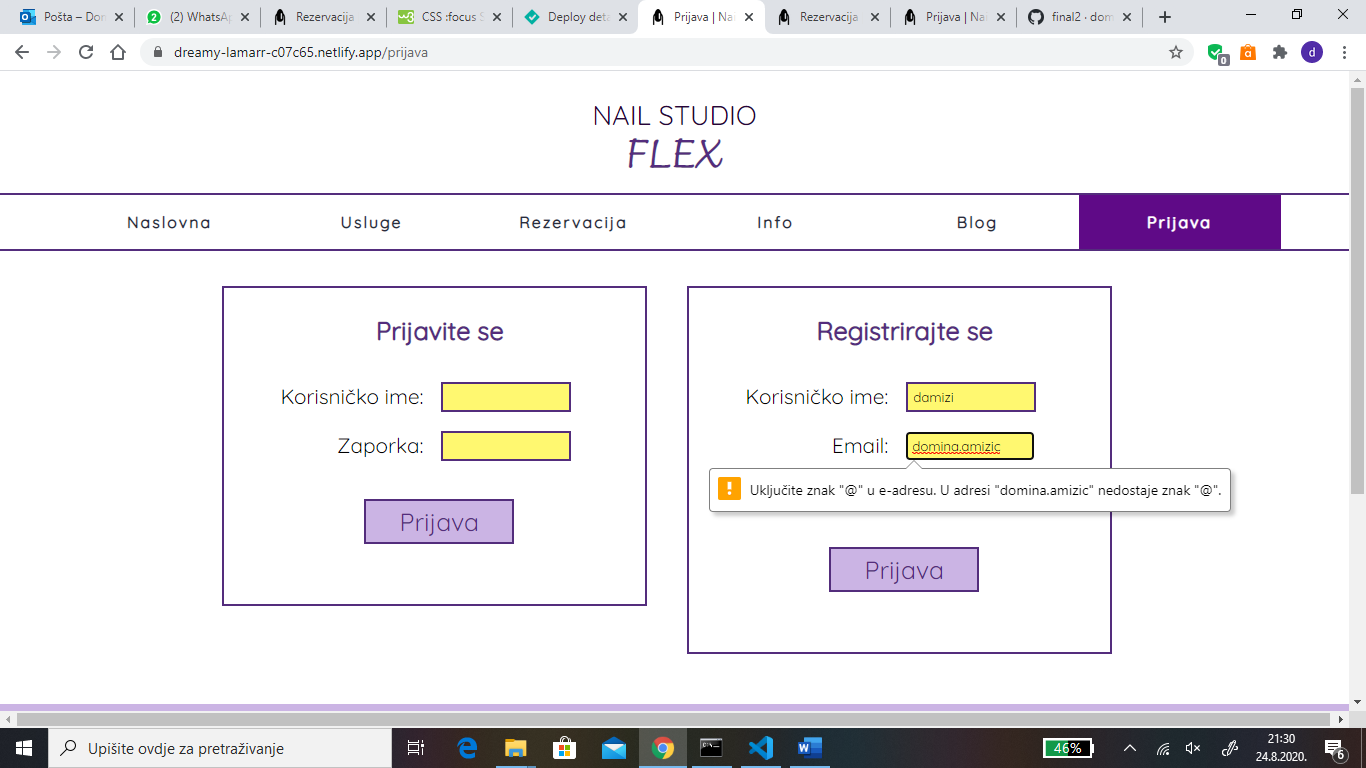


*Slika 11 - Sprječavanje grešaka*

**Detekcija i oporavak od pogreške**

Obavijesti o greškama moraju biti napisan razumljivim jezikom (bez kodova), precizno prikazati problem i uključivati potencijalni način rješavanja problema.

Primjer ove heuristike vidljiv je kod stranice za prijavu i registraciju. Prilikom prijave, ako korisnik unese neispravnu e-mail adresu, dobit će feedback da je email adresa pogrešno napisana.

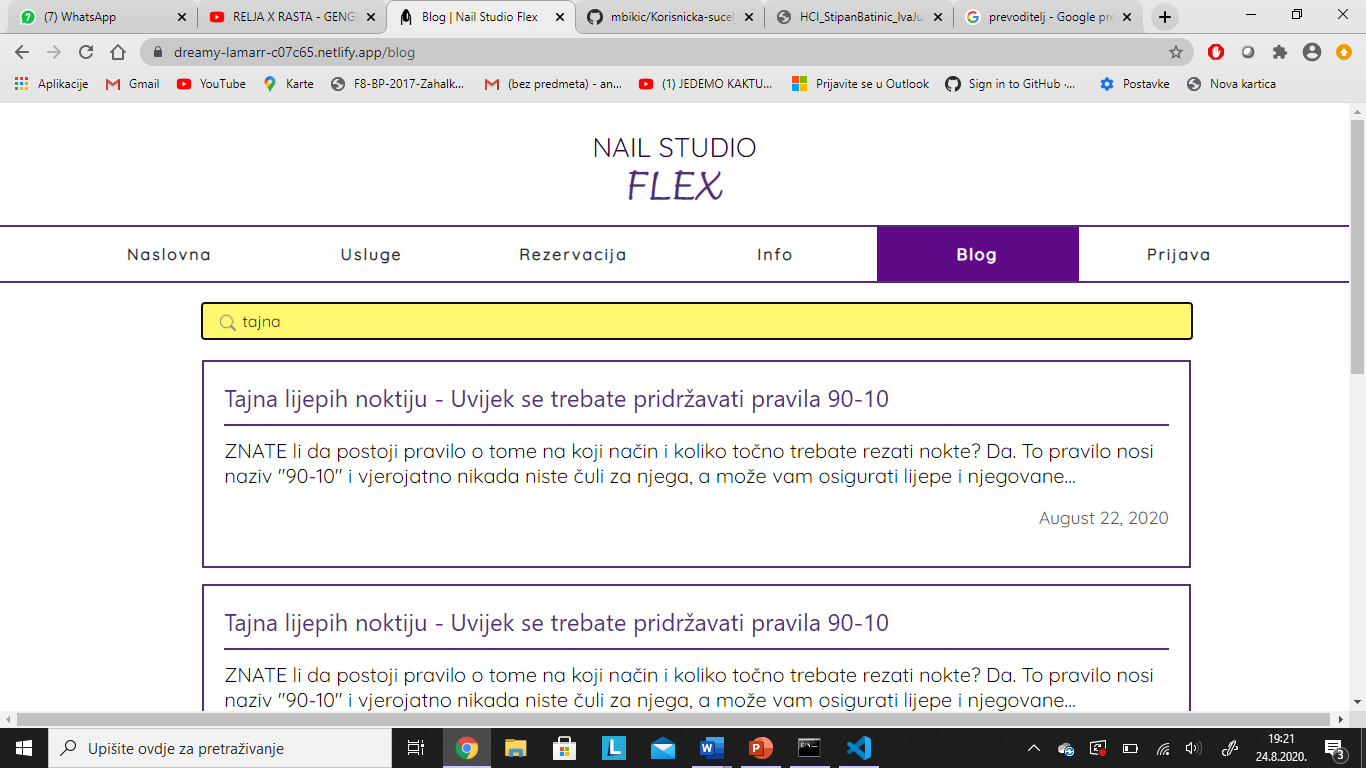


*Slika 12 - Detekcija i oporavak od pogreške*

**Prepoznavanje umjesto prisjećanje**

Sustav treba minimizirati količinu informacija i opcija koje korisnik mora zapamtiti. Korisnik ne smije biti primoran pamtiti informacije. Upute za korištenje sustava trebaju biti vidljive uvijek kada su potrebne.

Primjer navedene heuristike vidljiv je kod pretraživanja stranice bloga. Unosom riječi ili dijela riječi, korisnik može doći do željenog članka.



*Slika 13 - Prepoznavanje umjesto prisjećanje*

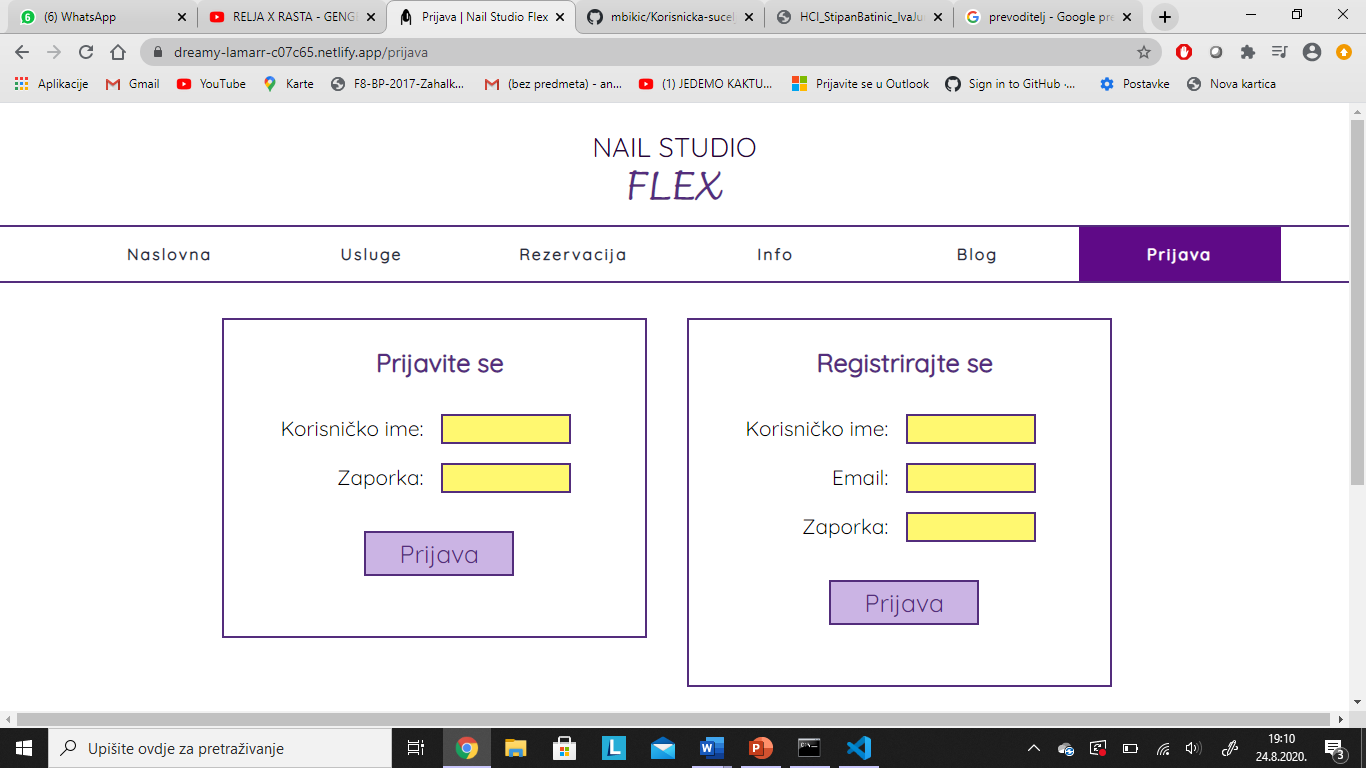
**Fleksibilnost i efikasnost korištenja**

Akceleratori mogu ubrzati interakcije iskusnih korisnika sa sustavom te na taj način sustav može odgovarati i novim i iskusnim korisnicima. Zbog jednostavnosti aplikacije, ova stranica nema implementirane akceleratore.

**Estetika i minimalistički dizajn**

Prilikom dizajniranja stranice pokušavamo izbjeći uporabu bespotrebnog sadržaja. Bilo koja neželjena informacija natječe se s relevantnim podacima i umanjuje njihovu vidljivost. Stranica u pravilu ne bi smjela sadržavati informacije koje se ne koriste i nisu od važnosti. Teži se minimalizmu i jednostavnošću odnosno „manje je više“.

Na slici se jasno vidi primjer minimalističkog dizajna.



*Slika 14 - Estetika i minimalistički dizajn*

**Pomoć i dokumentacija**

Premda bi bilo bolje da je sustav dizajniran na način da ga je moguće koristiti bez dodatne dokumentacije, u nekim slučajevima nemoguće ju je izbjeći. Pomoć bi trebala biti pristupačna i jasno interpretirati funkcije koje su nužne korisniku. Pomoć nije zamjena za loš dizajn!

Zbog jednostavnosti aplikacije pomoć nije implementirana.

# CRAP principi

CRAP – Contrast, Repetition, Aligment, Proximity - skup elementarnih principa koji se koriste u svrhu postizanja što prikladnijeg vizualnog dizajna.

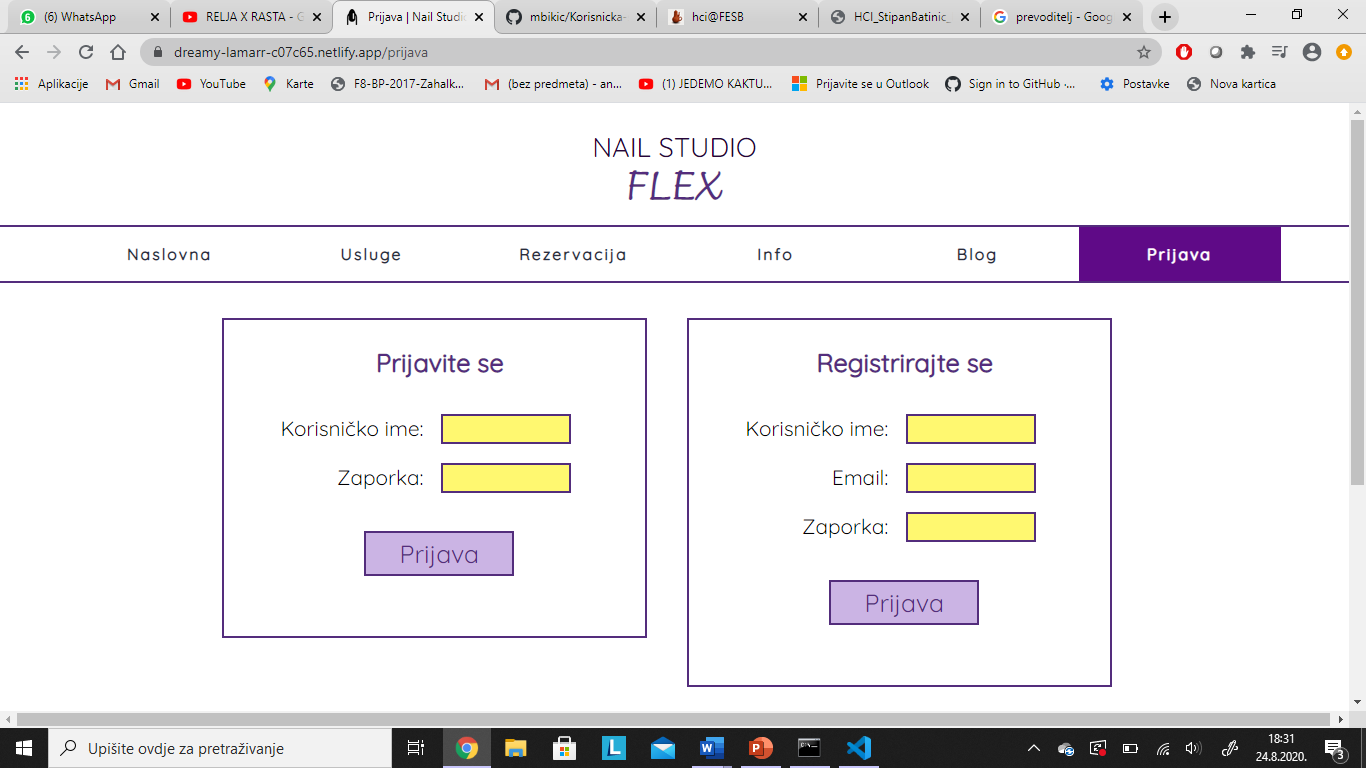
Kontrast (Contrast)

Princip kontrasta:

- Vizualni elementi koji su različiti trebali bi izgledati različito

- Važni elementi moraju biti naglašeni kako bi se istaknuli u odnosu na nevažne (povećati ih i pojačati boju)

Primjer kontrasta vidi se kod jasno kod odabira boje žute i ljubičaste, prozorčić gdje treba upisati podatke obojan je žutom bojom, koja se razlikuje od prevladavajuće ljubičaste.

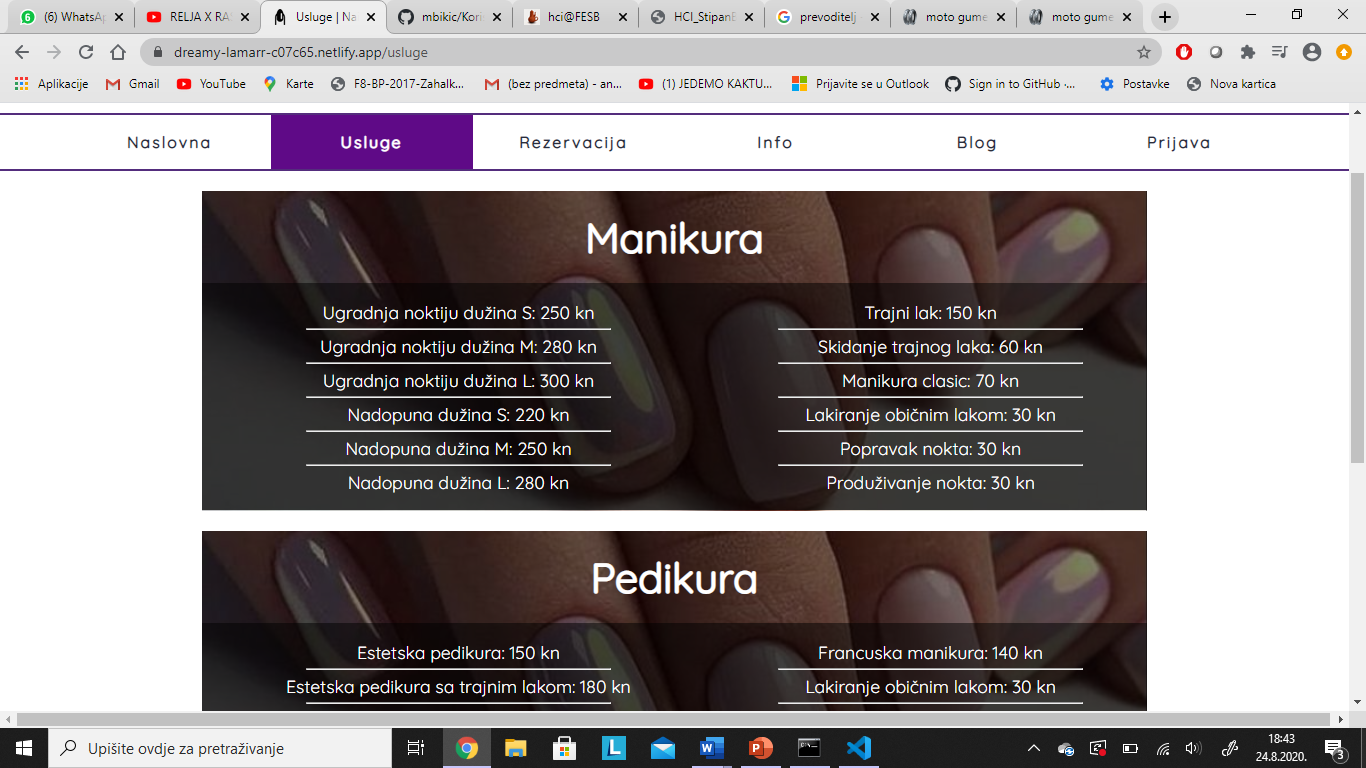


*Slika 14 - Contrast*

**Ponavljanje (Repetition)**

Vizualni elementi koji su od iste važnosti ili imaju jednaku razinu moraju izgledati isto. Naslovi, podnaslovi i font treba biti jednak na svim stranicama. Korištenje istog dizajna na stranici bitno je zbog održavanja konzistentnosti i jedinstva.

Primjer ovog principa na stranici vidljiv je tako što su svi naslovi, podnaslovi i font na svakoj stranici jednak.



*Slika 15 - Repetition*

**Poravnanje (Alignment)**

Princip poravnanja navodi da se niti nijedan element ne smije staviti na stranicu proizvoljno. Svaka stavka bi trebala imati vizualni vezu s nečim drugim na stranici. Ravne linije pružaju organiziran izgled te njihovo poravnanje označuje koheziju i pomaže konzistentnosti stranice. Iako na ekranu ne postoje linije koje povezuju elemente ljudsko oko i mozak ih stvaraju.

Primjer ovog principa se jasno vidi na stranici za usluge jer je svaka stavka usluge podebljana i jednako udaljena jedna od druge, kao i kod bloga.



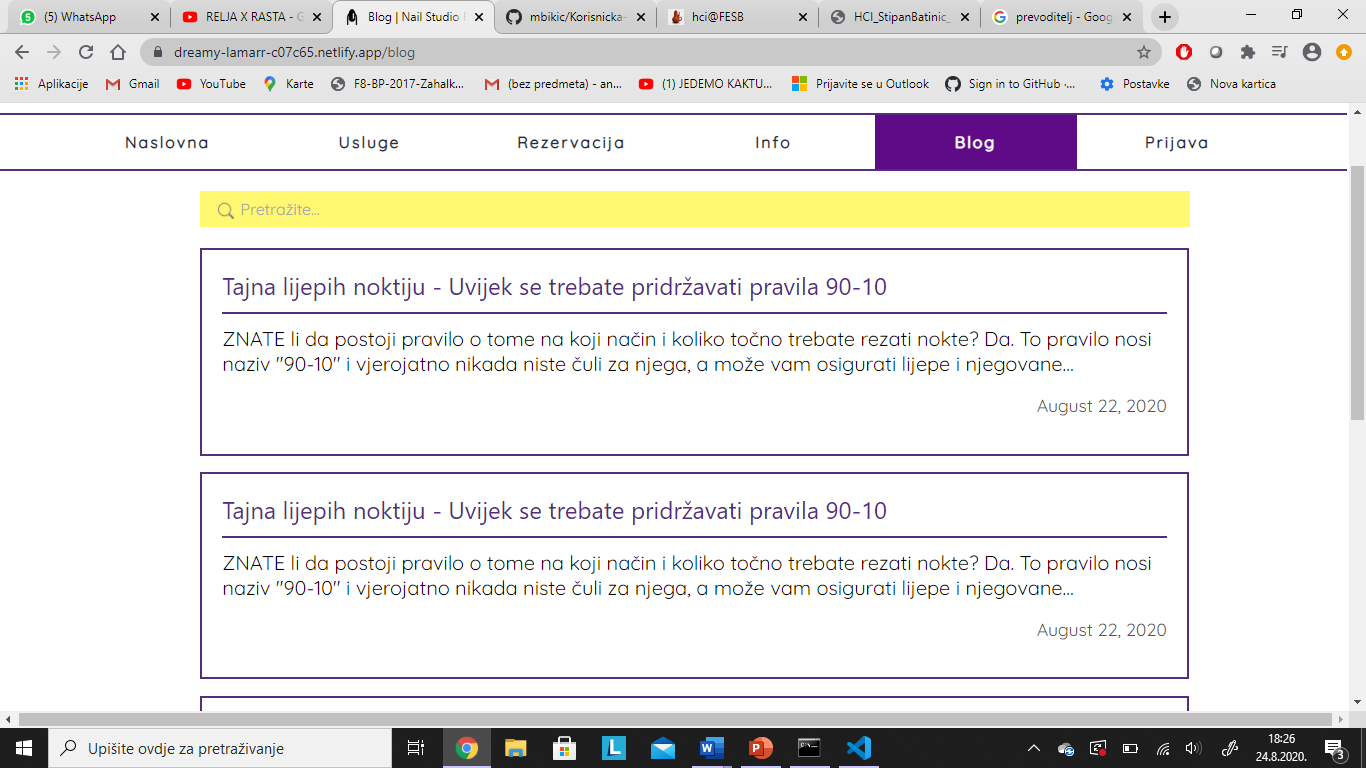


*Slika 16 - Aligment*

**Bliskost (Proximity)**

Princip bliskosti komponenti na stranici kaže nam da je povezane elemente važno staviti jedne kraj drugih kako korisnik ne bi doveo u pitanje koji elemente pripada kojoj cjelini. Također, elemente stranice koji nisu povezani treba razdvojiti. Princip bliskosti elemenata pomaže u organiziranju informacija, smanjuje nered i korisniku daje jasnu strukturu.

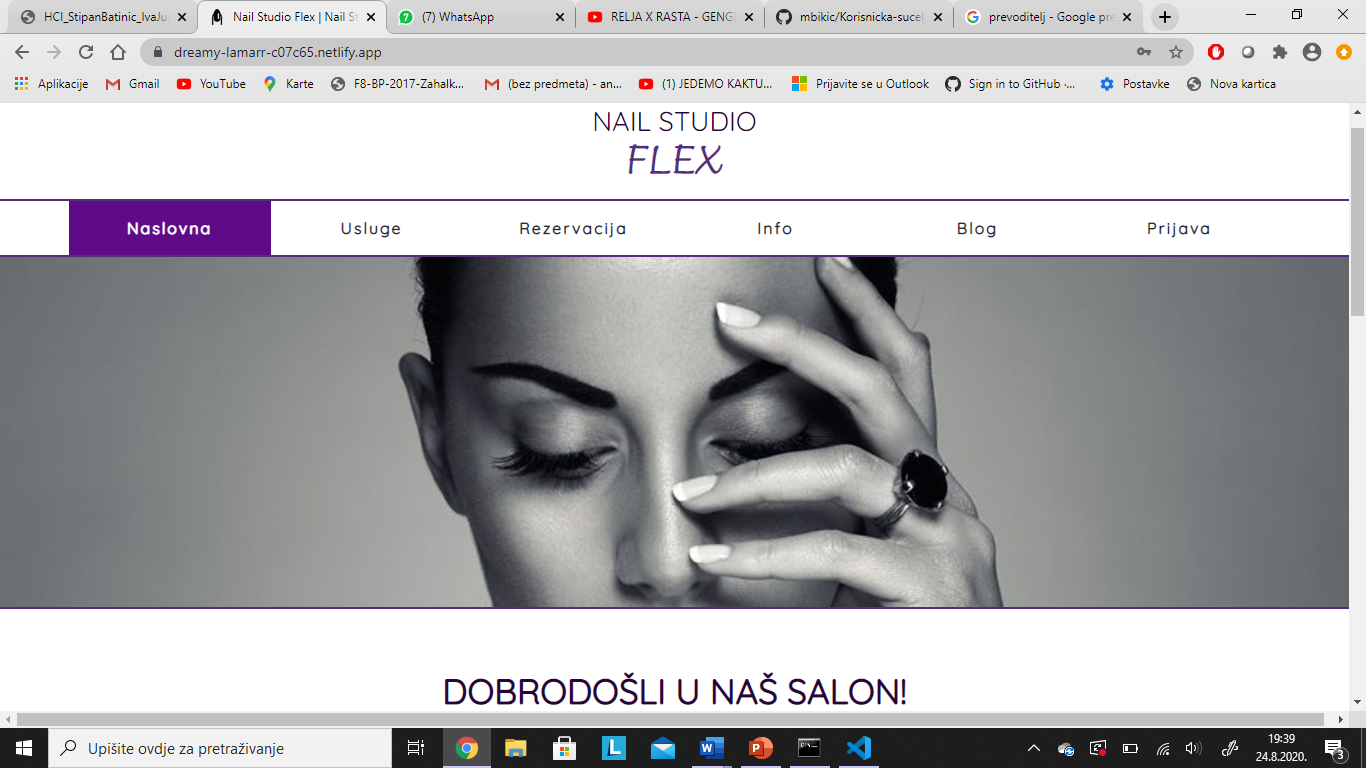
Primjer ovog principa je vidljiv na stranici za blog, jasno se vidi koji podnaslov pripada kojem članku.



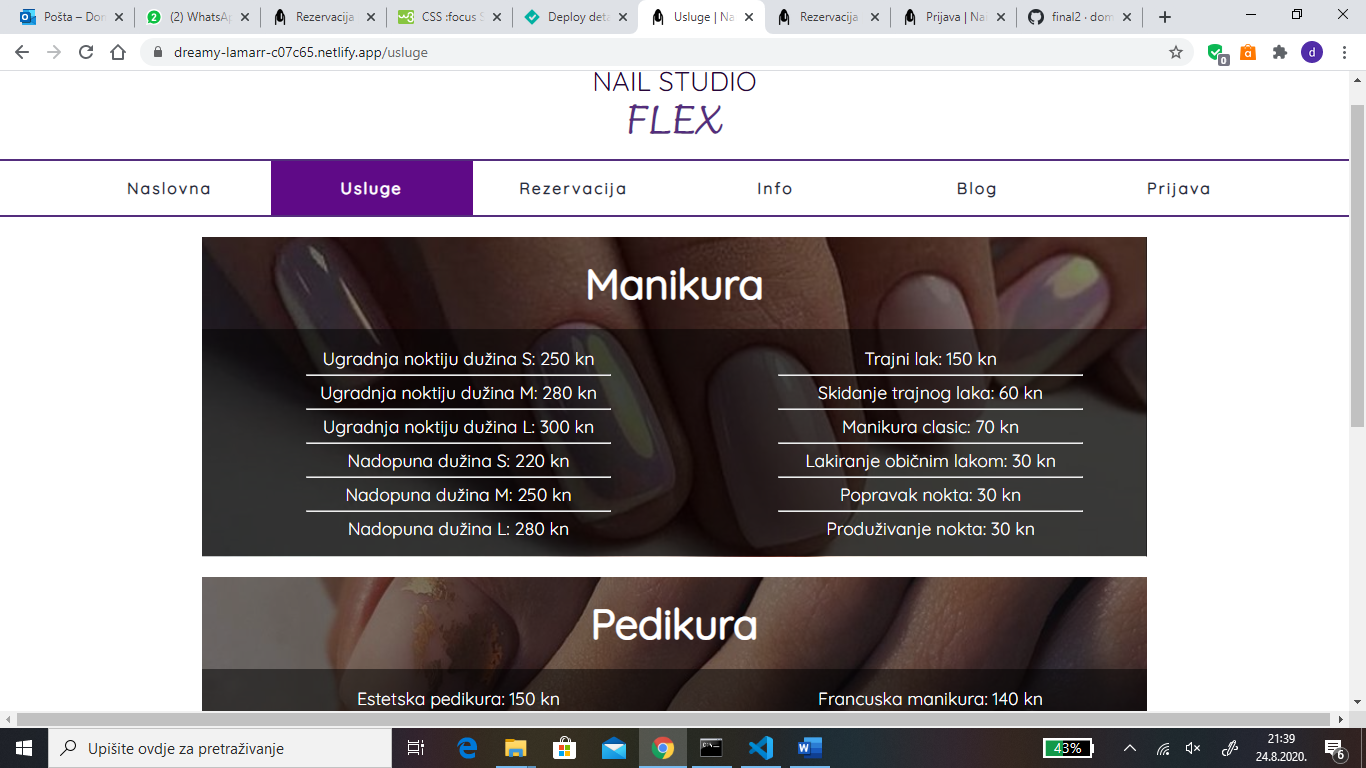
*Slika 17 - Proximity*

# Implementacija prototipa

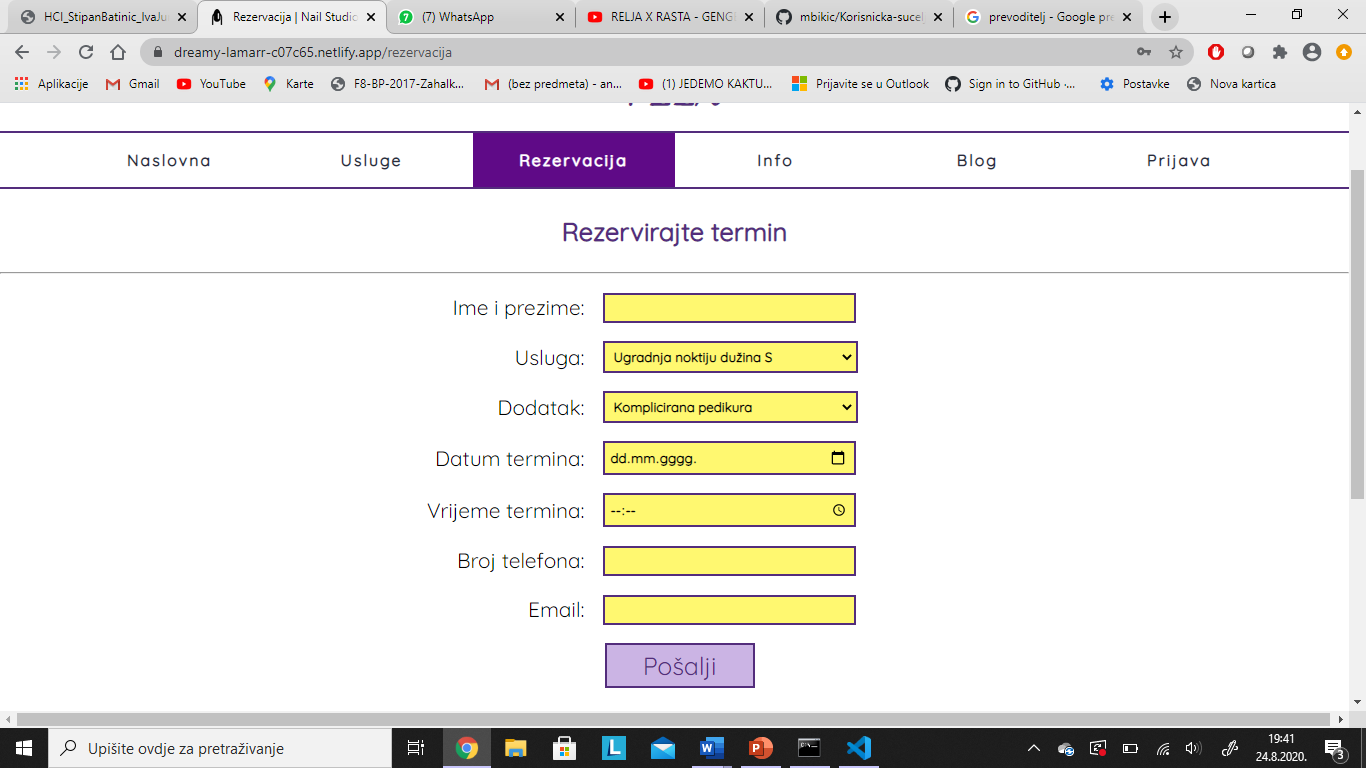
Nakon što smo primijenili određene heuristike, CRAP principe i ostale potrebne promjene slijedi faza implementacije prototipa. Stranica Nail Studio Flex napravljena je uz pomoć Gatsby-a i ReactJS-a.



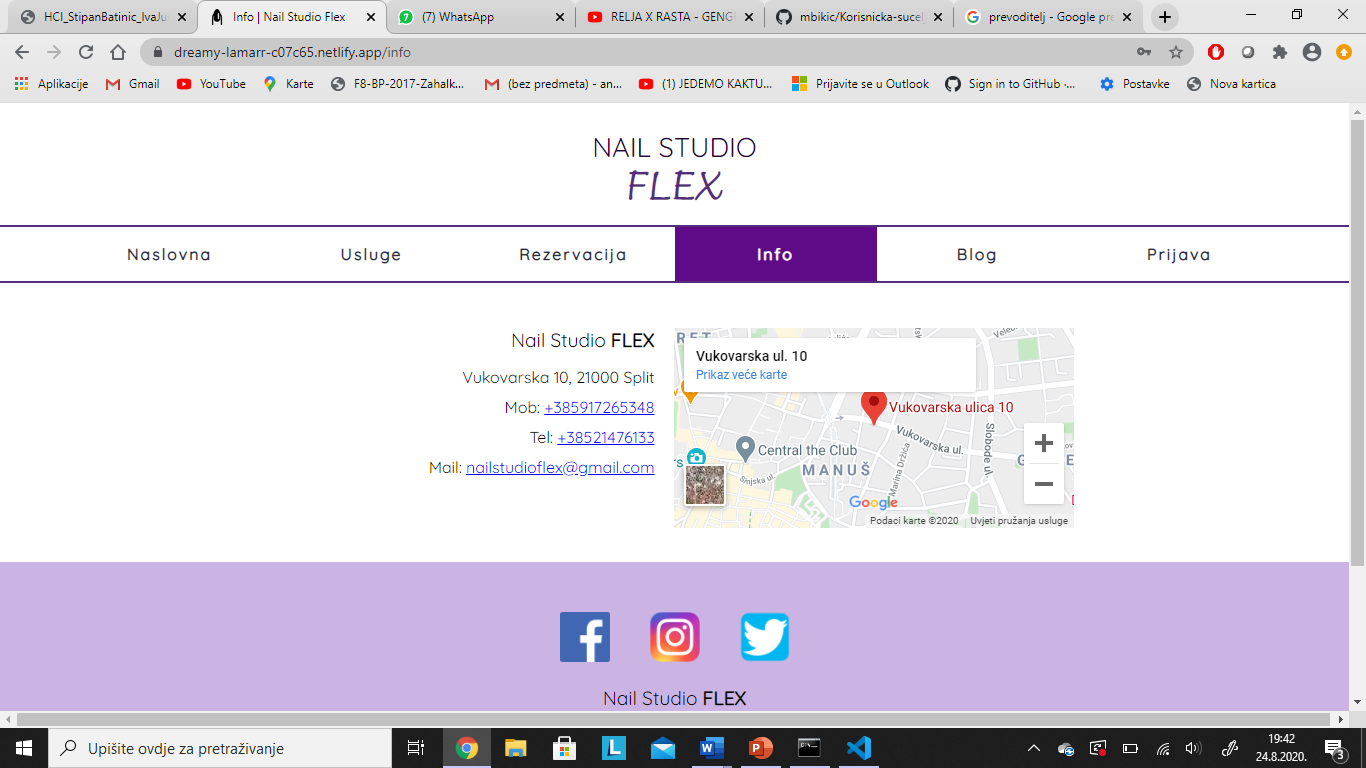
*Slika 18 - Naslovna*



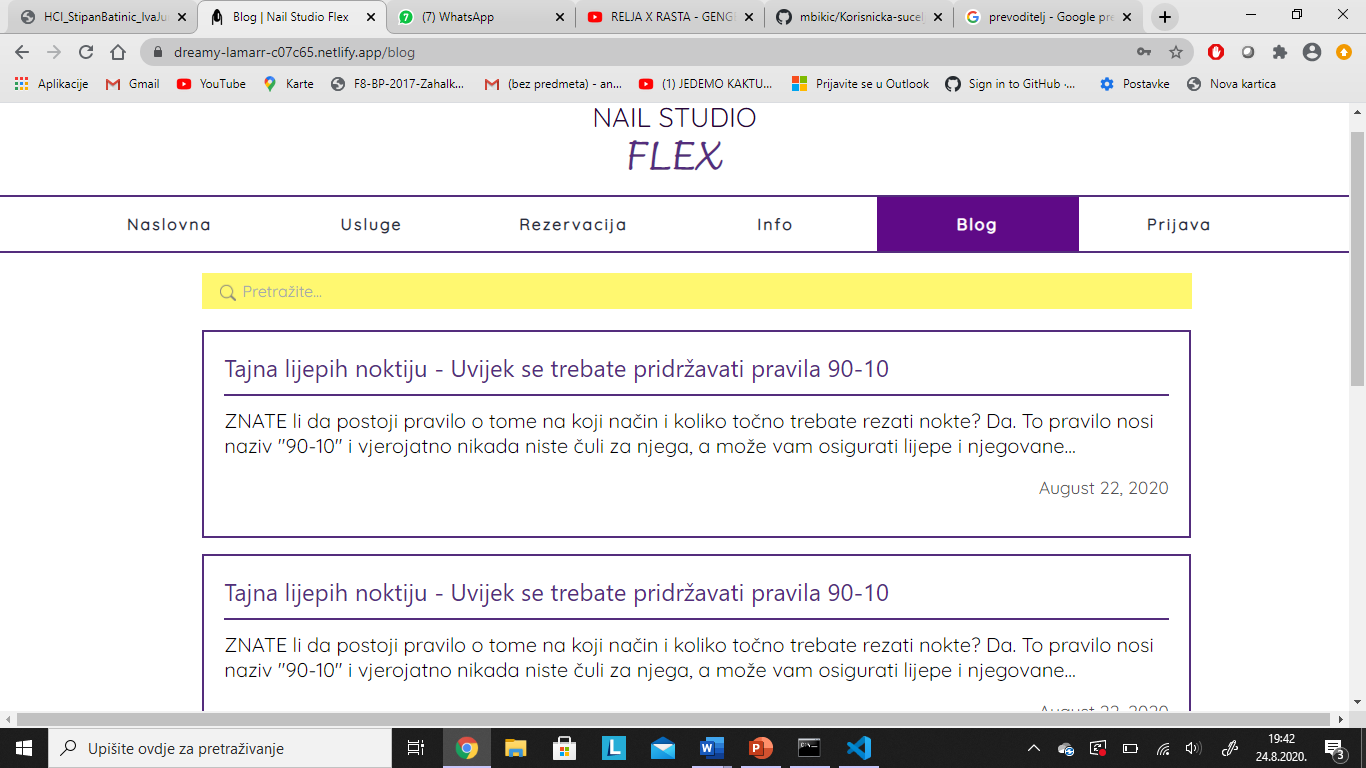
*Slika 19 - Usluge*



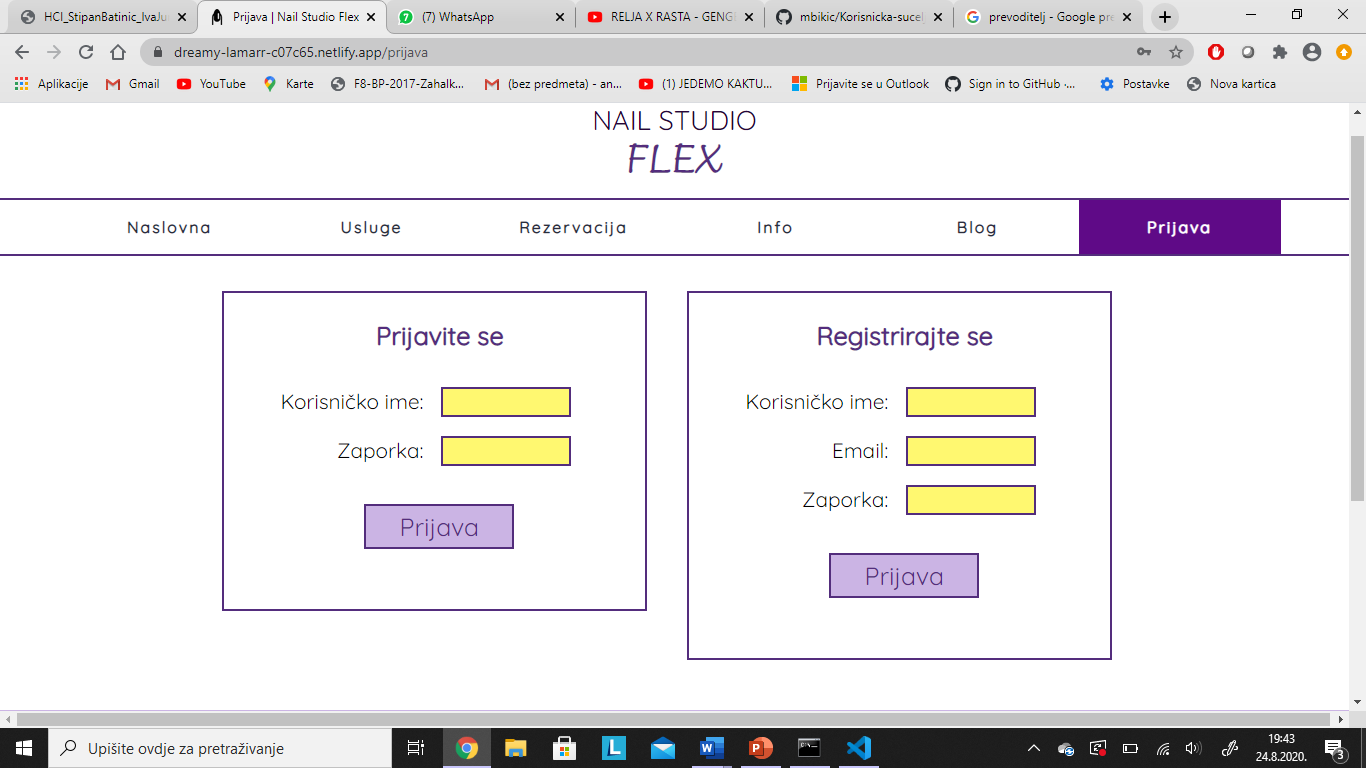
*Slika 20 - Rezervacija*



*Slika 21 - Info*



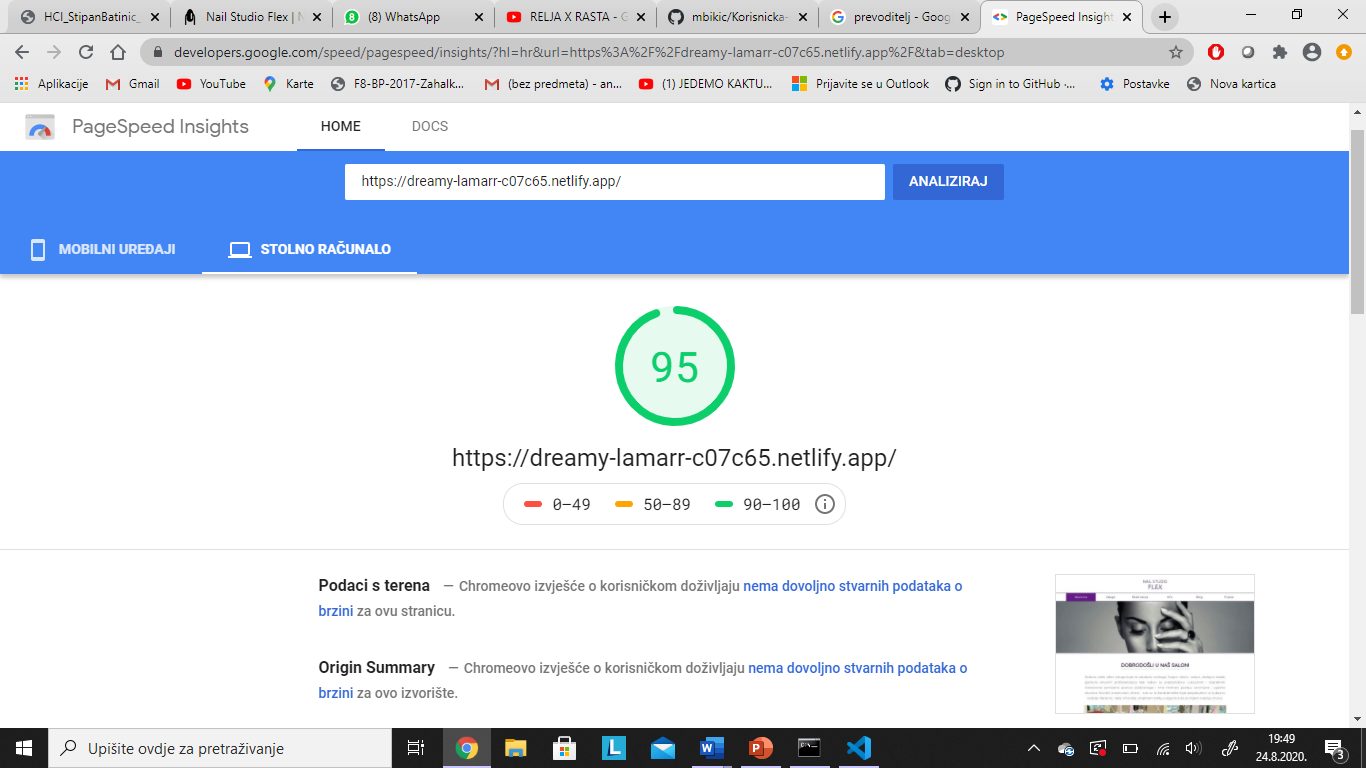
*Slika 22 - Blog*



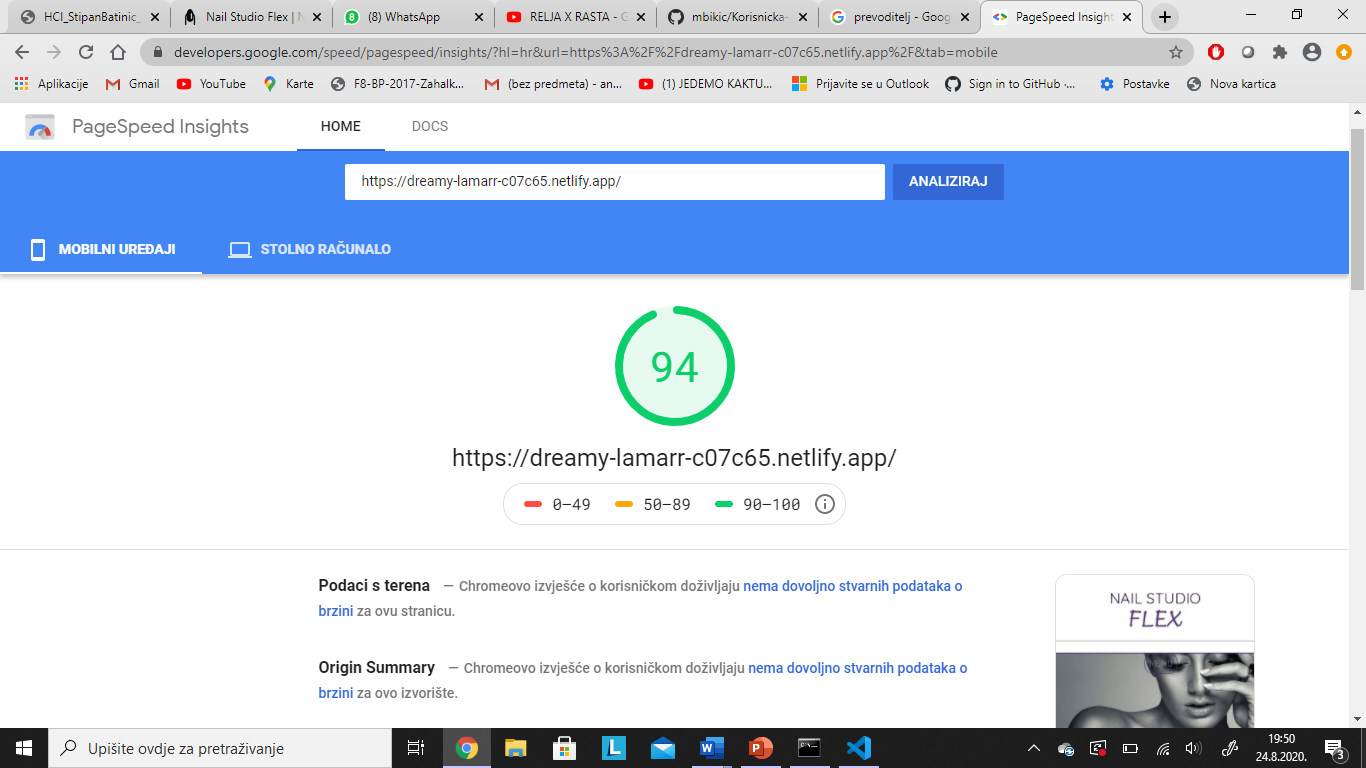
*Slika 23 - Prijava*

# PageSpeed Insights

Nail Studio Flex stranicu testirali smo koristeći Google alat [PageSpeed Insights](https://developers.google.com/speed/pagespeed/insights/). Dobiveni rezultati pokazuju da stranica ima odlično prvo renderiranje sadržaja i indeks brzine te da stranica u najkraćem roku postaje interaktivna i za desktop i za mobilnu verziju.



*Slika 24 - PageSpeed rezultat za stolno računalo*



*Slika 25 - PageSpeed rezultat za mobilni uređaj*

# Zaključak

Kroz postupak izrade projekta, prvi korak je da ispitamo korisnika koji su njegovi zahtjevi i želje, te nakon toga da otkrijemo metodu pomoću koje je moguće implementirati korisnikove zahtjeve. Koristimo usvojene heurističke metode i CRAP principe kako bismo ispunili korisnikove želje. Dizajn bi trebao biti jednostavan i instinktivan za korištenje, a to ćemo i postići ako koristimo pravila i principe koje smo opisali u prethodnim poglavljima.

.