

SVEUČILIŠTE U SPLITU,  
FAKULTET ELEKTROTEHNIKE, STROJARSTVA I BRODOGRADNJE

# DIGITAL MENU

(Web aplikacija za prikazivanje i vođenje digitalnog meni-a za restorane, hotele i barove)

**Kolegij:**

Korisnička sučelja

**Mentor:**

Mario Čagalj

**Studenti:**

Ante Kuić, Jakov Melvan

Split, veljača 2023.

# Sadržaj:

1.	Uvod .....	3
1.2	Korisnici .....	3
1.3	Ograničenja korištenja sustava .....	3
1.4	Zahtjevi i zadaće sustava .....	3
2.	Sitemap .....	4
3.	Low-fidelity prototip .....	5
2.1	Low-fidelity prototip za desktop verziju .....	5
2.2	Low-fidelity prototip za mobilnu verziju .....	9
4.	High-fidelity prototip .....	12
5.	Evaluacija prototipa .....	19
4.1	Heuristike iskoristivosti .....	19
4.1.1	Podudarnost između stvarnog svijeta i sustava .....	19
4.1.2	Konzistencija i standardi .....	20
4.1.3	Vidljivost statusa sustava .....	21
4.1.4	Korisnička sloboda i kontrola .....	23
4.1.5	Sprječavanje grešaka .....	24
4.1.6	Detekcija i oporavak od grešaka .....	25
4.1.7	Prepoznavanje umjesto prisjećanja .....	26
4.1.7	Fleksibilnost i efikasnost korištenja .....	26
4.1.9	Estetika i minimalistički dizajn .....	27
4.1.10	Pomoć i dokumentacija .....	29
4.2	CRAP principi .....	29
4.2.1	Contrast .....	29
4.2.2	Repetition .....	31
4.2.3	Alignment .....	32
4.2.4	Proximity .....	34
4.3	Teorije niske razine .....	35
4.3.1	Fittsov zakon .....	35
4.3.2	Hickov zakon .....	37
6.	Implementacija prototipa .....	39
7.	Zaključak .....	40

# 1. Uvod

Digital menu je web aplikacija za prikazivanje i vođenje digitalnog meni-a za restorane, hotele i barove.

Cilj ovog projekta je izrada aplikacije gdje korisnici putem web preglednika mogu pogledati ponudu uslužnog objekta, a osoblje putem dashboard-a upravljaju sadržajem na aplikaciji.

## 1.2 Korisnici

Web aplikacija namijenjena je širokom krugu korisnika. Pretpostavka je da će većina korisnika biti starosti između 15 i 70 godina te će imati barem osnovno informatičko znanje i iskustvo s radom na računalu. Međutim, ni ostali korisnici ne bi trebali imati problema u snalaženju na web stranici zbog jednostavnosti i *user-friendly* sučelja.

Dashboard dio aplikacije koristit će osoblje objekta, te će im se prije korištenja dati osnovne upute za korištenje istog, iako je dashboard koncipiran da bude vrlo lako shvatljiv.

## 1.3 Ograničenja korištenja sustava

Da bi mogao koristiti web aplikaciju, korisnik bi trebao:

- imati pristup Internetu
- imati uređaj s web preglednikom

Da bi korisnik mogao uređivati sadržaj web aplikacije, trebao bi osim gore navedenih ograničenja:

- imat pristupne podatke ( email i lozinku )

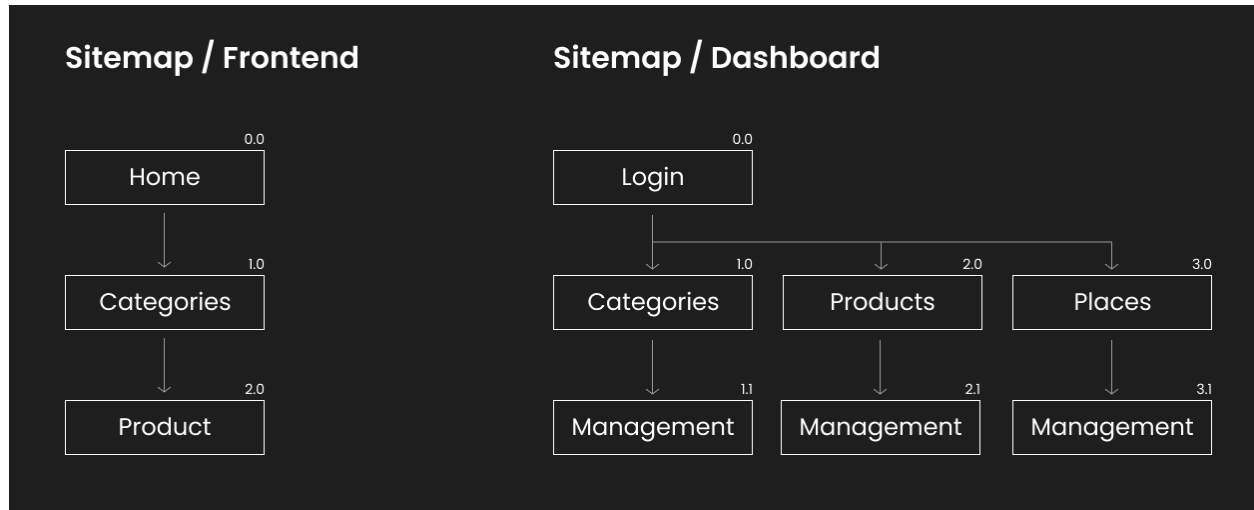
## 1.4 Zahtjevi i zadaće sustava

Zahtjevi i zadaće koje web aplikacija mora imati:

- web aplikacija koristit će se iz web preglednika
- web aplikacija koristit će se na uređajima različite veličine (*smartphones*, tableti, laptopi i desktop računala)
- korisnik će moći filtrirati produkte
- broj produkata koji se nudi putem web-a veći je od 20
- korisnik će moći koristiti web aplikaciju kao gost ili će se prijaviti u vlastiti profil
- broj javnih stranica je 5
- jedna od javnih stranica je blog (blog mora sadržavati minimalno 20 postova koje mogu uključivati slike, videa, *code snippet-e*)

## 2. Sitemap

Prije izrade prototipova izrađena je sitemap-a u Figmi. Sitemap-a nam pomaže kako bi lakše strukturirali aplikaciju i odredili smjer korištenja aplikacije. Prilikom same izrade prototipova sitemap-a nam je pomogla da lakše uočimo kako su stranice na web aplikaciji povezane i također da budemo sigurniji da smo pokrili sve zadaće sustava.



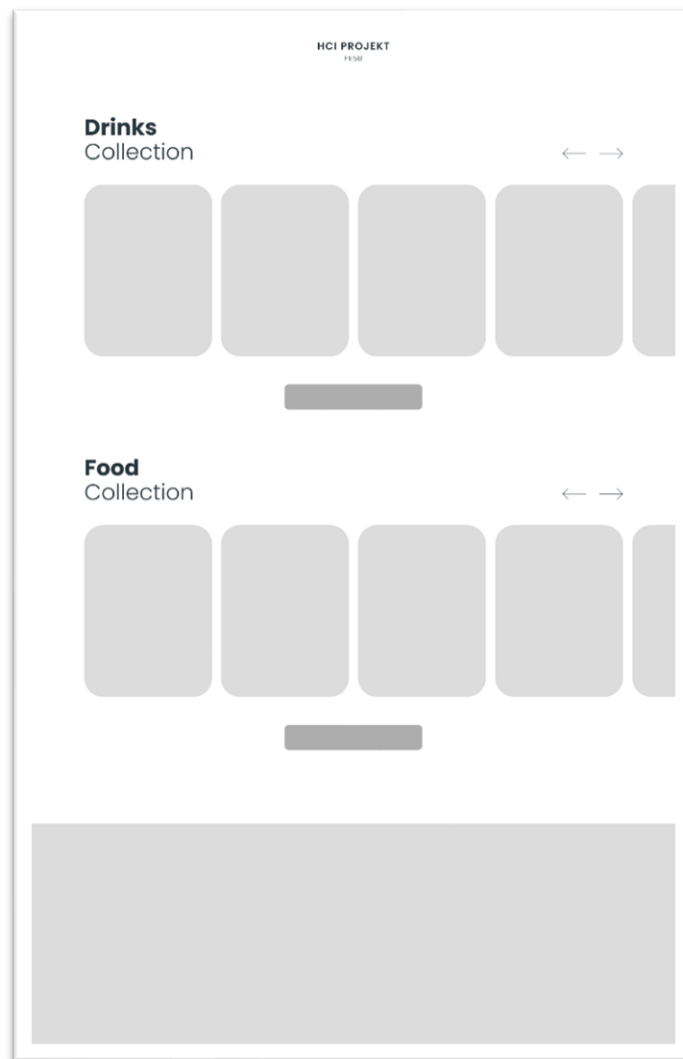
*Slika 1 Sitemap web aplikacije*

### 3. Low-fidelity prototip

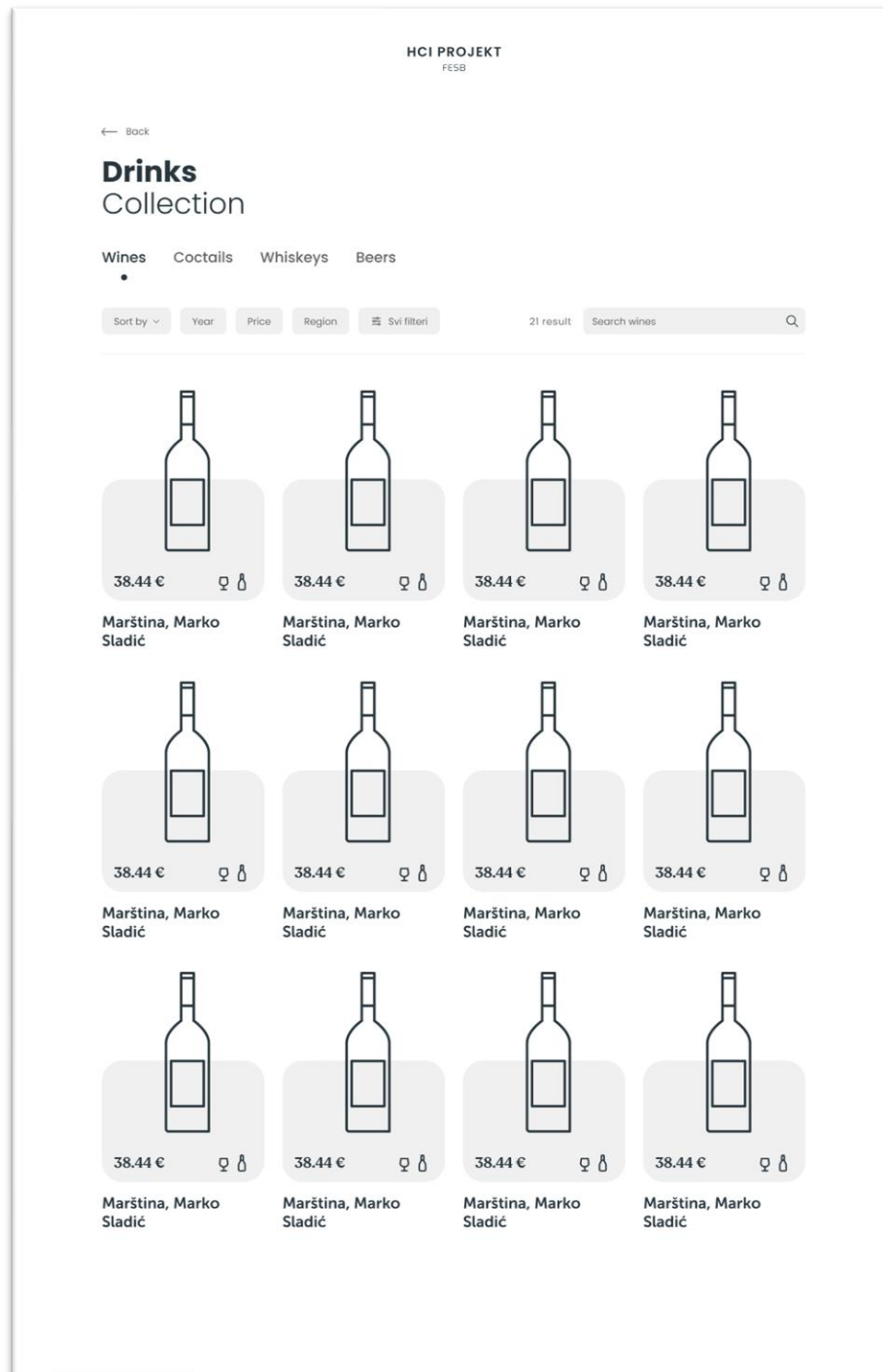
Prvi korak u izradi projekta je izrada *low-fidelity* prototipa. Cilj *low-fidelity* prototipa nije prikazati izgled finalnog proizvoda, već je to jednostavni način prikazivanja koncepta dizajna u svrhu početnog testiranja. Prototip je napravljen u Figma.

#### 2.1 Low-fidelity prototip za desktop verziju

Oznake ispod slika bit će nazvane kao stranice unutar sitemap-e.



Slika 2 Low-fidelity „Home“ stranice ( frontend )



Slika 3 Low fidelity „Categories“ stranice ( frontend )

[Drinks](#) > [Wines](#) > Marština, Marko Sladić



## Marština, Marko Sladić

### Sizes

		
20ml	500ml	750ml
7.84 €	30.44 €	38.44 €

### Alcohol

13.5%

### Region

Istra, Hrvatska

### Description

The white wine with bright color and fresh fruity taste. Soaked in oaked barrels for 4 months after fermentation with 22C. This wine was a winner of Sakura Woman's Wine Awards (Japan) in 2017.

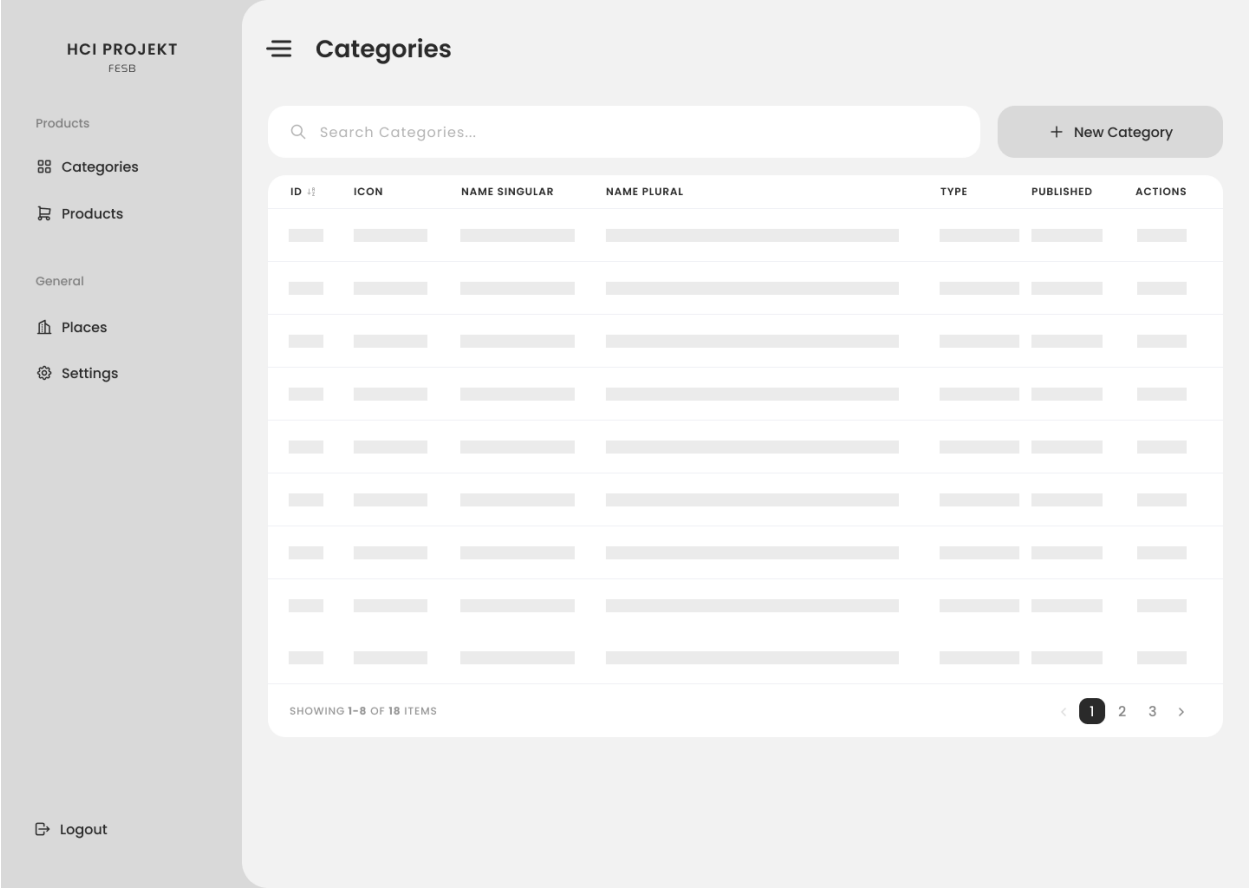
### Additional info

<b>Weight</b>	2 kg
<b>Type</b>	Red wine
<b>Winery</b>	Bertoša
<b>Sort</b>	Cabernet Sauvignon

## Best food match

			
38.44 €  	38.44 €  	38.44 €  	38.44 €  
Marština, Marko Sladić	Marština, Marko Sladić	Marština, Marko Sladić	Marština, Marko Sladić

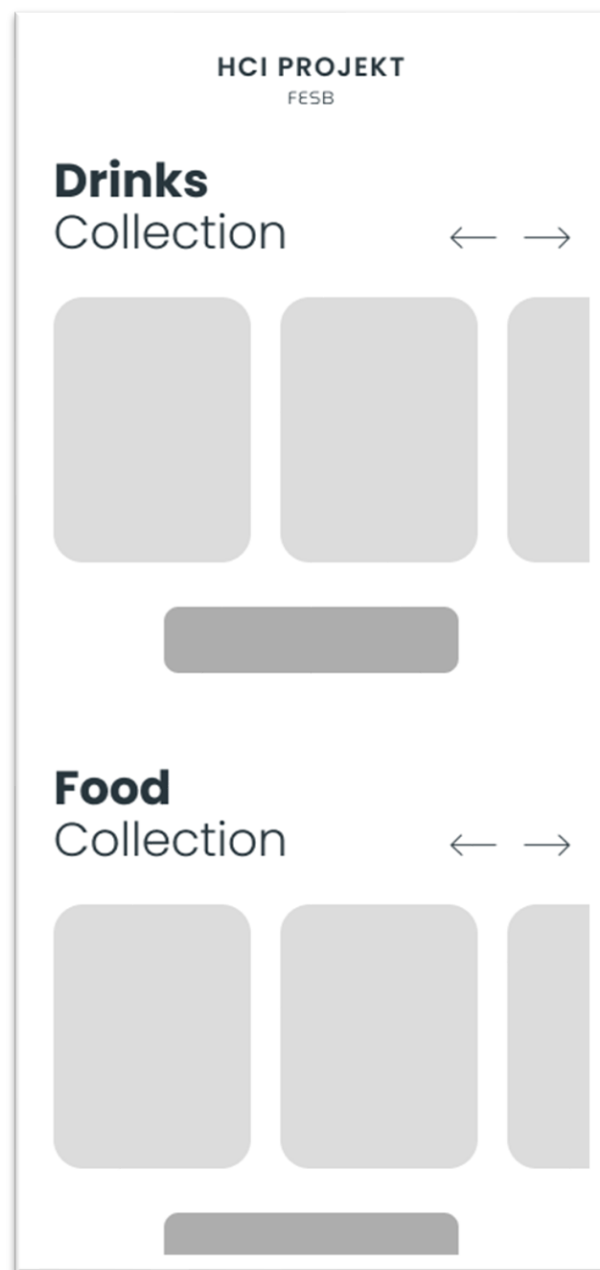
Slika 4 Low-fidelity „Product“ stranice ( frontend )



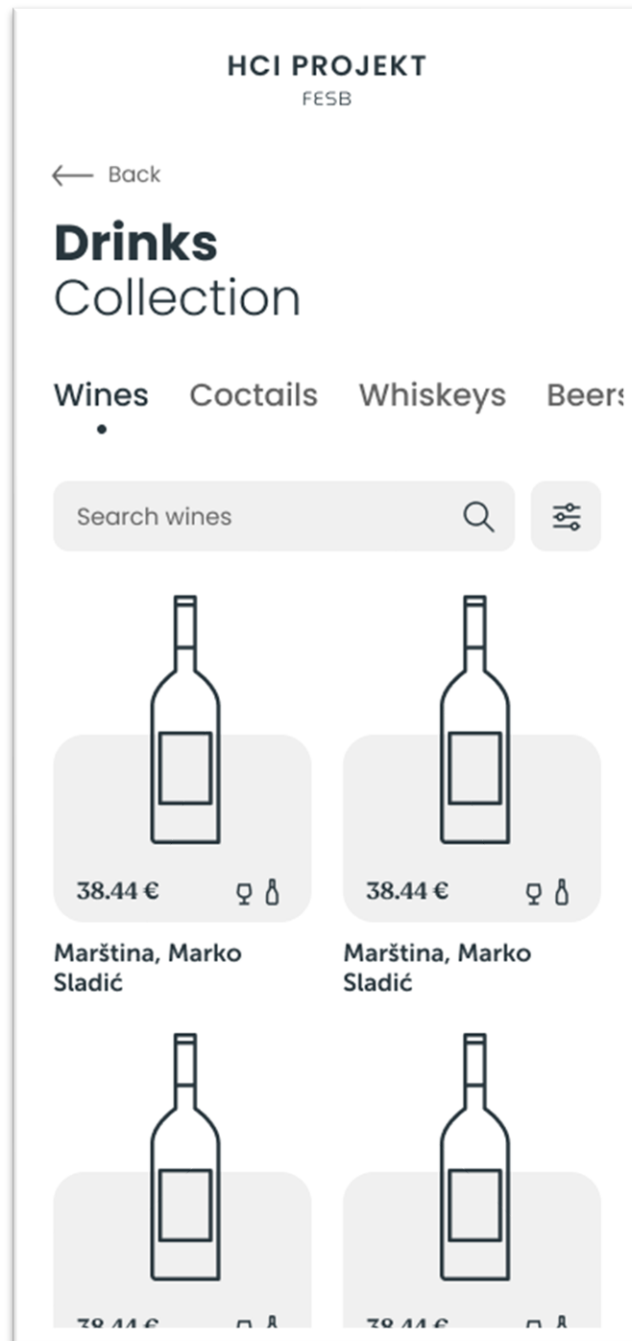
Slika 5 Low-fidelity dashboard-a



## 2.2 Low-fidelity prototip za mobilnu verziju



Slika 6 Low-fidelity mobile prototip za „Home“ stranicu ( Frontend )



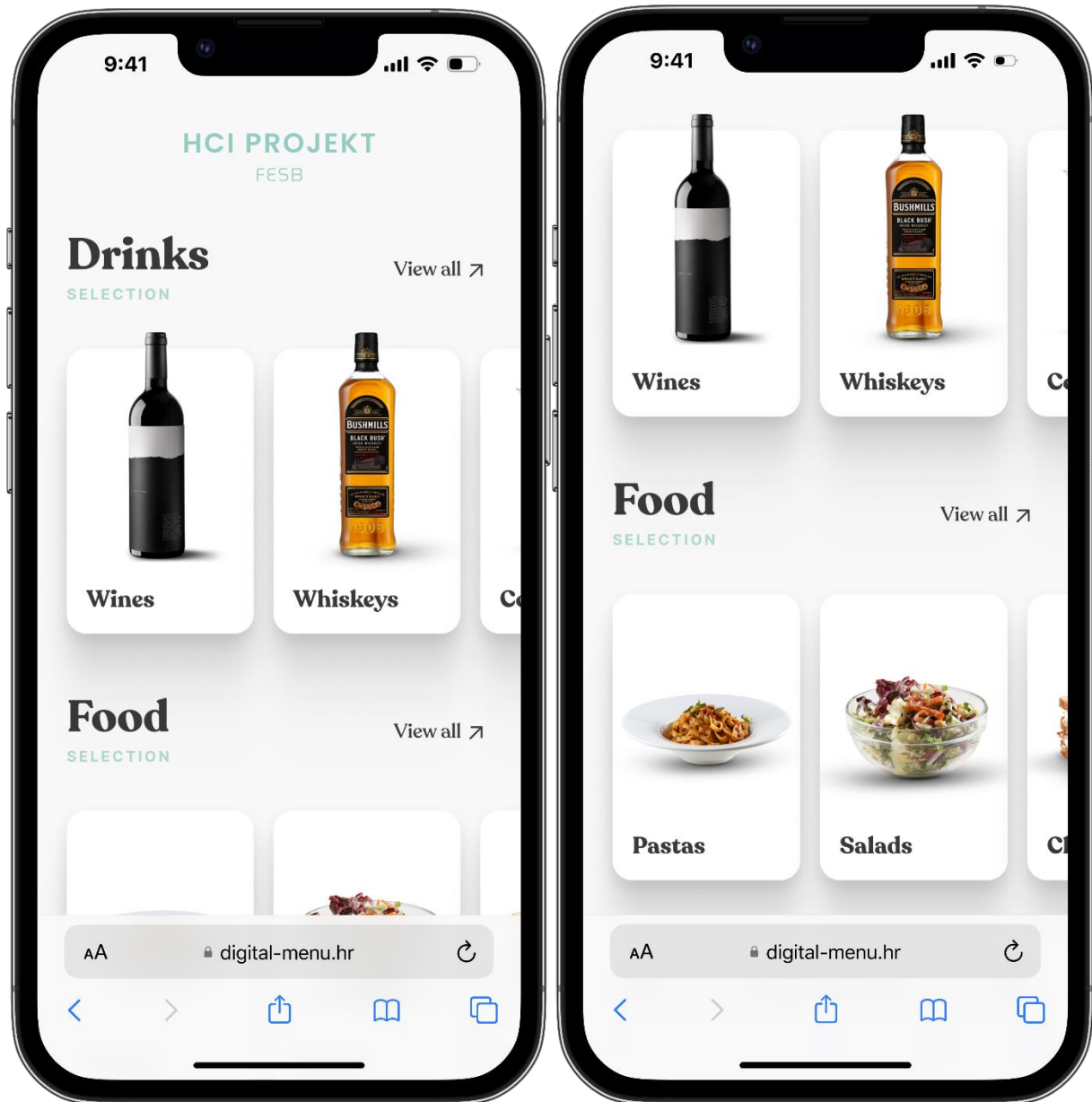
Slika 7 Low-fidelity mobile prototip za „Categories“ stranicu ( Frontend )



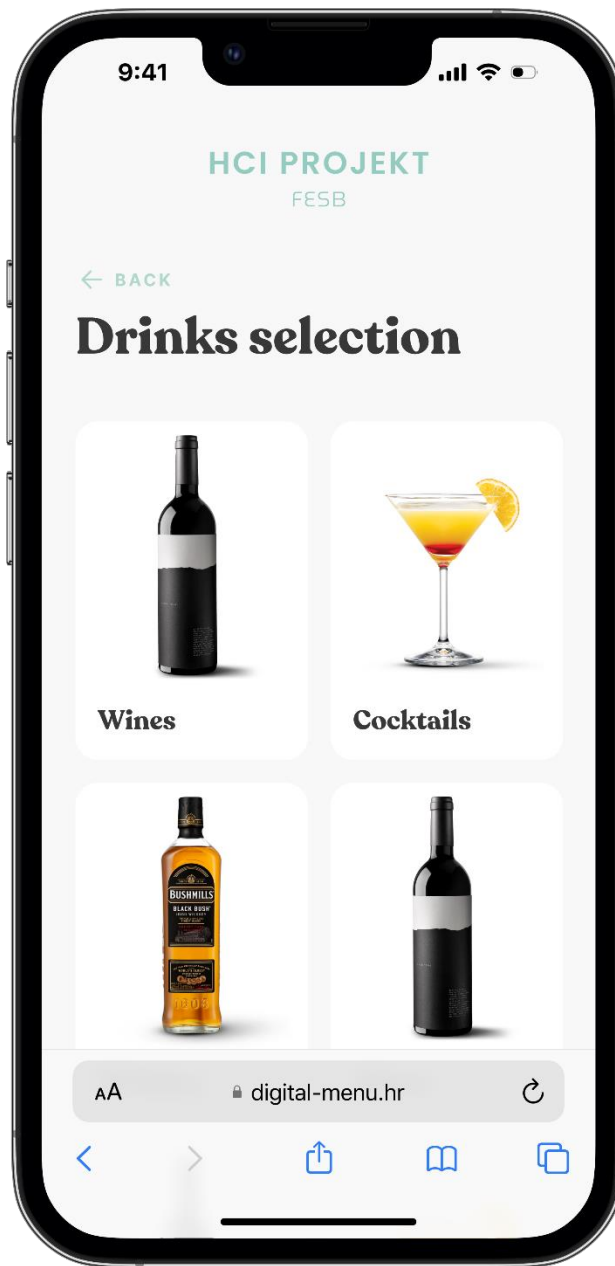
Slika 8 Low-fidelity mobile prototip za „Product“ stranicu ( Frontend )

## 4. High-fidelity prototip

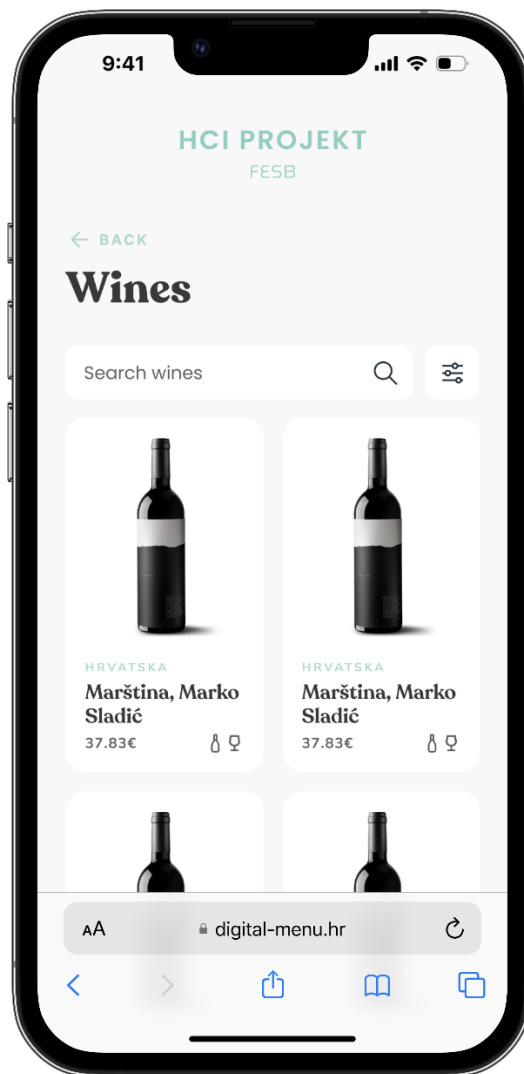
Nakon izrade low-fidelity prototip-a slijedi izrada high-fidelity prototip-a za mobile/desktop. Za izradu high-fidelity prototip-a koristili smo također Figma. Pomoću high-fidelity prototipa može se korisnicama prikazati stranica na interaktivan i funkcionalan način. Izradili smo high-fidelity mobile verziju za „frontend“ dio, a desktop verziju za „dashboard“ dio.



Slika 9 High-fidelity „Home“ stranice



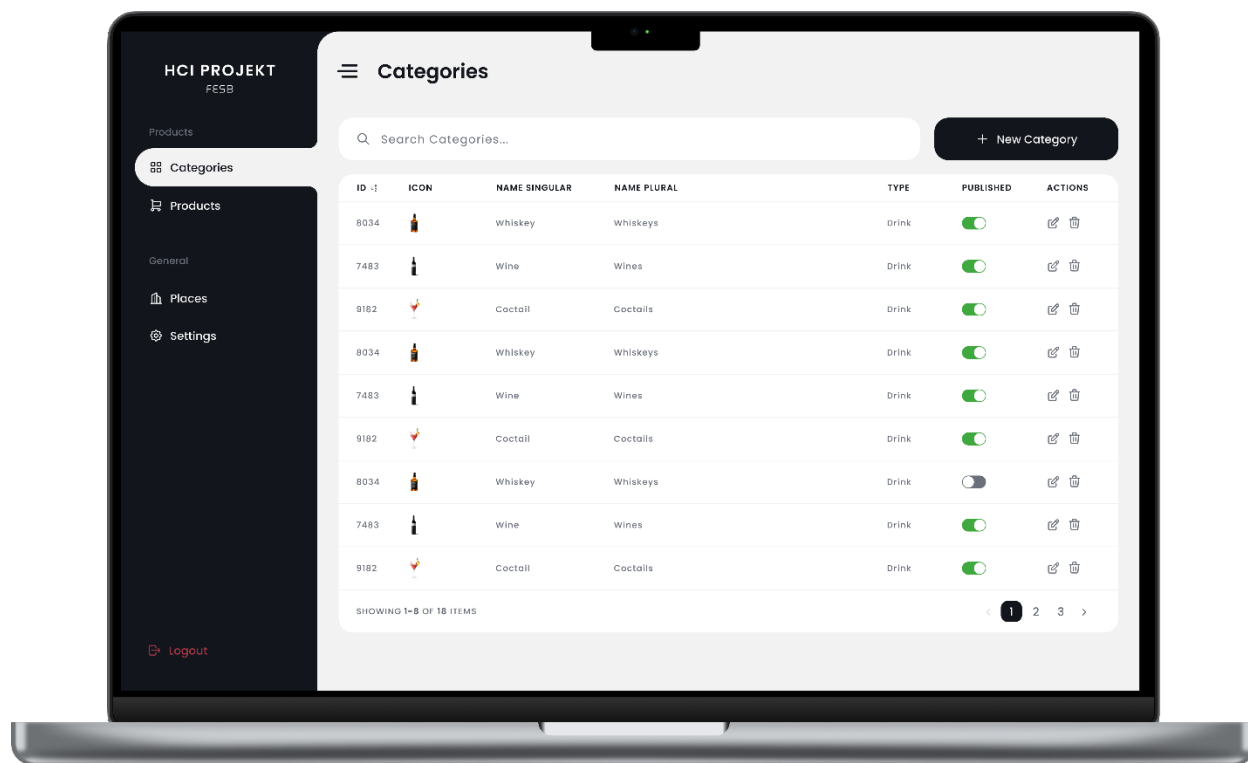
Slika 10 High-fidelity „View all“ stranice



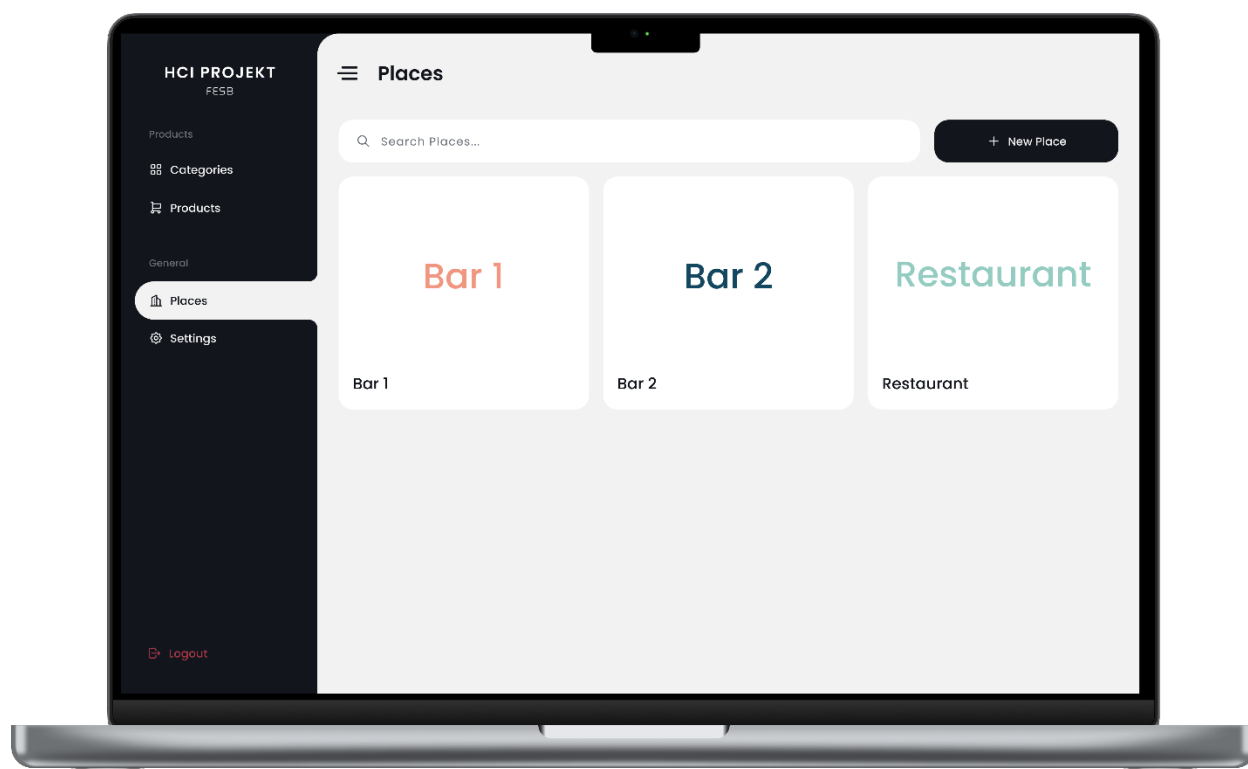
Slika 11 High-fidelity „Categories“ stranice



Slika 12 High-fidelity „Product“ stranice

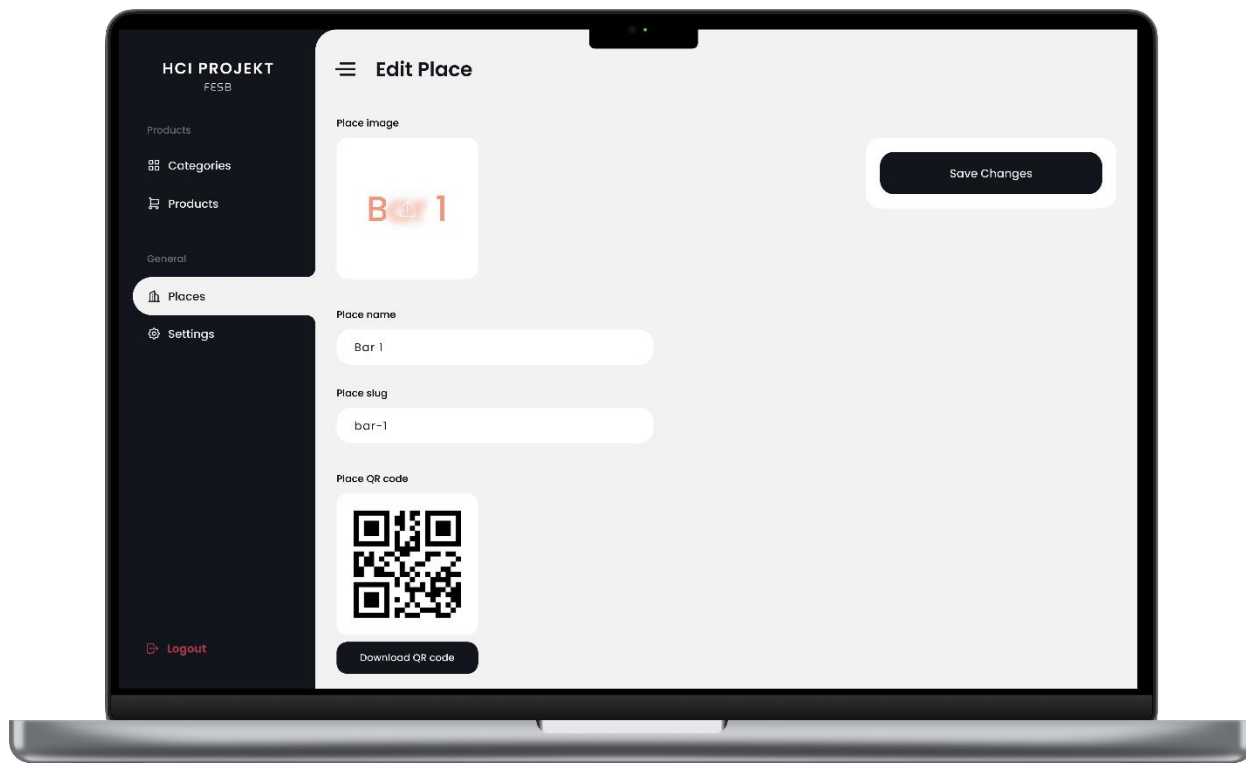


Slika 13 High-fidelity „Categories“ stranice ( Dashboard )

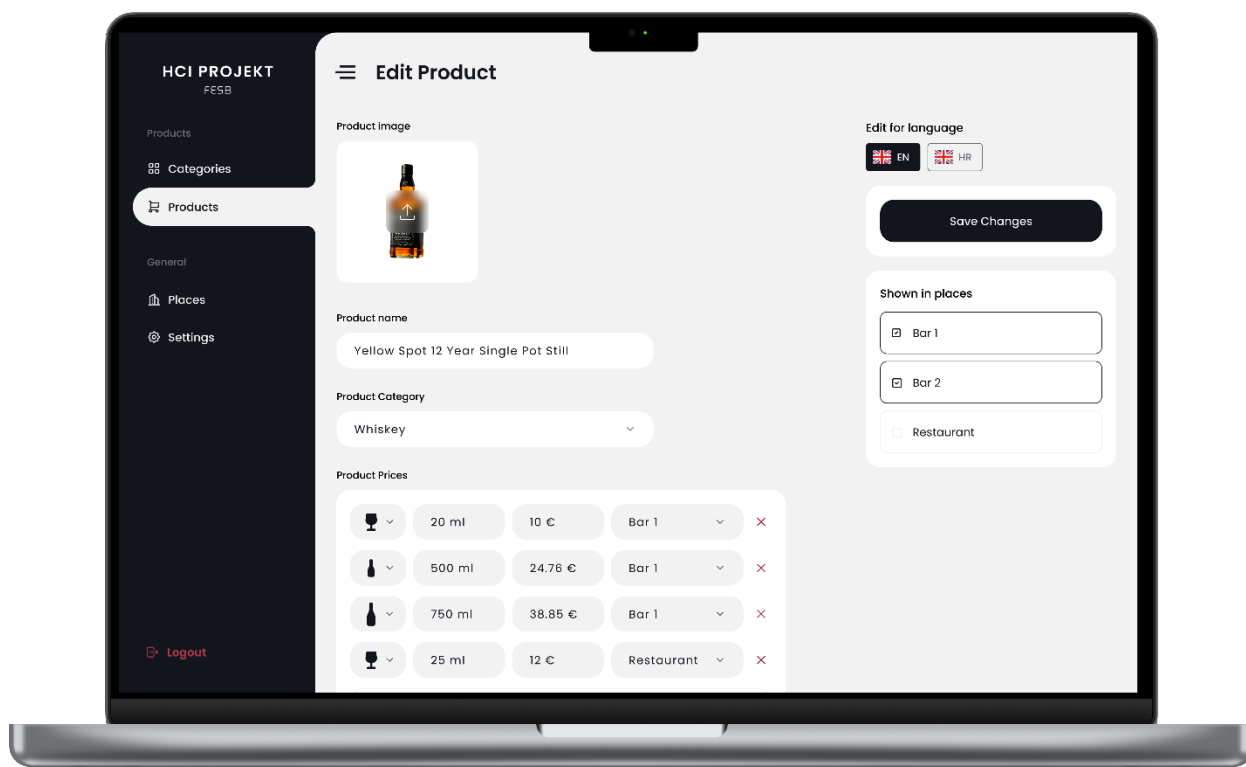


Slika 14 High-fidelity „Places“ stranice

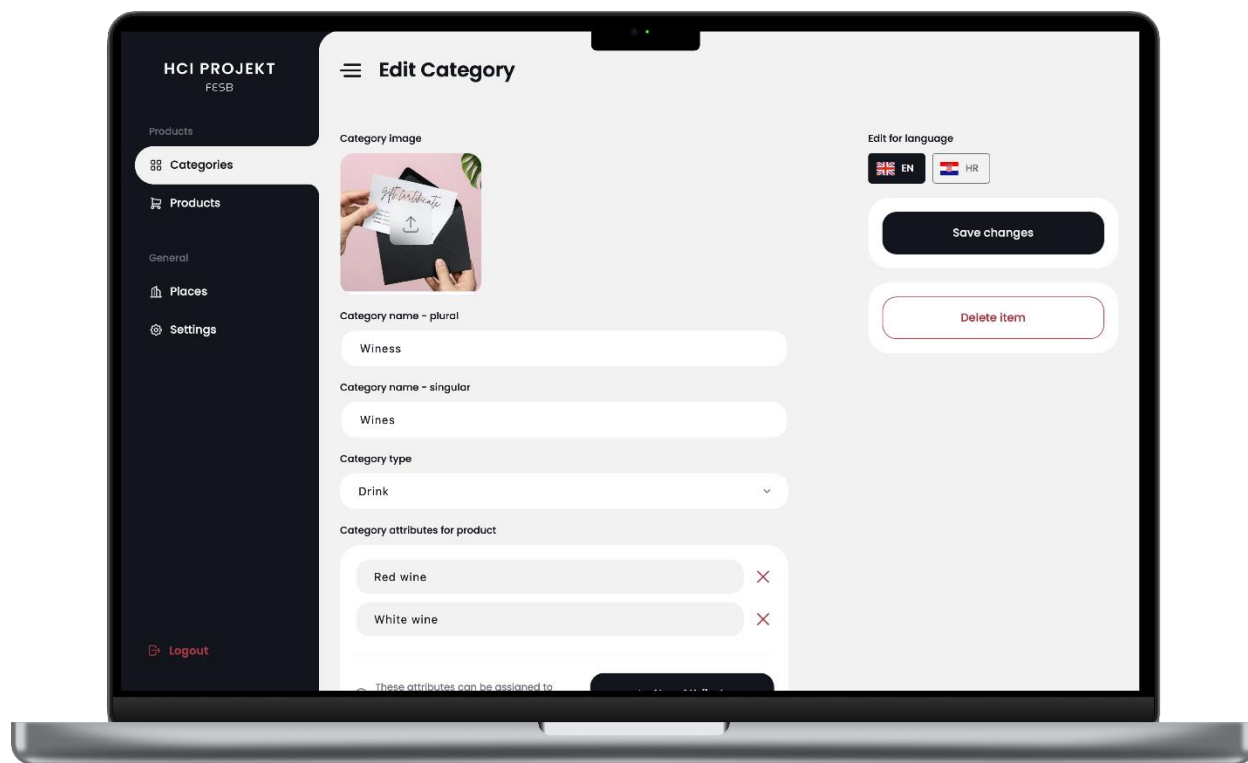




Slika 15 High-fidelity „Edit place“ stranice



Slika 16 High-fidelity „Edit product“ stranice



Slika 17 High-fidelity „Edit category“ stranice

## 5. Evaluacija prototipa

Na temelju principa i heuristika dizajna, prototip je poboljšán kako bi konačan proizvod imao što bolje korisničko iskustvo.

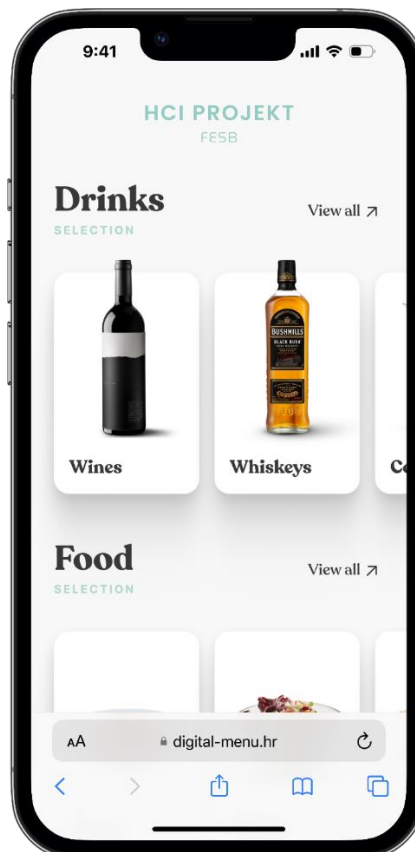
### 4.1 Heuristike iskoristivosti

Jacob Nielsen definirao je deset heuristika iskoristivosti za korisnička sučelja. U nastavku su navedene heuristike te načini na koji su provedene u ovom projektu kako bi se poboljšalo korisničko sučelje.

#### 4.1.1 Podudarnost između stvarnog svijeta i sustava

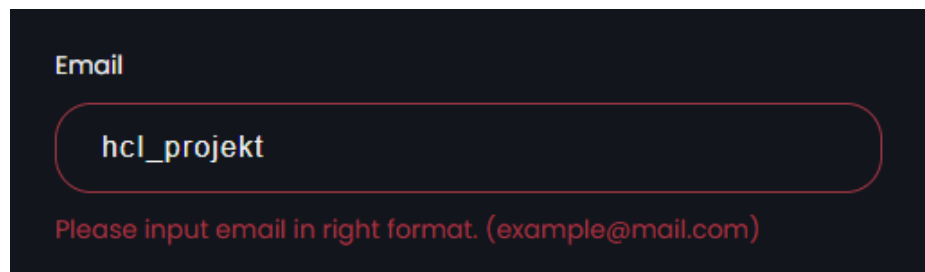
Sustav bi trebao pričati „korisnikovim“ jezikom, sa riječima, frazama i konceptima koji su poznati korisniku, a ne stručnim rječnikom. Informacije bi trebale biti prikazane prirodnim i logičkim putem.

Primjer korištenja heuristike vidljiv je odmah na početnoj stranici gdje nalaze lista kategorija. Kategorije su podijeljene po tipovima ( npr. hrana i piće ) te su korisniku prikazani kao kolekcija pića i kolekcija hrane. Korisniku ovaj način prikazivanja je razumljiviji i intuitivniji, nego da piše lista kategorija.



Slika 18 Primjer korištenja heuristike „Podudarnost između stvarnog svijeta i sustava“

Još jedan primjer korištenja ove heuristike bi bio prilikom pokušaja prijave u dashboard. Kada korisnik ne unese email adresu u pravom formatu, sustav mu vraća poruku koja nije iz tehničkog žargona, već korisnik odmah može razumjeti što treba ispraviti.

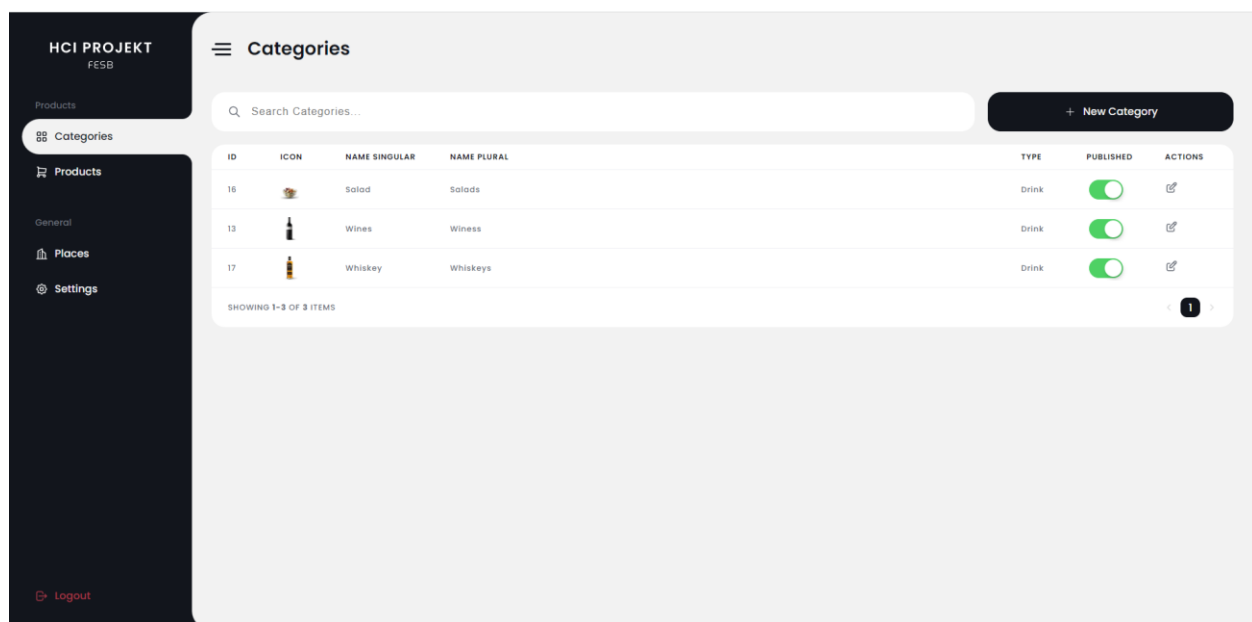


Slika 19 2. primjer korištenja heuristike „Podudaranost između stvarnog svijeta i sustava“

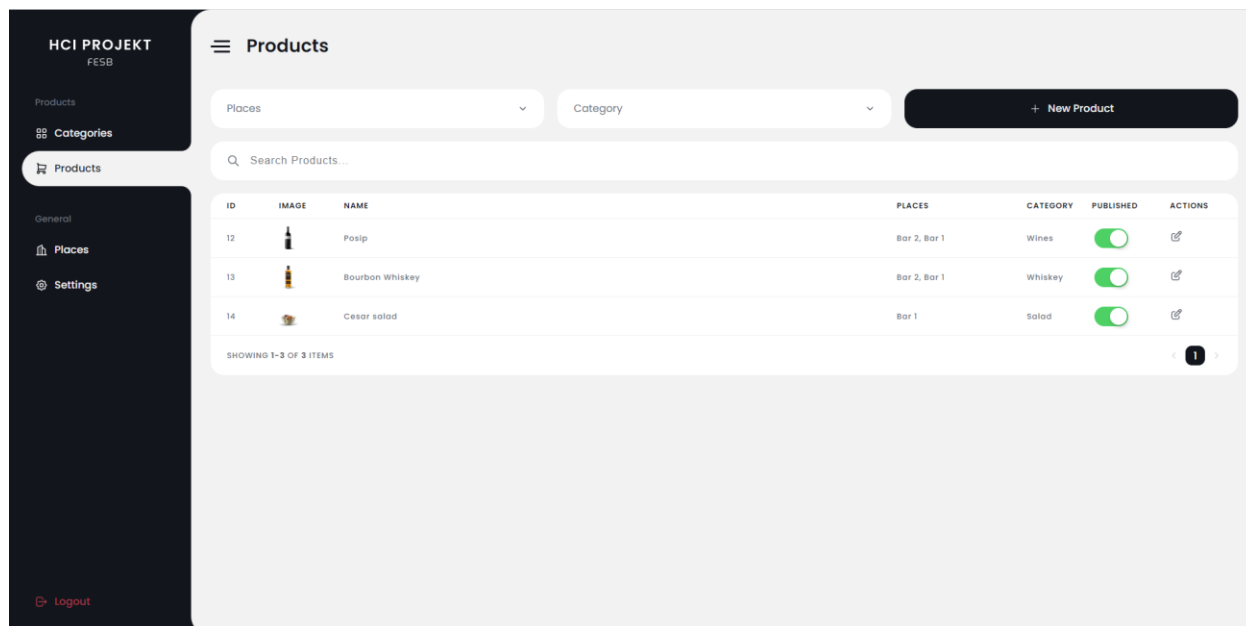
#### 4.1.2 Konzistentcija i standardi

Korisnici se ne bi trebali pitati znače li različite riječi, situacije ili akcije isto, već sustav mora slijediti inicijalno postavljanje konvencije.

Bitan je konzistentan izgled elementa i stranice koji imaju istu akciju u odnosu na situaciju. Jedan od primjera korištenja ove heuristike je prikazivanje proizvoda i kategorija u tablici, odnosno konzistentnost samog sučelja. Može se vidjeti da je na obje stranice botun za dodavanje novog proizvoda/kategorije na istom mjestu, također prikazani su na isti način. Također i stranice kojima se uređuje proizvod/kategorija imaju botune za uređivanje/brisanje na istom mjestu.

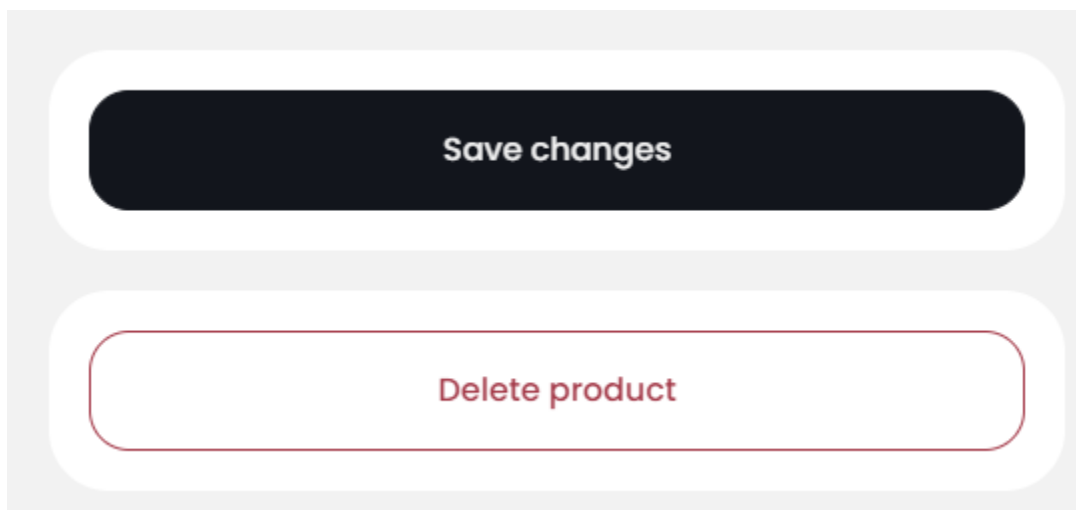


Slika 20 Stranica „Categories“ ( konzistentnost )



Slika 21 Stranica „Products“ ( konzistentnost)

Također konzistentno su prikazani botuni koji rade iste i različite akcije, primjer je pokazan na slici ispod. Botuni imaju isti izgled, ali različito isticanje što asocira na različitu akciju.

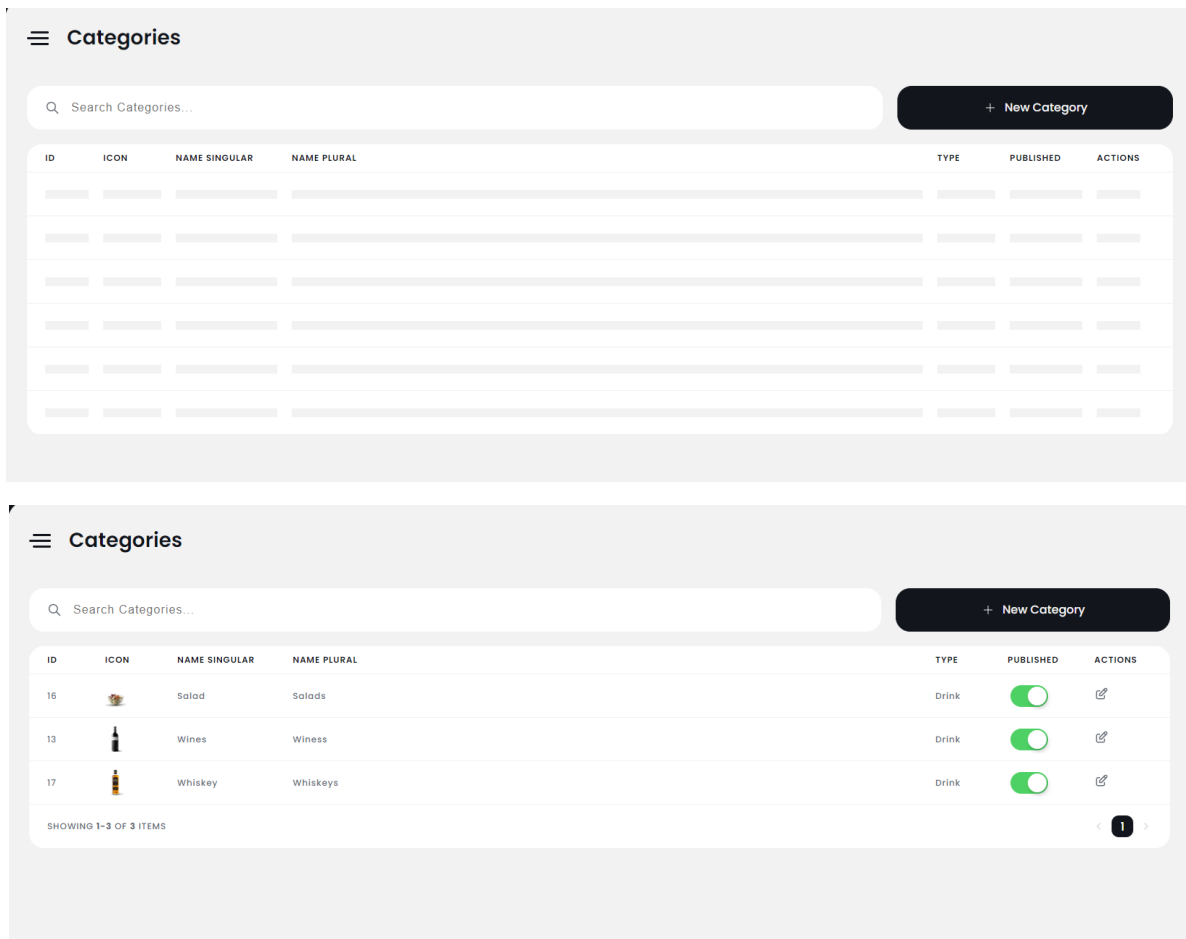


Slika 22 CTA botuni „Save changes“ i „Delete product“

#### 4.1.3 Vidljivost statusa sustava

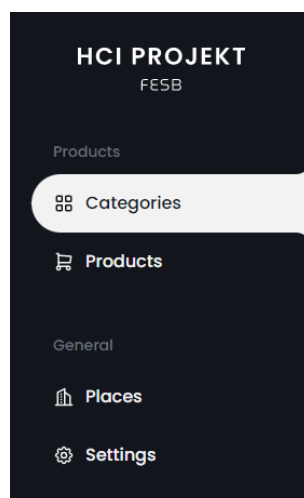
Sustav bi trebao u svakoj situaciji i u prikladnom vremenu informirati korisnika o tome što se događa.

Primjer korištenja ove heuristike je skeleton na tablicama prilikom učitavanja sadržaja. Na ovaj način korisnik zna da treba čekati dok se sadržaj ne učita. Primjer heuristika je prikazan na slici ispod.



Slika 23 Skeleton tablice

Također još jedan primjer ove heuristike bi bio side bar koji korisniku pokazuje na kojoj stranici se nalazi, a to je prikazano na slici ispod.



Slika 24 Sidebar status

#### 4.1.4 Korisnička sloboda i kontrola

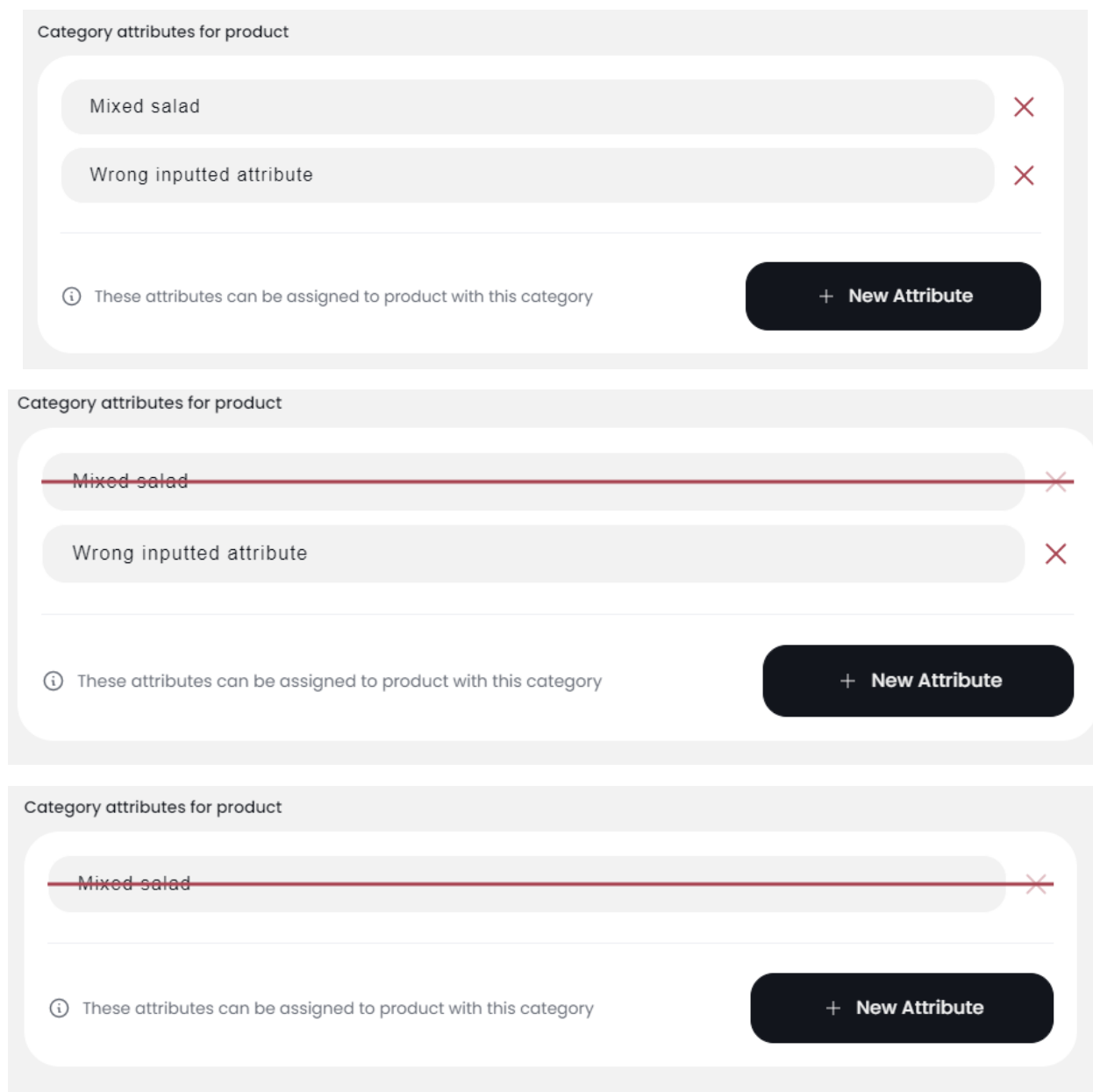
Korisnici često odaberu funkcionalnosti koje nisu željeli te je potrebno precizno definirati „izlaz u slučaju nužde“ kako bi napustili neželjeno stanje bez da moraju prolaziti kroz produženi dijalog.

Najjednostavniji primjer ove heuristike bi bio da ukoliko je korisnik greškom ušao u neki proizvod, sustav mu omogućuje povratak na stranicu natrag ( na kojoj je bio ) klikom na botun „Back“. To je prikazano na slici ispod.



Slika 25 Stranica proizvoda, „back“ botun

Također još jedan primjer ove heuristike bi bio jednostavno upravljanje atributima. Prilikom brisanja atributa, postoje dvije situacije. Ukoliko je atribut pohranjen u bazu podataka, on će se precrtati, a ukoliko nije on će se izbrisati ( primjer „undo“ operacije ).



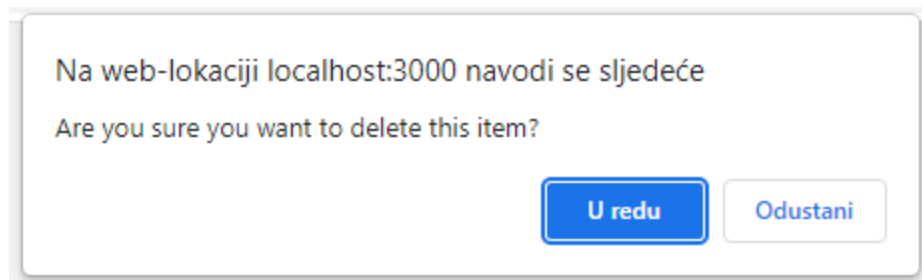
*Slika 26 „Undo“ operacija na atributima*

#### 4.1.5 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.

Prilikom brisanja proizvoda/kategorija korisniku se pojavljuje validacijski prozor gdje potvrđuje svoju akciju.





*Slika 27 Prompt prozor prilikom brisanja stavki*

Također prilikom unošenja podataka, ukoliko korisnik ne unese određeni podatak, sustav ga obavještava da je taj podatak potrebno unijeti. Na slici ispod prikazan je primjer gdje korisnik pokušava spremiti promjene na određenom proizvodu, a nije unio ime proizvoda.

#### 4.1.6 Detekcija i oporavak od grešaka

Obavijesti o greškama moraju biti prikazane na korisniku razumljiv način, prikaz problema mora biti očit, nedvosmislen i uključivati potencijalni način rješavanja problema.

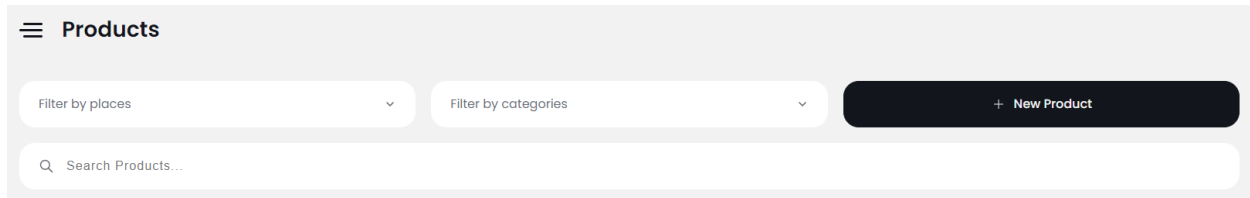
Ovu heuristiku je implementirana na inputima gdje se korisnika obavještava da je određeni input potrebno unijeti, a gdje je potrebno i format samog unosa. (npr. Slika 19 2. primjer korištenja heuristike „Podudaranost između stvarnog svijeta i sustava“) Korisniku se prilikom spremanja određenih promjena ukoliko postoji greška, vraća pripadajuća poruka greške, odnosno „hint“ za rješenje.

*Slika 28 Informiranje o greški*

#### 4.1.7 Prepoznavanje umjesto prisjećanja

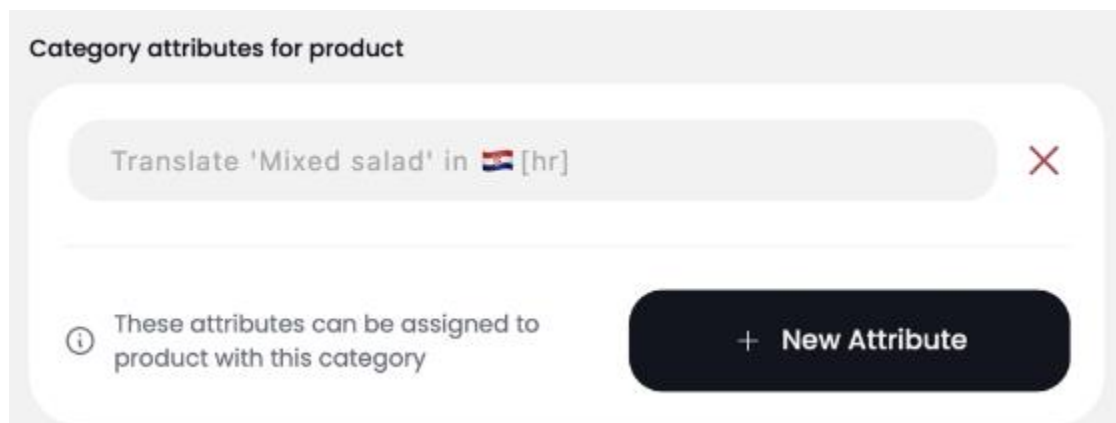
Sustav treba minimizirati količinu podataka koju korisnik treba pamtiti. Korisnik ne bi trebao pamtiti podatke iz jednog dijaloga u drugi. Instrukcije za korištenje sustava trebaju biti vidljive kada je to prikladno.

Unutar polja za pretraživanje nalazi se ikona povećala koja korisnika odmah asocira da se radi o polju za pretraživanje. Popratni tekst unutar polja dodatno opisuje što se pretražuje. Također filtriranje proizvoda omogućeno je preko select boxova, a uz popratni tekst „Filter by ...“ korisnik odmah zna da se radi o filtriranju.



*Slika 29 Primjer heuristike „Prepoznavanje umjesto prisjećanja“*

Također još jedan primjer ove heuristike bi bila pomoć prilikom prevođenja određenog sadržaja proizvoda/kategorije gdje se korisniku nalazi originalni sadržaj povezan sa zastavom jezika na koji se prevodi.



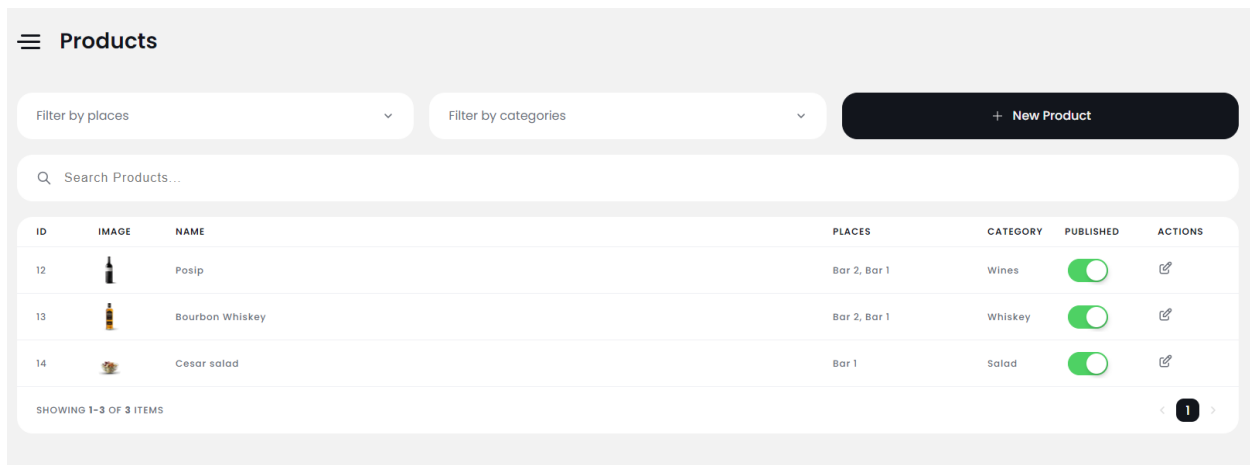
*Slika 30 2. primjer heuristike "Prepoznavanje umjesto prisjećanja"*

#### 4.1.7 Fleksibilnost i efikasnost korištenja

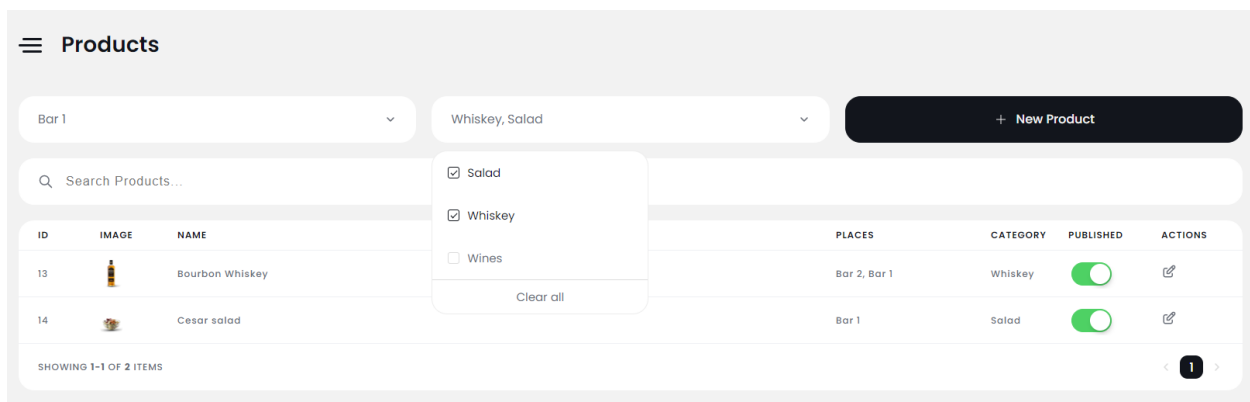
Ubrzavanja koja su često nevidljiva novim korisnicima, mogu ubrzati rad korisniku koji je upoznat sa sustavom. Sustav bi trebao biti takav da dobro odgovara i novom i uhodanom korisniku.

Primjer korištenja ove heuristike bi bio prilikom prijave u sustav, korisnik klikom na „enter“ tipku obavlja istu akciju kao da je stisnuo „Login“.

Također još jedan primjer korištenja ove heuristike bi bili dinamički filteri gdje se korisniku ubrzava pretraživanje proizvoda po kategorijama i mjestima.



Slika 31 Prikaz proizvoda prije filtriranja

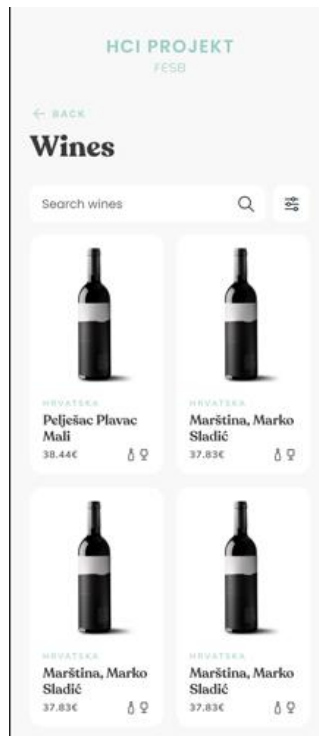


Slika 32 Prikaz proizvoda nakon filtriranja

#### 4.1.9 Estetika i minimalistički dizajn

Dijalozi ne smiju sadržavati informacije koje u tom trenutku nisu bitne ili se rijetko koriste. Svaka dodatna informacija na dijalogima konkurira postojećim, važnijim informacijama i tako smanjuje njihovu vidljivost.

Dizajn je zamišljen da su proizvodi izlistani samo sa osnovni informacijama, a prilikom ulaska u pojedini proizvod dobivaju se sve informacije o istom. Na taj način sakrili smo nepotrebno u datom kontekstu.



Slika 33 Lista proizvoda

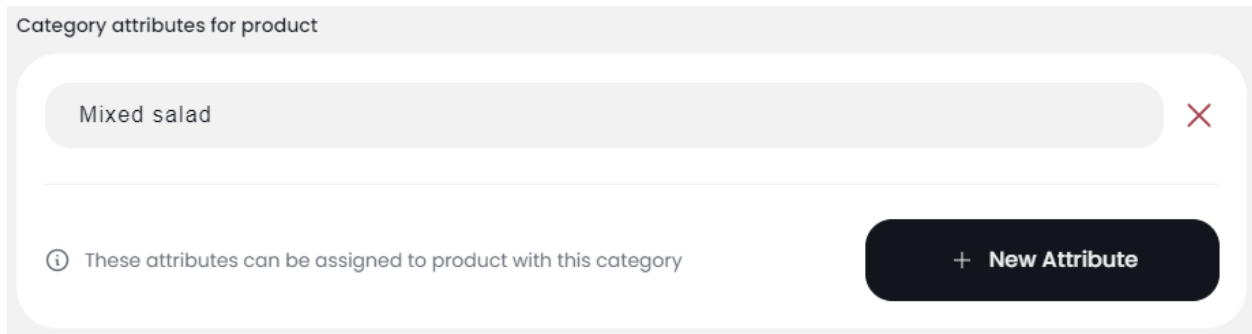


Slika 34 Informacije o proizvodu

#### 4.1.10 Pomoć i dokumentacija

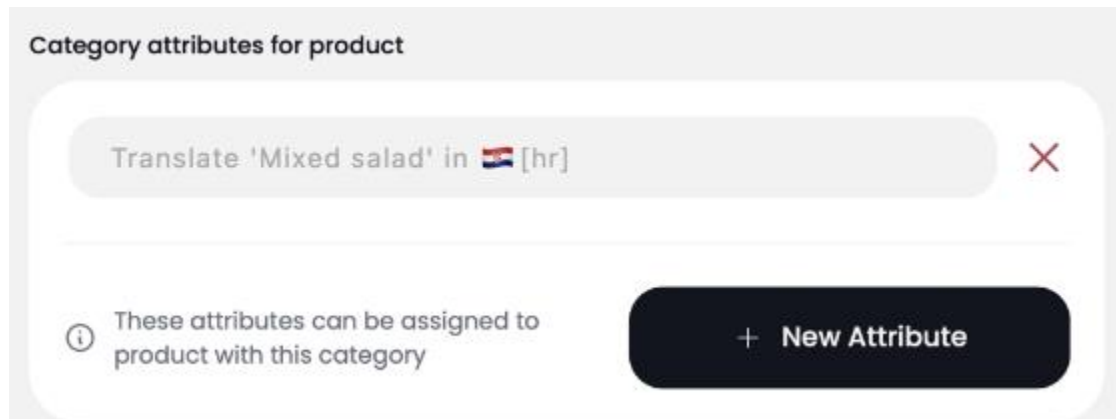
Iako bi bilo bolje da je sustav dizajniran na način da ga je moguće koristiti bez dodatne dokumentacije, nekad ju je nemoguće izbjeći. Pomoć mora biti lako dostupna i jasno objašnjavati funkcije koje su potrebne korisniku.

Primjer sljedeće heuristike prikazano je na slici ispod. Značenje atributa kategorije opisano je kraj „info“ ikone, kako bi korisniku bilo jasnije što ta značajka znači.



*Slika 35 Heuristika "Pomoć i dokumentacija"*

Još jedan primjer ove heuristike bi bio placeholder kada se prevodi određeni sadržaj proizvoda/kategorije gdje je korisniku napisan originalni naziv istog.



*Slika 36 2. primjer heuristike "Pomoć i dokumentacija"*

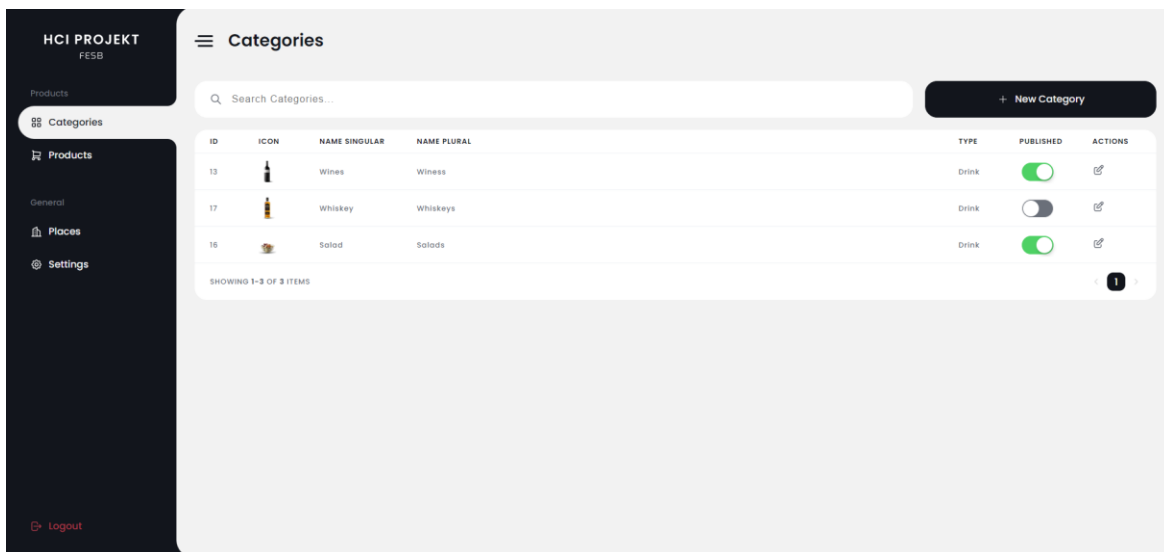
## 4.2 CRAP principi

CRAP je skup dizajnerski principa koji označavaju kontrast, ponavljanje, poravnanje i blizinu. Praćenje i razumijevanje ovih principa rezultira dobrim dizajnom u svim primjenama.

### 4.2.1 Contrast

Elementi koji imaju različitu namjenu ili različitu razinu važnosti, trebaju različito izgledati. Važniji elementi trebaju biti istaknutiji od manje važnih. Dobar kontrast rezultira višu zainteresiranost korisnika.

Prvi primjer korištenja ovog principa vidljiv je na slici ispod. Side bar se različitom pozadinom ističe od ostatka stranice. Time smo postigli odvajanje statičkog i dinamičkog djela stranice. Razlika u bojama te boja aktivnog navigacijskog elementa dodatno ističe sadržaj stranice, tako što stvara osjećaj da se sadržaj nalazi bliže korisniku, odnosno da je bitniji. CTA botun „New Category“ se ističe kontrastom boja u odnosu na pozadinu. Također aktivni element paginacije tablice je istaknut. U slučaju kada je svijetla pozadina koristimo tamni tekst, a njegova jačina boje ovisi o važnosti sadržaja. Tako su naprimjer naslovi stupaca tamniji u odnosu na popunjena polja.



Slika 37 Kontrast na primjeru dashboard-a

Drugi primjer korištenja ovog principa je veličina font-a kako bi se istakle bitnije informacije hijerarhijski. Tako je recimo naziv proizvoda najvećeg font size-a. Manje bitni elementi imaju manju veličinu fonta te su svjetlije boje. Također svjetlijom pozadinom ističemo sliku boce.

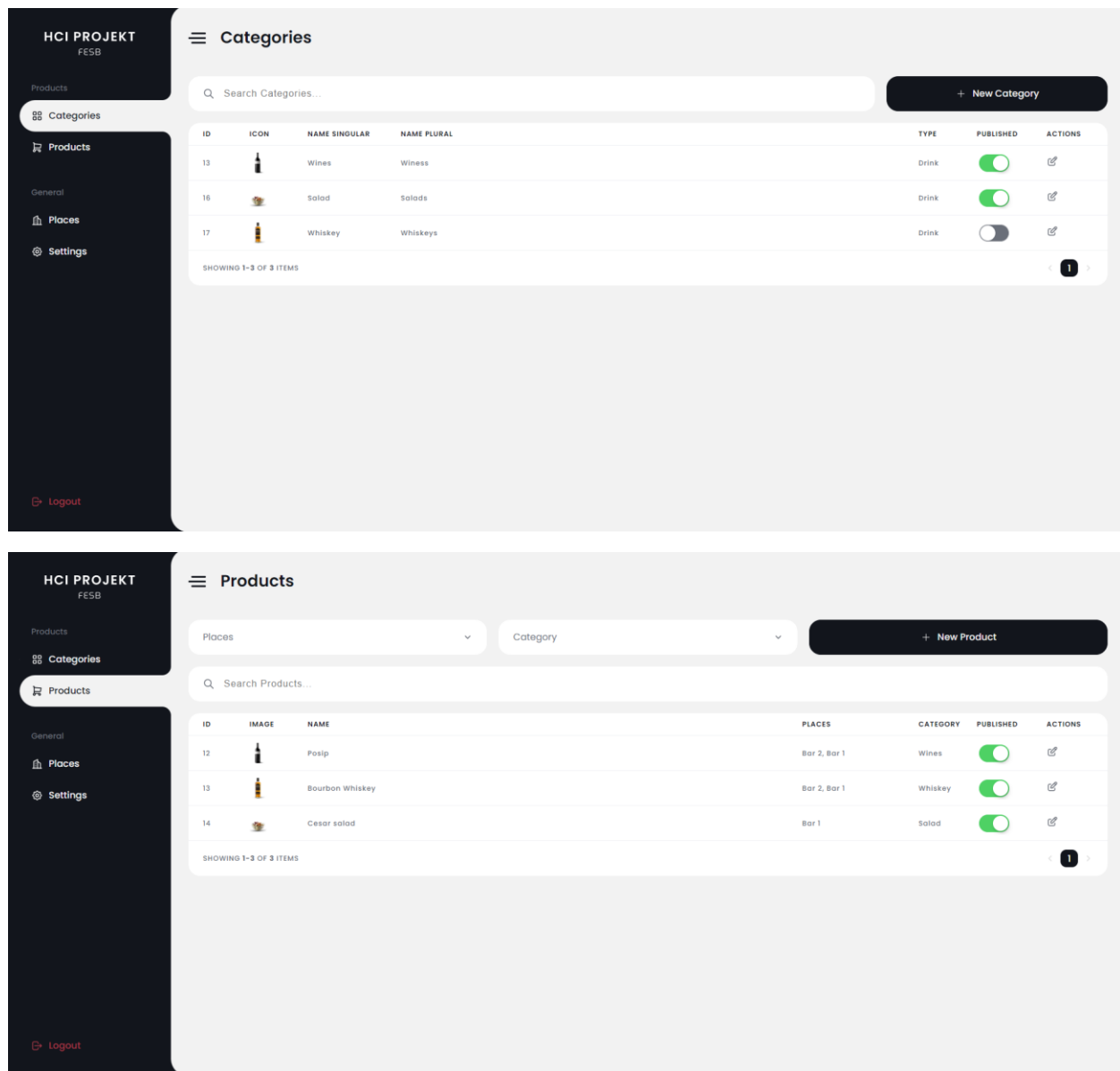


Slika 38 Kontrast na primjeru frontend-a

#### 4.2.2 Repetition

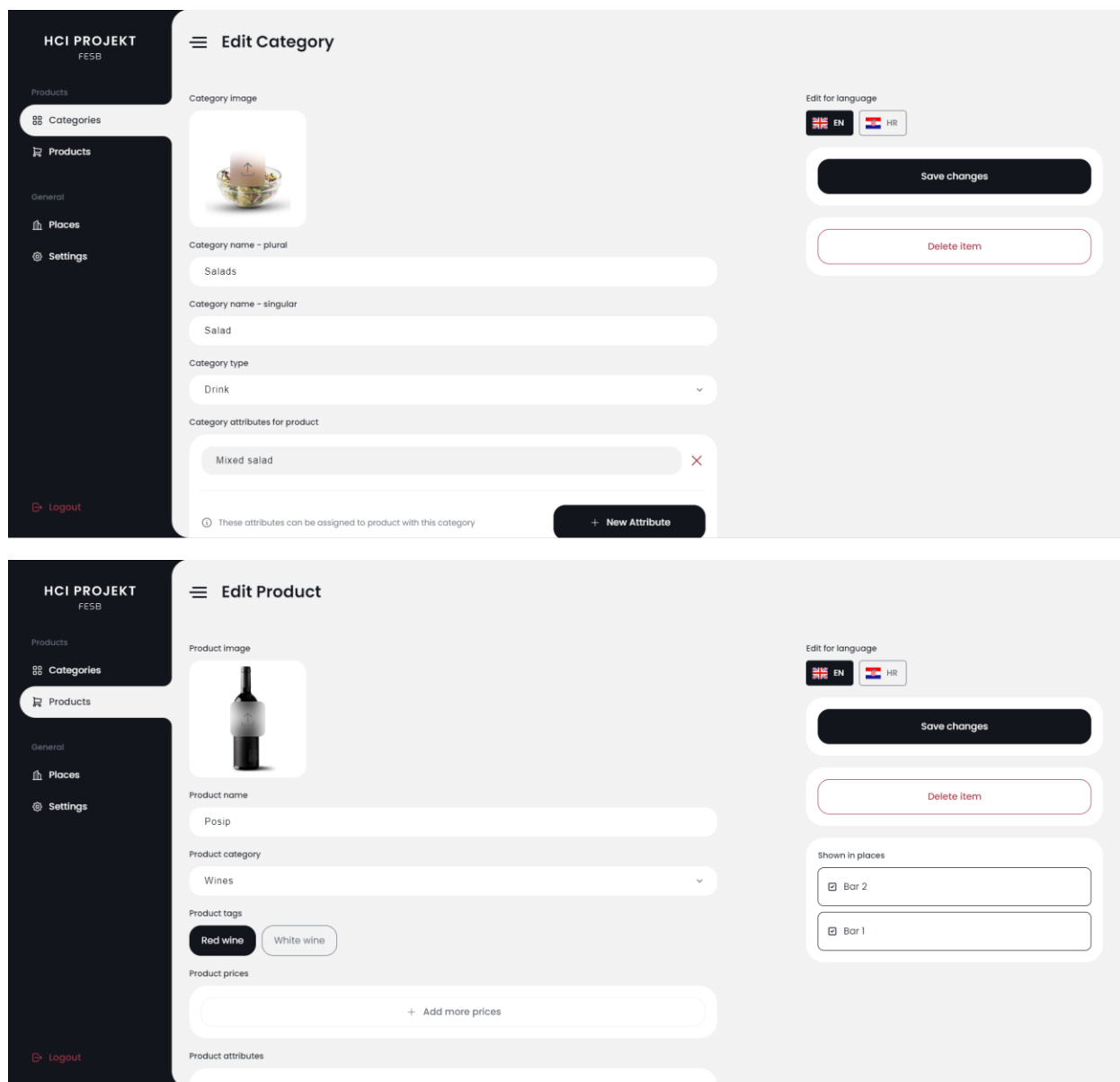
Elementi s istom namjenom ili važnosti trebaju izgledati isto. Ako se dosta različitih elemenata nalazi na stranici, stranica neće izgledati kohezivno.

Princip ponavljanja vidljiv je prolaskom kroz web aplikaciju. Ponavljanje istih elemenata uočljiv je prelaskom sa sekcije na sekciju, odnosno ponavljaju se CTA botuni, stil prikaza naziva sekcije, traka za pretraživanje, prikaz stavki ( tablica ), naslovi, inputi, veličina ikona... Na slikama ispod prikazane su stranice za prikaz kategorija i proizvoda te je njihova sličnost uočljiva



Slika 39 Ponavljanje layout-a

Također stranice za uređivanje proizvoda i kategorija imaju jednak raspored CTA botuna i povezanih input-a, a to je prikazano na slikama ispod. Vidimo da se botuni za brisanje i uređivanje nalaze na istom mjestu, te da je raspored unosa sličan koliko je god to moguće.



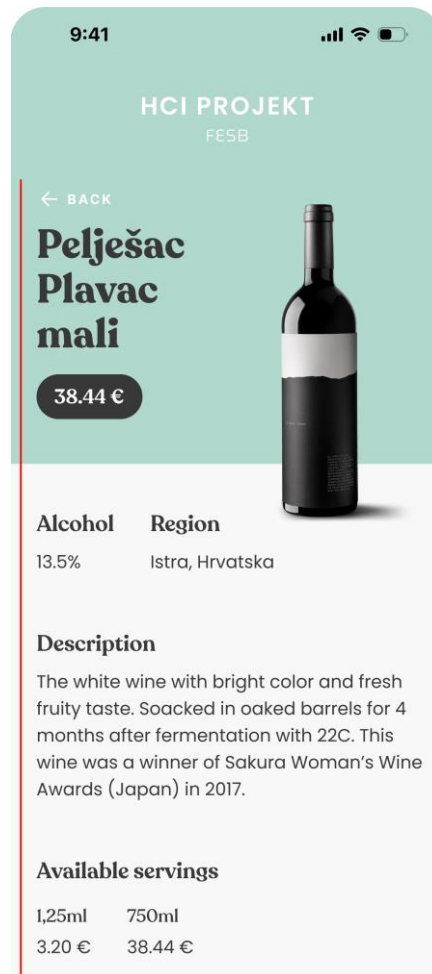
Slika 40 Ponavljanje stranice za uređivanje

#### 4.2.3 Alignment

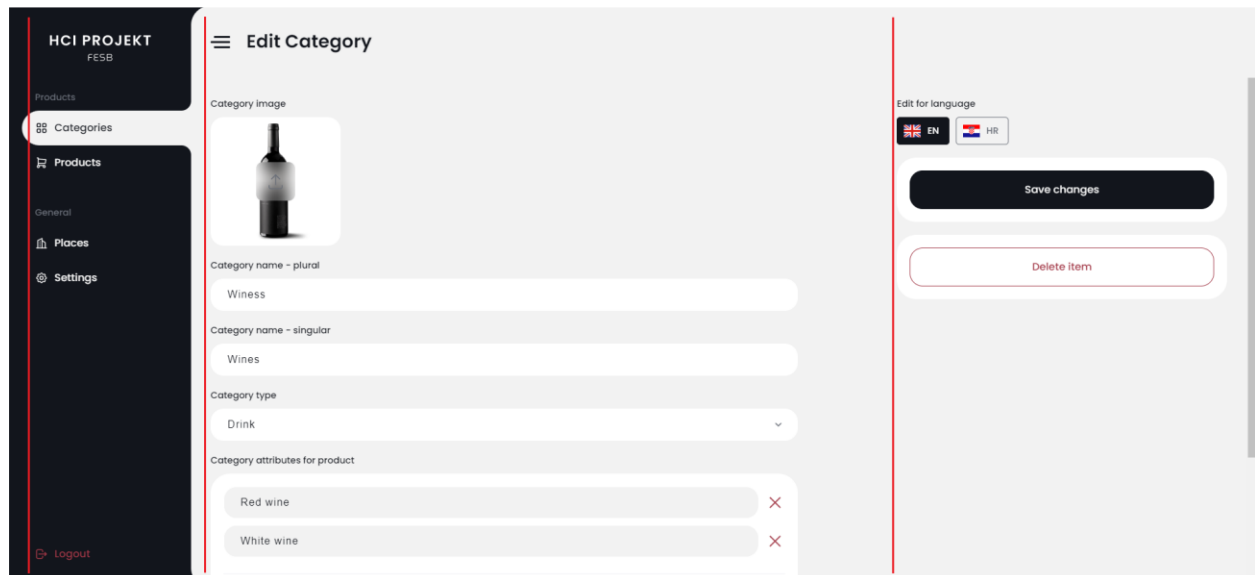
Na stranici ništa ne bi smjelo stajati proizvoljno, svaki element trebao bi biti povezan s nečim drugim. Čak i kada su poravnati elementi fizički razdvojeni jedan od drugog, postoji nevidljiva linija koja ih povezuje u oku i umu.

U web aplikaciji poravnanje se primjenjuje na svim stranicama, prozorima i ostalim elementima. Primjer poravnanja prikazan je na slikama ispod. Aplikacija je jednostavna za čitanje i sa ovim principom ostvarili smo da korisnikovo oko ne bježi po stranici jer su elementi poravnati. Tanka crvena linija je dodana kako bi se pokazalo lijevo poravnanje na stranici.





Slika 41 Primjer korištenja principa "Poravnanje"



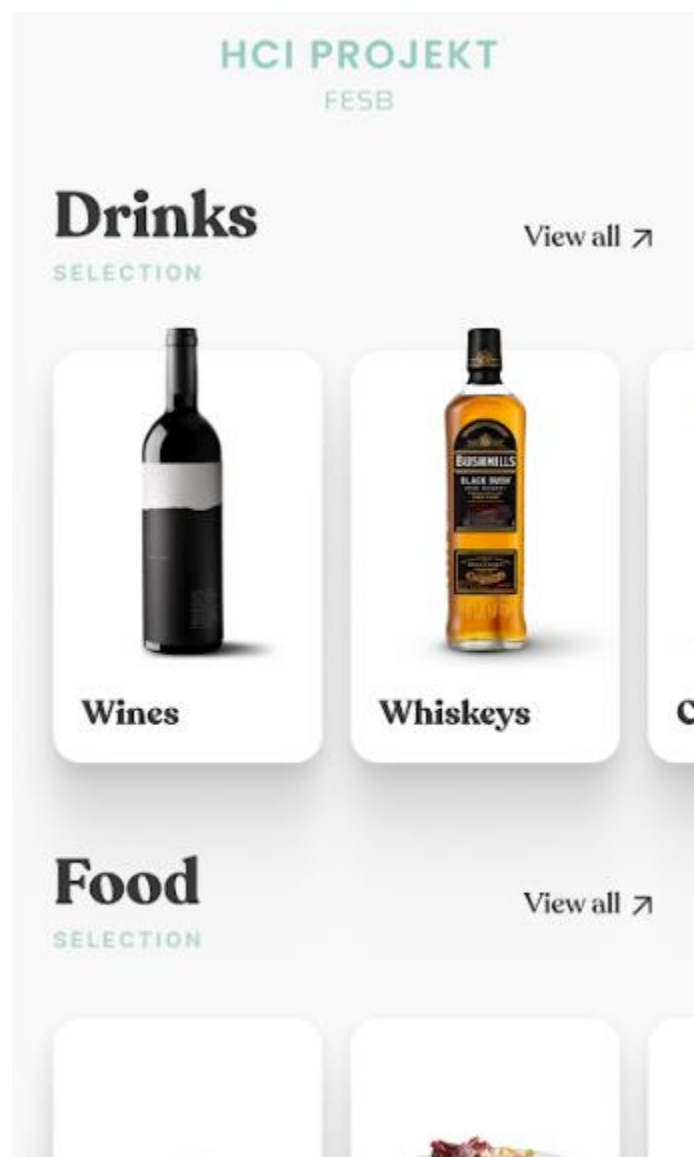
Slika 42 2. primjer korištenja principa "Poravnanje"

#### 4.2.4 Proximity

Princip proximity ( bliskost ) odnosi se na grupiranost elemenata. Elementi koji su povezani trebali bi biti blizu jedan drugog, a oni koji nisu povezani razdvojeni. Dizajner se ne bi trebao bojati praznog prostora. Na ovakav način stvaraju se vizualne jedinice i korisničko iskustvo je bolje. Broj vizualnih jedinica bi trebao biti razuman, ne valja da ih je premalo, a ne valja ni da ih je previše.

Postoji više primjera korištenja ovog principa u ovoj web aplikaciji. Prvi bi bio odvajanje side bar-a na dashboard-u od ostatka sadržaja. Na taj način korisnik zna koji mu je statični, a koji dinamični dio stranice. Na taj način dobili smo dvije vizualne jedinice, jedna za statički dio ( side bar ), jedna za dinamički dio ( sadržaj stranice ).

Drugi primjer bi bio odvajanje kategorija po tipovima na „frontend“ dijelu. Kategorije su odvojene ovisno o tome kojem tipu pripadaju, također slika i naziv pojedine kategorije su odvojeni.



Slika 43 "Proximity" kod prikaza kategorija

Još jedan primjer korištenja ovog principa uočljiv je i prilikom dodavanja/uređivanja kategorija/proizvoda. Polja za unos grupirana su ovisno o podatku koji se traži.

The image shows a user interface for managing product information, organized into three main sections:

- Product tags:** Contains two buttons, "Red wine" (highlighted in dark) and "White wine" (outlined).
- Product prices:** A horizontal bar containing four input fields: a bottle icon with a dropdown arrow, "0.75l", "350€", and "Bar 2" with a dropdown arrow. A red "X" icon is at the end of the bar. Below this bar is a button labeled "+ Add more prices".
- Product attributes:** A section with a table-like structure:
  - Label:** A text input field containing "Region".
  - Type:** A dropdown menu currently showing "Text".
  - Content:** A text input field containing "Croatia".
  - Remove attribute:** A red text link below the content field.

At the bottom of the "Product attributes" section, there is a note: "These fields will be shown in the product details" with an information icon, and a dark button labeled "+ New Field".

Slika 44 3. primjer "Proximity" principa

## 4.3 Teorije niske razine

Teorije koje pokušavaju predvidjeti ljudsko ponašanje u interakciji sa stranicom. Najpoznatije su Fittsov (Fitts' law) te Hickov (Hick's law) zakon.




### 4.3.1 Fittsov zakon

Fittsov zakon kaže da je vrijeme potrebno da osoba pomakne pokazivač (npr. kursor miša) na ciljno područje funkcija udaljenosti do cilja podijeljene s veličinom cilja. Dakle, što je veća udaljenost i manja veličina mete, potrebno je više vremena. Dakle dizajner bi prilikom dizajna stranice za neku akciju trebao uzeti u obzir korisnikov trenutni položaj miša u odnosu na sljedeći klik korisnika. Smatra se da rubovi ekrana sadrže beskonačno dugu širinu tako da možemo zaključiti kako su rubovi najpristupačnija područja na zaslonu.

Primjer dobrog korištenja ovog zakona je botun za spremanje promjena prilikom dodavanja/uređivanja proizvoda. Botun je dovoljno velik i nalazi se na rubu ekrana. Također smatra se da će korisnik unositi podatke odozgo prema dolje s lijeva na desno, iz tog razloga je pozicioniran na desnom dijelu ekrana. U ovom primjeru bi posljednja izmjena korisnika bila biranje mjesta gdje će se proizvod prikazati.

Slika 45 Primjer korištenja "Fittsovog" zakona

Također ispravka jedne stavke iz high-fidelity prototipa po ovom zakonu bi bilo polje za uređivanje i brisanje unutar tablice. Naime ikonice su premale i preblizu jedna drugoj te postoji mogućnost da korisnik klikne na jednu, a ustvari želi odabrati drugu.

ID	ICON	NAME SINGULAR	NAME PLURAL	TYPE	PUBLISHED	ACTIONS
8034		Whiskey	Whiskeys	Drink	<input checked="" type="checkbox"/>	 

Slika 46 Loš UX po Fittsovom zakonu

Ovaj problem riješen je prilikom implementacije aplikacije na način da je u stupcu „Actions“ ostala samo ikonica za uređivanje, a ikonica za brisanje je izbačena. Akcija za brisanje je prebačena unutar stranice za uređivanje.

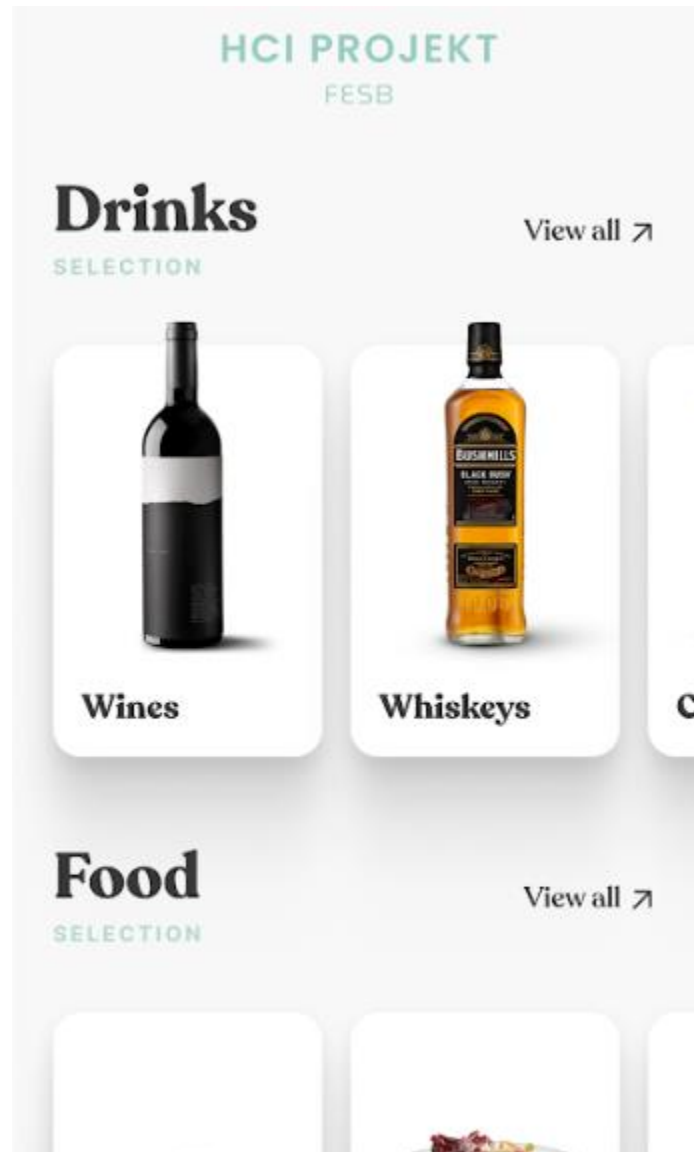
ID	IMAGE	NAME	PLACES	CATEGORY	PUBLISHED	ACTIONS
12		Posip	Bar 2, Bar 1	Wines	<input checked="" type="checkbox"/>	

Slika 47 Evaluacija po Fittsovom zakonu

#### 4.3.2 Hickov zakon






















Hickov zakon predstavlja olakšavanje korisniku odluke odabira od više ponuđenih elemenata. Prividnim smanjenjem ponuđenih opcija na zaslonu, sučelje djeluje prihvatljivije korisniku. Hickov govori da što je ponuda veća, korisniku treba više vremena za izbor, a time se pogoršava korisničko iskustvo.

Najočitiji primjer primjene ovog zakona je na početnoj stranici aplikacije gdje su korisniku izbačene neke kategorije, a klikom na „View All“ opciju otvara se lista svih kategorija.



*Slika 48 Primjer korištenja Hickovog zakona*

Još jedan dobar primjer korištenja Hickovog zakona bi bila tablica za prikazivanje proizvoda. Kako se korisnika ne bi zatrpalo sa velikom listom proizvoda, napravljena je paginacija.

ID 	ICON	NAME SINGULAR	NAME PLURAL	TYPE	PUBLISHED	ACTIONS
8034		Whiskey	Whiskeys	Drink	<input checked="" type="checkbox"/>	
7483		Wine	Wines	Drink	<input checked="" type="checkbox"/>	
9182		Cocktail	Cocktails	Drink	<input checked="" type="checkbox"/>	
8034		Whiskey	Whiskeys	Drink	<input checked="" type="checkbox"/>	
7483		Wine	Wines	Drink	<input checked="" type="checkbox"/>	
9182		Cocktail	Cocktails	Drink	<input checked="" type="checkbox"/>	
8034		Whiskey	Whiskeys	Drink	<input type="checkbox"/>	
7483		Wine	Wines	Drink	<input checked="" type="checkbox"/>	
9182		Cocktail	Cocktails	Drink	<input checked="" type="checkbox"/>	
SHOWING 1-8 OF 18 ITEMS					 <b>1</b> 2 3 	

Slika 49 2. primjer korištenja Hickovog zakona

## 6. Implementacija prototipa

Nakon evaluacije dizajna i donošenja potrebnih promjena na osnovu heuristika, principa i teorija niskih razina, prototip je implementiran. REST API napravljen je pomoću Adonis.js-a, PostgreSQL i Redis tehnologija, a sama web aplikacija napravljena je pomoću Next.js v13.0.7. Sustav je deploy-an na Ubuntu VPS-u, link se nalazi unutar repozitorija na github-u. ( <https://github.com/jmelvan/HCI/> )

## 7. Zaključak

Izrada prototipa aplikacije omogućava lakše prenošenje zamisli i slaganje samog sustava. Samo stvaranje dizajna i prototipa aplikacije predstavlja veliki izazov jer sve zamisli i stavke sustava potrebno je prezentirati korisniku kroz sučelje. Loša prezentacija proizvoda može uzrokovati ne prihvaćanje ideje koja je dobra. Također kasnije u implementaciji lakše je razviti i strukturirati kod aplikacije kada postoji gotov dizajn.

Evaluacijom prototipa kroz heuristike, principe i zakone ispravljaju se greške u dizajnu same aplikacije i aplikacija dobiva bolje korisničko iskustvo koje je jako važan faktor u uspjehu same aplikacije. Prilikom dizajna aplikacije treba biti objektivan i prihvatiti greške koje smo uočili prilikom evaluacije prototipa i poboljšati sami dizajn u odnosu na njih. Nekad male stvari znače puno za korisničko iskustvo.

Dizajn aplikacije treba biti namijenjen korištenju stvarnih korisnika, a ne samo osobama koje su razvijali aplikaciju. Postizanje takvog dizajna kod malo složenijih sustava ne bi bilo moguće bez da se prate koraci evaluacije prototipa.