

Univerzitet „Džemal Bijedić“ u Mostaru,
Fakultet informacijskih tehnologija, Softverski inženjering

Testiranje web aplikacije „Online Bingo prodavnica“

PREDMET: FORMALNE METODE

Autori: Lamiya Babović IB230398,
Mustafa Isić IB190022,
Ismihana Međedović IB190022,
Inas Bajrektarević IB200007

Asistent: mr. Ahmet Mulalić

Uvod

U ovom seminarskom radu ćemo testirati funkcionalnosti odabrane web aplikacije. Fokus naše analize biti će **Online Bingo prodavnica**. Ovo je aplikacija koja se koristi za obavljanje online kupovine. S obzirom na sklonost ovog tipa aplikacija greškama koje mogu značajno uticati na njihov rad, uradit ćemo niz testova kako bi se identificirali potencijalni nedostaci. Kroz ove testove, mi ćemo analizirati ključne aspekte kao što su funkcionalnost, interfejs i sigurnost kako bismo poboljšali ovu web aplikaciju.

Zadatak 1.

Napišite 10 proizvodnih zahtjeva za navedenu web aplikaciju i izvedite 5 funkcionalnih i 5 ne-funkcionalnih zahtjeva iz proizvodnih zahtjeva.

Zahtjevi su stanje ili sposobnost koja je potrebna korisniku za rješavanje problema ili postizanje cilja. Sa zahtjevima je najbolje raditi prije početka razvoja.

Proizvodni zahtjevi su funkcije i ograničenja za rješenje koje treba razviti (npr. softverski sistem). Oni postavljaju ograničenja koja rješenja mogu i smiju koristiti.

Funkcionalni zahtjevi obično opisuju ono što sistem ili aplikacija treba raditi – u njima se navode njegove funkcionalnosti. To su specifični zadaci, akcije ili usluge koje rješenje mora pružiti. Na primjer, kod online prodavnice, funkcionalni zahtjevi mogu uključivati kreiranje korisničkog računa, pretragu proizvoda, dodavanje proizvoda u korpu, plaćanje itd..

Nefunkcionalni zahtjevi opisuju kako sistem treba raditi ili pružati funkcionalnosti. To su karakteristike sistema koje nisu direktno povezane s određenim zadacima ili funkcionalnostima. Nefunkcionalni zahtjevi mogu obuhvatati performanse, sigurnost, pouzdanost, skalabilnost, upotrebljivost, održivost itd.. Na primjer, za online prodavnicu, nefunkcionalni zahtjevi mogu uključivati brzinu učitavanja stranica, zaštitu korisničkih podataka, dostupnost servisa 24/7 itd..

Proizvodni zahtjevi:

1. Kao eksterni korisnik želim imati mogućnost registracije vlastitog računa.
2. Kao eksterni korisnik želim imati mogućnost pregleda proizvoda po kategorijama, te cijenama.
3. Kao eksterni korisnik želim imati mogućnost pretraživanja proizvoda po nazivu.
4. Kao eksterni korisnik želim imati uvid u detalje proizvoda.
5. Kao eksterni korisnik želim imati mogućnost dodavanja proizvoda u korpu.
6. Kao eksterni korisnik želim imati mogućnost dodavanja proizvoda na listu želja.
7. Kao eksterni korisnik želim imati mogućnost usporedbe proizvoda.
8. Kao eksterni korisnik želim imati uvid u listu proizvoda dodanih u korpu.
9. Kao eksterni korisnik želim imati mogućnost kupovine tj. plaćanja online.
10. Kao eksterni korisnik želim imati mogućnost ostavljanja recenzije na proizvod.

Funkcionalni zahtjevi:

1. Kao eksterni korisnik želim imati mogućnost kreiranja vlastitog računa. (Kao eksterni korisnik želim imati mogućnost upravljanja vlastitim računom).
2. Kao eksterni korisnik želim imati mogućnost pretraživanja proizvoda.
3. Kao eksterni korisnik želim imati mogućnost kupovine, online putem.
4. Kao eksterni korisnik želim imati mogućnost pregleda historije svojih narudžbi.

5. Kao eksterni korisnik želim imati mogućnost kontaktiranja korisničke podrške putem telefona.
6. Kao eksterni korisnik želim imati mogućnost filtriranja cijena proizvoda određene kategorije
7. Kao eksterni korisnik želim imati mogućnost slanja zahtjeva za dostavu na adresu.

Nefunkcionalni zahtjevi:

1. Učitavanje stranice se treba izvršavati zadovoljavajućom brzinom.
2. Stranica treba raditi neometano i u situacijama povećanog broja korisnika.
3. Stranica treba biti dostupna 24/7.
4. Stranica treba biti kompatibilna odnosno treba biti dostupna na različitim web browser-ima.
5. Stranica treba da bude intuitivna i jednostavna za korištenje svim korisnicima.

Zadatak 2.

Izaberite određenu funkcionalnost na web aplikaciji (osim “Log In” i “Registration/Sign Up” funkcionalnosti) i istu testirajte korištenjem svake od ispod navedenih tehnika testiranja (slijedite uputstva za svaku tehniku). Ako navedenu funkcionalnost ne možete testirati korištenjem određene tehnike testiranja, izaberite po svojoj volji neku drugu funkcionalnost na istoj web aplikaciji i primijenite navedenu tehniku testiranja na izabranu funkcionalnost. Za svaku tehniku testiranja objasnite šta ste uradili i zašto.

1. Ekvivalentnost particioniranja (Equivalence Partitioning)

Uradite tabelu parametara, particija, tip particija (važeće/nevažeće) i ulaznih vrijednosti

Ekvivalentna particija dijeli podatke na particije (također poznate i kao klase ekvivalencije) na način da se očekuje da će svi članovi date particije biti obrađeni na isti način. Postoje particije ekvivalencije i za važeće i za nevažeće vrijednosti. Važeće vrijednosti su vrijednosti koje komponenta ili sistem treba prihvatiti dok nevažeće vrijednosti su vrijednosti koje komponenta ili sistem treba odbiti.

Ovom tehnikom ćemo pokriti sve moguće particije za određeno polje, zatim ćemo navesti jednu ulaznu vrijednost koja odgovara toj particiji i navest ćemo da li je ona važeća ili nevažeća. Da bi se postigla 100% pokrivenost ovom tehnikom, test slučajevi moraju pokriti sve identificirane particije (uključujući nevažeće particije) korištenjem najmanje jedne vrijednosti iz svake particije. Dobar primjer funkcionalnosti za ovu tehniku su polja za unos koji nam omogućavaju filtriranje cijena proizvoda određene kategorije. Na tabelama ispod možemo vidjeti kakva su ograničenja polja za unošenje filtera za navedene parametre.

Parametar	Particija ekvivalencije	Važeća/nevažeća	Ulazna vrijednost
Polje za unos najmanje cijene	Vrijednost <= 922337203685477	Nevažeća	-
	Vrijednost > 922337203685477	Nevažeća	922337203685499
	-922337203685477 -922337203685477	Važeća	155
	Vrijednost non int	Važeća	7.55

Parametar	Particija ekvivalencije	Važeća/nevažeća	Ulazna vrijednost
Polje za unos najveće cijene	Vrijednost < - 922337203685480	Navažeća	-9,22337E+14
	Vrijednost > 922337203685477	Navažeća	922337203689999
	-922337203685477 - 922337203685477	Važeća	169
	Vrijednost non int	Važeća	11.25

2. Analiza granične vrijednosti (Boundary Value Analysis)

Uradite tabelu parametara, particije ekvivalencije i graničnih vrijednosti

Analiza granične vrijednosti (BVA) produžetak je ekvivalentnosti particioniranja, ali se može koristiti samo kada je particija naručena, a sastoji se od numeričkih ili sekvencijalnih podataka. Granične vrijednosti su minimalne i maksimalne vrijednosti (ili prva i zadnja vrijednost) particije.

Koristit ćemo istu funkcionalnost i uraditi analizu granične vrijednosti. Već su nam poznate važeće particije ekvivalencije za parametre iz prethodnog testa, a sada ćemo odrediti koje su važeće i nevažeće granice za unos tih parametara.

Parametar	Particija ekvivalencije	Granične vrijednost
Polje za unos najmanje cijene	-922337203685477 922337203685477	-55(v), 899(v), -922337203685478(n), 992337203685477(n)
Polje za unos najveće cijene	-922337203685477 922337203685477	105.5(v), 133(v), -922337203685998(n), 999999999999999(n)

3. Testiranje tabele odluka (Decision Table Testing)

Uradite tabelu odluka sa unosima i radnjama

Tabele odlučivanja (Decision Tables) su grafički prikazi logičkih pravila i uslova koji vode ka određenim akcijama ili rezultatima. Ove tabele pomažu u testiranju složenih sistema gdje postoji veliki broj mogućih ulaza i kombinacija uslova.

Testiranje tabela odluka (Decision Table Testing) je proces provjere ispravnosti sistema ili aplikacije u skladu sa definisanim poslovnim pravilima predstavljenim u tabeli odlučivanja. Ovaj proces uključuje identifikaciju i provjeru različitih kombinacija ulaza kako bi se osiguralo da sistem ispravno reaguje na različite scenarije.

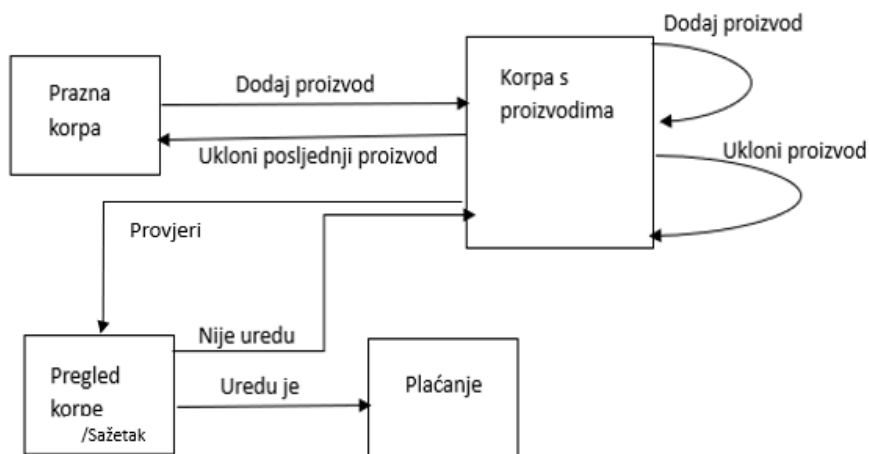
Tabela odluke za testiranje funkcionalnosti besplatne dostave na ovoj stranici ima za cilj proveru ispravnosti sistema u skladu s definisanim poslovnim pravilima. Ova stranica nudi besplatnu dostavu na adresu koju korisnik unese. Da bi dobili besplatnu dostavu, iznos korisnikove narudžbe treba biti 100KM ili više i treba popuniti formu za dostavu. Cilj nam je pokriti sve moguće slučajeve i navesti njihov ishod. Kao uslove ćemo navesti da je iznos narudžbe veći od granice koja određuje besplatnu dostavu i da li je korisnik popunio formu za dostavu. Zatim ćemo navesti sve moguće slučajeve. Ispod ćemo za svaki slučaj navesti ishod to jeste da li će korisnik dobiti besplatnu dostavu.

Uslov	Slučaj 1	Slučaj 2	Slučaj 3	Slučaj 4
Iznos narudžbe je >100	T	F	T	F
Popunjena forma za dostavu?	T	F	F	T
Radnja				
Besplatna dostava	Besplatna dostava	Nije besplatna dostava (nema dostave)	Nije besplatna dostava (nema dostave)	Dostava nije besplatna (ima dostave)

4. Testiranje tranzicije stanja (State Transition Testing) Nacrtajte dijagram tranzicije stanja i navedite sve testne slučajeve

Testiranje tranzicije stanja (State Transition Testing) je tehnika testiranja koja se koristi za provjeru ispravnosti sistema koji mijenja svoje stanje u zavisnosti od različitih događaja. Ova tehnika fokusira se na identifikaciju i testiranje prelaza između različitih stanja sistema. Stanja sistema predstavljaju određene uslove ili situacije u kojima se sistem može nalaziti.

Korpa za kupovinu na online Bingo prodavnici počinje kao prazna. Kada su proizvodi odabrani, dodaju se u korpu za kupovinu. Artikli se također mogu ukloniti iz korpe za kupovinu. Kada kupac želi provjeriti korpu, prikazat će mu se stanje korpe i ukupni trošak, kako bi kupac mogao provjeriti je li to uredu ili ne. Ako mu je sve uredu nastavlja dalje na plaćanje. U suprotnom se vraća na kupovinu (tako da može dodati ili ukloniti proizvode ukoliko to želi).



Zatim ćemo navesti sve testne slučajeve koji su navedeni na dijagramu. Ovaj dijagram i povezani testni slučajevi pokrivaju različite korake u procesu kupovine, od dodavanja proizvoda u korpu do provjere korpe i odluke o nastavku na plaćanje ili povratku na kupovinu, ili brisanja produkta iz korpe. Možemo primjetiti da nam nisu dostupne sve akcije na svakom stanju.

Stanje	Prazna korpa	Korpa s proizvodima	Pregled korpe /Sažetak	Plaćanje
Dodaj proizvod	Korpa s proizvodima	Korpa s proizvodima	/	/
Ukloni proizvod	/	Korpa s proizvodima	/	/
Ukloni posljednji proizvod	/	Prazna korpa	/	/
Provjeri	/	Pregled korpe /Sažetak	/	/
Nije uredno	/	/	Korpa s proizvodima	/
Uredno je	/	/	Plaćanje	/

5. Testiranje izjava (statements) i pokrivenost

Za navedenu/izabranu funkcionalnost napišite pseudo kod ili nacrtajte kontrolni dijagram toka i navedite sve testne slučajeve sa prikladnim ulaznim vrijednostima

U ovoj tehnici potrebno je kreirati testne slučajeve koji pokrivaju unaprijed definirani procenat iskaza. Dakle, potrebno je definirati najmanji broj putanja koje je potrebno preći da bi se pokrili svi iskazi. Pokrivenost se definira kao $\frac{\text{broj izvršenih iskaza}}{\text{ukupni broj iskaza}}$.

Često se za potrebe ove tehnike kreira i dijagram toka na osnovu izvornog koda.

U nastavku je prikazan pseudo kod za dio aplikacije koji se tiče filtriranja proizvoda na osnovu cijena te tabela sa tesnim slučajevima. (Za potrebe zadatka 6 je kreiran i dijagram toka).

```

read min_price;
read max_price;

if (!isNumber(min_price) || !isNumber(max_price))
    print(Error_WrongType);
else if (!isValidValue(min_price) or !isValidValue(max_price))
    print(Error_InvalidValue);
else if (min_price>max_price)
    set max_price=min_price;
    print (result_list);
else print (result_list);|

```

Test case 1:	min_price="a" max_price="b"
--------------	--------------------------------

Test case 2:	min_price="-123456" max_price="12345789123465784"
Test case 3:	min_price="10000" max_price="100"
Test case 4:	min_price="100" max_price="200"

Kroz prethodna 4 gore navedena testna slučaja postiže se potpuna pokrivenost izjava, što znači da su svi dijelovi koda izvršeni tokom izvođenja testova.

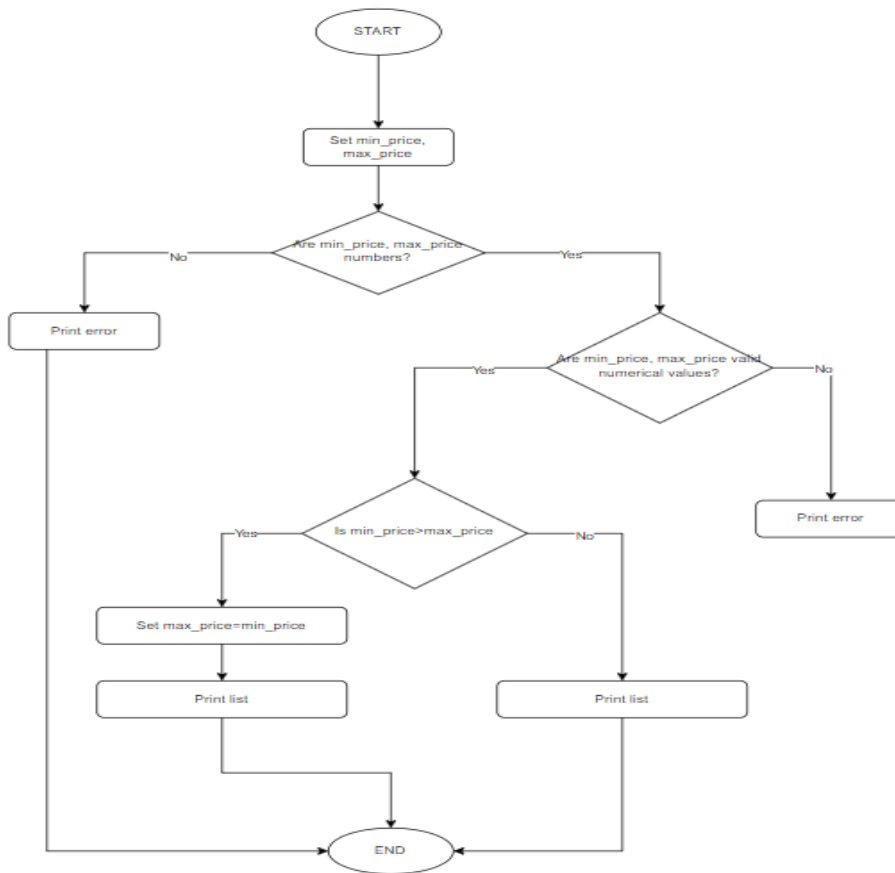
6. Testiranje odluka (decisions) i pokrivenost

Za navedenu/izabranu funkcionalnost napišite pseudo kod ili nacrtajte kontrolni dijagram toka i navedite sve testne slučajeve sa prikladnim ulaznim vrijednostima

Testiranje odluka je tehnika slična onoj u prethodnom zadatku, međutim ova tehnika se smatra jačom odnosno 100% pokrivenosti odluka znači i 100% pokrivenosti iskaza ali ovo ne mora vrijediti i u obratnom slučaju.

Potrebno je definirati testne slučajeve koji će ispuniti definirani procetan pokrivenosti. U ovom slučaju pokrivenost se definira kao $\frac{\text{broj izvršenih odluka}}{\text{ukupni broj odluka}}$.

U nastavku se nalazi dijagram toka i tabela sa tesnim slučajevima.



Test case 1:	min_price="tekst" max_price="abc"
Test case 2:	min_price="987654321987654321" max_price="-1000"
Test case 3:	min_price="123456" max_price="0"
Test case 4:	min_price="10000" max_price="20000"

7. Pogađanje pogreške (Error guessing)

Nabrojite sve moguće defekte i dizajnirajte testove koji “napadaju” te defekte

Pogađanje pogreške" (Error guessing) je tehnika testiranja koja se oslanja na iskustvo i znanje testera kako bi identifikovala moguće defekte u softveru. Umjesto da slijedi formalne procedure ili specifične test planove, tester koristi svoje razumjevanje sistema kako bi predvidio gdje se mogu pojaviti problemi. Strukturirani pristup je nabrojati popis mogućih defekata (bugova) i dizajnirati testove koji napadaju te defekte (bugove).

Ovaj sistematski pristup naziva se napadom na grešku (fault attack). Ove liste defekata (bugova) i kvarova mogu se sastaviti na osnovu: iskustva dostupnih podataka o defektima (bugovima) i kvarovima općepoznatih razloga zašto softver otkazuje (se kvari).

POTENCIJALNA GREŠKA	TESTNI SLUČAJ
Moguće unijeti negativnu vrijednost u filter	Unesi negativnu vrijednost u polje za minimalnu cijenu
Moguće je unijeti karakter u polje predviđeno za unos cijene	Unesi slovo i specijalni karakter u polje predviđeno za maksimalnu cijenu
Moguće je unijeti početnu vrijednost intervala veću od krajnje	Unesi obje vrijednosti filtera s tim da je početna vrijednost veća od krajnje
Sistem ne prikazuje odgovarajući raspon cijene koji je naveden u filteru	Otići na pretragu proizvoda, idi na filter po cijeni i postavi raspon cijene od 400-800 KM
Sistem ima probleme prilikom čitanja ili upita prema bazi podataka koji dovode do netačnih rezultata kada se unese raspon cijene.	Provjeri funkcionalnosti prilikom simuliranja problema s bazom. Unijeti realan interval cijena koje postoje, a koje nisu najmanja i najveća cijena.
Sistem vraća sve artikle koji su u bazi, ne uzima filter u obzir	Unijeti realan interval cijena koje postoje, a koje nisu najmanja i najveća cijena.
Sistem ima problema sa tumačenjem decimalnih mjesta (zarez ili tačka)	Unos različitih formata decimalnih mjesta 1.50 i 1,50 i provjeriti kako sistem obrađuje ove vrijednosti.

8. Istraživačko testiranje (Exploratory testing)

Objasnite kako bi ste organizovali istraživačko testiranje za navedenu funkcionalnost, koji tip istraživačkog testiranja bi ste koristili i zašto.

Istraživačko testiranje je tip testiranja koji se bazira na iskustvu testera. Ovaj tip testiranja najviše koristimo kada postoji vremenski pritisak odnosno kratak rok za izvršavanje testiranja, imamo lošu specifikaciju ili naš tim nije obučen za testiranje ali ima dobro znanje iz drugih domena. Ne postoji predefinirana procedura, plan ili struktura koja se slijedi prilikom izvršavanja testova.

Osnovni tipovi istraživačkog testiranja su:

-freestyle testiranje (bazirano na slobodnom stilu)

-testiranje bazirano na scenariju

-testiranje bazirano na strategiji

Za konkretnu funkcionalnost filtriranja cijena mogli bismo iskoristiti testiranje bazirano na scenariju. U tom slučaju se možemo staviti u poziciju krajnjeg korisnika te isprobati različite scenarije. Neki od njih su filtriranje bez unosa odnosno po default-nim vrijednostima. Dalje možemo testirati mogućnosti sa nekim manjim rasponima cijena, potom možemo unijeti prevelike i premale vrijednosti. Kao krajnji korisnik mogli bi kombinovati filtriranje cijene sa drugim filterima ili funkcijama pretrage. Na primjer mogli bi filtrirati po cijeni i to kombinirati sa filtriranjem po kategorijama proizvoda. Možemo provjeriti kako se ovaj filter ponaša u različitim browserima.

Testirati kako se funkcionalnost ponaša na različitim uređajima, uključujući računare, laptope, tablete i mobilne telefone. Time bi provjerili odziv i izgled na različitim veličinama ekrana i uređajima sa različitim karakteristikama.

Istražiti kako se korisnici samostalno snalaze korištenjem ove funkcionalnosti, može pružiti dodatne uvide.

Mogli bi i provjeriti kako se sistem nosi sa lošim unosima, nevalidnim.

Korištenjem testiranja bazirano na strategiji mogli bi razmotriti nekoliko aspekata. Mogli bi definisati strategiju za testiranje performanse ove funkcionalnosti. To bi uključivalo procjenu brzine filtriranja i odazivom sistema pod različitim uslovima.

Mogli bi razviti strategiju za procjenu skalabilnosti. Testirali bi kako se sistem ponaša sa rastućim brojem korisnika ili artikala u bazi podataka. Zatim simulirati više korisnika koji istovremeno koriste funkcionalnost filtriranja cijena kako bismo provjerili reaktivnost sistema ili eventualne konflikte.

Strategija za testiranje kako se Funkcionalnost ponaša prilikom ažuriranja softvera.

Zadatak 3.

Napišite najmanje 10 različitih testnih slučajeva na niskom nivou (korištenjem šablona za pisanje testnih slučajeva na niskom nivou sa predavanja).

Test slučaj na niskom nivou 1

Requirement ID									
Test Case ID		1	Test Case Description		Valid search using existing item				
Created By		Lamija Babovic	Reviewed By			Version		0.1	
QA Tester's Log									
Tester's Name		Lamija Babovic	Date Tested		12.1.2024	Test Case (Pass/Fail/Not Exec)		Pass	
S #		Prerequisites:			S #		Test Data		
1		Install and open web browse (Google Chrome)			1		search value: "kolica"		
2		Open onlineshop.ba web application			2				
3					3				
4					4				
Test Scenario									
Step #		Step Details		Expected Results		Actual Results		Pass / Fail / Not Executed / Suspended	
1		Click the field with "Upišite artikal za pretragu..." placeholder		Cursor is shown in the field		Cursor is dispalyed in the search field		Pass	
2		Type in value for search: "kolica"		Field is populated with search value "kolica"		Field is populated with searched value - word "kolica"		Pass	
3		Click button for searh (button with the magnifying glass icon)		New page is loaded with list of products that contain searched value ("kolica") in name		Page is loaded and all products that contain searched value - word "kolica" are listed		Pass	
4									
5									

Test slučaj na niskom nivou 2

[illegible]

Test slučaj na niskom nivou 3

Requirement ID							
Test Case ID		3	Test Case Description		Adding two products to comparison list and verifying it		
Created By		Lamija Babovic	Reviewed By		Version		0.1
QA Tester's Log							
Tester's Name		Lamija Babovic	Date Tested		12.1.2024	Test Case (Pass/Fail/Not Exec)	
						Pass	
S #		Prerequisites:		S #		Test Data	
1		Install and open web browse (Google Chrome)		1		selected category: "Laptop"	
2		Open onlineshop.ba web application		2		selected product 1: "LAPTOP 15,6" ASUS E510MA-EJ951W 8GB/256GB SSD 49882"	
3				3		selected product 2: "LAPTOP 15,6" ASUS X515EA-BQ322W FHD IPS 8GB/512GB I3-115G4	
4				4			
Test Scenario							
Step #		Step Details		Expected Results		Actual Results	
						Pass / Fail / Not Executed / Suspended	
1		Click dropdown menu "Kategorije" and select item "Laptop"		Page with products is loaded. Only products from selected category are displayed		Page is loaded and only laptops are displayed.	
2		Click on one item.		Page with product details and available actions is loaded.		Page is loaded and all details are displayed	
3		Click "Uporedi artikle" button.		Product is added to list. Message "Artikal LAPTOP 15,6" ASUS E510MA-EJ951W 8GB/256GB SSD 49882 je ubacen u listu poređenja" Button "Uporedi artikle" is replaced with "Dodan na listu" link. Numerical indicator next to comparison icon in navigation menu is shown.		Messge is displayed. Button is replaced with link "Dodan na listu" (link to comparison page). Product is displayed in table when the comparison page is open	
4		Click on different item.		Page with product details and available actions is loaded.		Page is loaded and all details are displayed	
5		Click "Uporedi artikle" button.		Product is added to list. Message "Artikal LAPTOP 15,6" ASUS X515EA-BQ322W FHD IPS 8GB/512GB I3-115G4 48090 je ubacen u listu poređenja" Button "Uporedi artikle" is replaced with "Dodan na listu" link. Number next to comparison icon in navigation menu increases.		Messge is displayed. Button is replaced with link "Dodan na listu" (link to comparison page). Product is displayed in table when the comparison page is open	
6		Click comparison icon in navigation menu.		Comparison page is opened. There are two products in the table.		Page is loaded. Products "LAPTOP 15,6" ASUS E510MA-EJ951W 8GB/256GB SSD 49882" and "LAPTOP 15,6" ASUS X515EA-BQ322W FHD IPS 8GB/512GB I3-115G4 48090" are dispalzey in the table	
						Pass	

[illegible]

Requirement ID	5	Test Case Description	User login with a valid email and password		
Test Case ID		Created By	Ismihana Mededović	Reviewed By	Ismihana Mededović
				Version	0.1
QA Tester's Log					
Tester's Name	Ismihana Mededović	Date Tested	13.01.2024	Test Case (Pass/Fail/Not Executed)	Pass
S # Prerequisites:			S # Test Data		
1	Open Google Chrome web browser		1	URL: https://www.onlineshop.ba/	
2	Go to https://www.onlineshop.ba/		2	email address: testFm13@outlook.com	
3	Log in using invalid data		3	password: testiranje@trance13	
4	Click button "Prijava"		4		
Test Scenario					
Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not Executed / Suspended	
1	Click field "Email adresa"	Cursor is in field "Email adresa".	Cursor is in field "Email adresa"	Pass	
2	Type in email address in field "Email adresa"	Field is populated with respective value.	Field is populated with value of email address	Pass	
3	Click field "Lozinka"	Cursor is in field "Lozinka"	Cursor is in field "Lozinka"	Pass	
4	Type in password in field "Lozinka"	Field is populated with hidden respective value. User is successfully logged in and redirected to home page.	The field is populated with a hidden password value. The user is logged in and the home page is loaded. In the upper right corner, we have a	Pass	
5	Click button "Login"			Pass	
6					

Test slučaj na niskom nivou 6

Requirement ID															
Test Case ID		6		Test Case Description		Adding product to the basket									
Created By		Ismihana Međedović		Reviewed By		Ismihana Međedović				Version		0,1			
QA Tester's Log															
Tester's Name		Ismihana Međedović		Date Tested		13.01.2024				Test Case (Pass/Fail)					
S #		Prerequisites:				S #		Test Data							
1		Low level test case 1'IA1				1		URL:https://www.onlineshop.ba/							
2						2		product: Kolica classic set zoomix knorr baby							
3						3									
4						4									
Test Scenario															
Pass / Fail / Not Executed / Suspended															
Step #		Step Details				Expected Results				Actual Results					
1		Click on the product whose name (link) is "Kolica classic set zoomix knorr baby"				A new page is being loaded with product details.									
2		Click button "Dodaj u korpu"				A new page is loaded with the product added to our cart. We have the possibility to view the basket and how much the total price of the order is.									
3															
4															
5															
6															

Test slučaj na niskom nivou 7

[illegible]

Test slučaj na niskom nivou 8

Requirement ID											
Test Case ID		8		Test Case Description		User login with an invalid email and password					
Created By		Mustafa Isić		Reviewed By		Mustafa Isić		Version		0.1	
QA Tester's Log											
Tester's Name		Mustafa Isić		Date Tested		13.01.2024		Test Case (Pass/Fail/Not		Pass	
S #		Prerequisites:		S #		Test Data					
1		Open Google Chrome web browser		1		URL:https://www.onlineshop.ba/					
2		Go to https://www.onlineshop.ba/		2		email address: testFm15@outlook.com					
3		Log in using invalid data		3		password: testiranje@stranice15					
4		Click button "Prijava"		4							
Test Scenario											

Test slučaj na niskom nivou 9

Requirement ID									
Test Case ID		9	Test Case Description Removing a product from the with list						
Created By		Mustafa Isić	Reviewed By		Mustafa Isić	Version			0,1
QA Tester's Log									
Tester's Name		Mustafa Isić	Date Tested		13.01.2024	Test Case (Pass/Fail/Not		Pass	
S #		Prerequisites:			S # Test Data				
1		Low level test case 4'IA1			1				
2					2				
3					3				
4					4				
Test Scenario									
Step #		Step Details		Expected Results		Actual Results		Pass / fail / Not Executed / Suspended	
1		Click on the "X" in the column "Ukloni"		The product "DRUŠTVENA IGRA MEMORING		The product is removed from the wis		Pass	
2									
3									
4									
5									
6									

Test slučaj na niskom nivou 10

Requirement ID											
Test Case ID		10	Test Case Description								
Created By		Mustafa Isić	Reviewed By		Mustafa Isić	Version		0.1			
QA Tester's Log											
Tester's Name		Mustafa Isić	Date Tested		12.1.2024	Test Case (Pass/Fail)		Pass			
S #			Prerequisites:			S #			Test Data		
1			Install and open web browse (Google Chrome)			1			selected category: "Mobiteli"		
2			Open onlineshop.ba web application			2			selected sorting: "Sortiraj po cijeni (najniža - najviša)"		
3						3			selected sorting: "Sortiraj po cijeni (najviša - najniža)"		
4						4					
Test Scenario											
Step #		Step Details		Expected Results		Actual Results		Pass / Fail / Not Executed /			
1		Click dropdown menu "Kategorije"		Show a list of times that are the categories		The dropdown is shown and all the items are loaded		Pass			
2		Select item "Mobitel"		Load the page with the products that belong in the selected category		Page is loaded and all the products from the category "Mobiteli" are displayed.		Pass			
3		Click on dropdown that currently says "Sortiraj po nazivu"		Show a list of items that represent the sorting for the products		The dropdown is shown and all the items are loaded		Pass			
4		Select item "Sortiraj po cijeni (najniža - najviša)"		sorted from lowest priced to highest		products are sorted from lowest to highest"		Pass			
5											
6											

Zadatak 4.

Automatizirajte prethodno napisane testne slučajeve na niskom nivou korištenjem Selenium-a, Python-a i Pytest-a. Svaki testni slučaj mora biti zaseban test i mora imati određenu validaciju (assert) koja će utvrditi da li je test uspješno izvršen ili ne. Nemojte zaboraviti napraviti setup i teardown u jednom file-u za sve vaše testove. U testovima ste dužni iskoristiti sve stvari koje smo prošli na predavanjima. NAPOMENA: Nemojte koristiti apsolutne putanje u XPathu!

Test add to compare list

```
from selenium.webdriver.common.by import By
from selenium.webdriver.support.wait import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC

def test_add_to_compare_list(driver):
    wait = WebDriverWait(driver, timeout=50)

    driver.get("https://onlineshop.ba/")
    driver.maximize_window()

    cookie_consent_button = driver.find_element(By.CLASS_NAME, 'pjsAcceptCookieBarBtn')
    cookie_consent_button.click()

    categories_tuple = (By.XPATH, "//*[text()='Kategorije']")
    categories_field = wait.until(EC.element_to_be_clickable(categories_tuple))
    categories_field.click()

    category_field = driver.find_element(By.XPATH, "//*[text()=' Laptop ']")
    category_field.click()

    product_tuple = (By.XPATH, "//*[id='listlist']/li[1]/div[@class='media']/div[@class='media-left']/a")
    product_field = wait.until(EC.element_to_be_clickable(product_tuple))
    product_field.click()

    add_product_tuple = (By.XPATH, "//*[div[@class='summary entry-summary']/div[@class='action-buttons']/a[@class='add-to-compare-link']")
    add_product_field = wait.until(EC.element_to_be_clickable(add_product_tuple))
    add_product_field.click()

    swal_tuple = (By.CLASS_NAME, 'swal2-container')
    wait.until(EC.visibility_of_element_located(swal_tuple))
    wait.until(EC.invisibility_of_element_located(swal_tuple))

    categories_tuple = driver.find_element(By.XPATH, "//*[text()='Kategorije']")
    categories_field = wait.until(EC.element_to_be_clickable(categories_tuple))
    categories_field.click()

    category_field = driver.find_element(By.XPATH, "//*[text()=' Laptop ']")
    category_field.click()

    product_tuple = (By.XPATH, "//*[id='listlist']/li[2]/div[@class='media']/div[@class='media-left']/a")
    product_field = wait.until(EC.element_to_be_clickable(product_tuple))
    product_field.click()

    add_product_tuple = (By.XPATH, "//*[div[@class='summary entry-summary']/div[@class='action-buttons']/a[@class='add-to-compare-link']")
    add_product_field = wait.until(EC.element_to_be_clickable(add_product_tuple))
    add_product_field.click()

    swal_tuple = (By.CLASS_NAME, 'swal2-container')
    wait.until(EC.visibility_of_element_located(swal_tuple))
    wait.until(EC.invisibility_of_element_located(swal_tuple))

    search_button_tuple = driver.find_element(By.CLASS_NAME, "ec-compare")
    search_button = wait.until(EC.element_to_be_clickable(search_button_tuple))
    search_button.click()

    compare_product_element_tuple = (By.XPATH, "//*[table[@class='table table-compare compare-list']/tbody/tr[1]/td[2]/a[@class='product']/div[@class='product-info']/h3")
    compare_product_text_element = wait.until(EC.visibility_of_element_located(compare_product_element_tuple))
    compare_product_resultat = compare_product_text_element.text

    assert compare_product_resultat == "LAPTOP 15,6\" ASUS E510PA-EJ951W 8GB/256GB SSD 40882"

    compare_second_product_element_tuple = (By.XPATH, "//*[table[@class='table table-compare compare-list']/tbody/tr[1]/td[2]/a[@class='product']/div[@class='product-info']/h3")
    compare_second_product_text_element = wait.until(EC.visibility_of_element_located(compare_second_product_element_tuple))
    compare_second_product_resultat = compare_second_product_text_element.text

    assert compare_second_product_resultat == "LAPTOP 15,6\" ASUS E510PA-EJ951W 8GB/256GB SSD 40882"
```


Test add to wish list

```
from selenium.webdriver.common.by import By
from selenium.webdriver.support.wait import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC

def test_add_to_wishlist(driver):
    wait = WebDriverWait(driver, timeout=60)

    driver.get("https://onlineshop.ba/")
    driver.maximize_window()

    categories_tuple = (By.XPATH, "///a[text()='Kategorije']")
    categories_field = wait.until(EC.element_to_be_clickable(categories_tuple))
    categories_field.click()

    category_field = driver.find_element(By.XPATH, "///a[text()=' Igračke ']")
    category_field.click()

    cookie_consent_button = driver.find_element(By.CLASS_NAME, 'pjAcceptCookieBarBtn')
    cookie_consent_button.click()

    product_tuple = (By.XPATH, "///ul[@id='listList']/li[1]/div[@class='media']/div[@class='media-left']/a")
    product_field = wait.until(EC.element_to_be_clickable(product_tuple))
    product_field.click()

    add_product_tuple = (By.XPATH, "///div[@class='summary entry-summary']/div[@class='action-buttons']/a[@class='add_to_wishlist']")
    add_product_field = wait.until(EC.element_to_be_clickable(add_product_tuple))
    add_product_field.click()

    swal_tuple = (By.CLASS_NAME, 'swal2-container')
    wait.until(EC.visibility_of_element_located(swal_tuple))
    wait.until(EC.invisibility_of_element_located(swal_tuple))

    search_button_tuple = driver.find_element(By.CLASS_NAME, "ec-favorites")
    search_button = wait.until(EC.element_to_be_clickable(search_button_tuple))
    search_button.click()

    wishlist_product_element_touple=(By.XPATH, "///tbody[@class='wishlist_body']/tr/td[@class='product-name']/a")
    wishlist_product_text_element=wait.until(EC.element_to_be_clickable(wishlist_product_element_touple))
    wishlist_product_rezultat = wishlist_product_text_element.text

    assert wishlist_product_rezultat == 'DRUŠTVENA IGRA MEMORING'
```

Test correct search

```
from selenium.webdriver.common.by import By
from selenium.webdriver.support.wait import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC

def test_correct_search(driver):
    wait = WebDriverWait(driver, timeout=60)

    driver.get("https://onlineshop.ba/")
    driver.maximize_window()

    cookie_consent_button = driver.find_element(By.CLASS_NAME, 'pjAcceptCookieBarBtn')
    cookie_consent_button.click()

    search_field_tuple = (By.ID, "search")
    search_field = wait.until(EC.element_to_be_clickable(search_field_tuple))
    search_field.click()
    search_field.clear()
    search_field.send_keys("kolica")
    search_button = driver.find_element(By.CLASS_NAME, "ec-search")
    search_button.click()

    trazen_i_rezultat_text_element_touple=(By.ID, "title")
    trazen_i_rezultat_text_element=wait.until(EC.visibility_of_element_located(trazen_i_rezultat_text_element_touple))
    trazen_i_rezultat = trazen_i_rezultat_text_element.text

    assert trazen_i_rezultat == 'Traženi rezultati: kolica'
```

Test incorrect search

```
from selenium.webdriver.common.by import By
from selenium.webdriver.support.wait import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC

def test_incorrect_search(driver):
    wait = WebDriverWait(driver, timeout=60)

    driver.get("https://onlineshop.ba/")
    driver.maximize_window()

    cookie_consent_button = driver.find_element(By.CLASS_NAME, 'pjAcceptCookieBarBtn')
    cookie_consent_button.click()

    search_field_tuple = (By.ID, "search")
    search_field = wait.until(EC.element_to_be_clickable(search_field_tuple))
    search_field.click()
    search_field.clear()
    search_field.send_keys("kolicaaa")
    search_button = driver.find_element(By.CLASS_NAME, "ec-search")
    search_button.click()

    trazen_i_rezultat_text_element_touple=(By.XPATH, "//h1[@class='lead-title text-center nemaartikala']")
    trazen_i_rezultat_text_element=wait.until(EC.visibility_of_element_located(trazen_i_rezultat_text_element_touple))
    trazen_i_rezultat = trazen_i_rezultat_text_element.text

    assert trazen_i_rezultat == "Nema artikala za prikaz"
```

Test correct login

```
test_correct_login.py X
test_correct_login.py > test_correct_login
1 from selenium.webdriver.common.by import By
2 from selenium.webdriver.support.wait import WebDriverWait
3 from selenium.webdriver.support import expected_conditions as EC
4
5
6 def test_correct_login(driver):
7     wait=WebDriverWait(driver, timeout=60)
8
9     driver.get("https://onlineshop.ba/")
10    driver.maximize_window()
11
12    login_link=driver.find_element(By.LINK_TEXT, "Prijava")
13    login_link.click()
14
15    username_field_tuple=(By.ID,"Email")
16    username_field=wait.until(EC.element_to_be_clickable(username_field_tuple))
17    username_field.click()
18    username_field.clear()
19    username_field.send_keys("testFm13@outlook.com")
20
21    cookie_consent_button = driver.find_element(By.CLASS_NAME, 'pjAcceptCookieBarBtn')
22    cookie_consent_button.click()
23
24
25    lozinka_field=driver.find_element(By.ID,"Lozinka")
26    lozinka_field.click()
27    lozinka_field.clear()
28    lozinka_field.send_keys("testiranjestrance13")
29
30    loginButton=driver.find_element(By.XPATH, "//input[@value='Login' and @class='woocommerce-Button button']")
31    loginButton.click()
32
33    dobrodosli_text_element_touple=(By.XPATH, "//li[contains(text(),'Dobro došli')]/a/b")
34    dobrodosli_text_element=wait.until(EC.visibility_of_element_located(dobrodosli_text_element_touple))
35    dobrodosli_text = dobrodosli_text_element.text
36
37    assert dobrodosli_text=="testiranje"
38
39
```

Test add product to the basket

```
test_add_product_to_the_basket.py X
test_add_product_to_the_basket.py > test_add_product_to_the_basket
1 import time #implicit wait
2 from selenium.webdriver.common.by import By
3 from selenium.webdriver.support.wait import WebDriverWait
4 from selenium.webdriver.support import expected_conditions as EC
5
6 def test_add_product_to_the_basket(driver):
7     wait = WebDriverWait(driver, timeout=60)
8
9     driver.get("https://onlineshop.ba/")
10    driver.maximize_window()
11
12    search_field_tuple = (By.ID, "search")
13    search_field = wait.until(EC.element_to_be_clickable(search_field_tuple))
14    search_field.click()
15    search_field.clear()
16    search_field.send_keys("kolica")
17    search_button = driver.find_element(By.CLASS_NAME, "ec-search")
18    search_button.click()
19
20    cookie_consent_button = driver.find_element(By.CLASS_NAME, 'pjAcceptCookieBarBtn')
21    cookie_consent_button.click()
22
23    product_item=driver.find_element(By.XPATH,"//h3[contains(text(), 'KOLICA CLASSIC SET ZOOMIX KNORR BABY')]")
24    product_item.click()
25
26    dodaj_u_korpu_button=driver.find_element(By.ID,"unos")
27    dodaj_u_korpu_button.click()
28    product_in_basket_touple=(By.XPATH, "//tr[@class='cart_item']/td[@class='product-name']")
29    product_in_basket_element=wait.until(EC.visibility_of_element_located(product_in_basket_touple))
30    product_text = product_in_basket_element.text
31
32    assert product_text=="KOLICA CLASSIC SET ZOOMIX KNORR BABY"
33
```

Test delete product from the basket

```
test_delete_product_from_basket.py X
test_delete_product_from_basket.py > test_delete_product_to_the_basket
1 import time #implicitni wait
2 from selenium.webdriver.common.by import By
3 from selenium.webdriver.support.wait import WebDriverWait
4 from selenium.webdriver.support import expected_conditions as EC
5
6 def test_delete_product_to_the_basket(driver):
7     wait = WebDriverWait(driver, timeout=60)
8
9     driver.get("https://onlineshop.ba/")
10    driver.maximize_window()
11
12    search_field_tuple = (By.ID, "search")
13    search_field = wait.until(EC.element_to_be_clickable(search_field_tuple))
14    search_field.click()
15    search_field.clear()
16    search_field.send_keys("kolica")
17    search_button = driver.find_element(By.CLASS_NAME, "ec-search")
18    search_button.click()
19
20    cookie_consent_button = driver.find_element(By.CLASS_NAME, 'pjAcceptCookieBarBtn')
21    cookie_consent_button.click()
22
23    product_item=driver.find_element(By.XPATH,"//h3[contains(text(), 'KOLICA CLASSIC SET ZOOMIX KNORR BABY')]")
24    product_item.click()
25
26    dodaj_u_korpu_button=driver.find_element(By.ID,"unos")
27    dodaj_u_korpu_button.click()
28
29    brisi_iz_korpe_tuple = (By.CLASS_NAME, "delete")
30    brisi_iz_korpe=wait.until(EC.element_to_be_clickable(brisi_iz_korpe_tuple))
31    brisi_iz_korpe.click()
32    text_in_basket_touple = (By.CLASS_NAME, "lead-title.text-center.cart-empty")
33    text_in_basket_element = wait.until(EC.visibility_of_element_located(text_in_basket_touple))
34    basket_text = text_in_basket_element.text
35    expected_text = "Vaša korpa je trenutno prazna"
36    assert basket_text == expected_text
37
38
39
```

Test invalid login

```
test_invalid_login.py X
test_invalid_login.py > test_incorrect_login
1 from selenium.webdriver.common.by import By
2 from selenium.webdriver.support.wait import WebDriverWait
3 from selenium.webdriver.support import expected_conditions as EC
4
5 def test_incorrect_login(driver):
6     wait = WebDriverWait(driver, timeout=60)
7
8     driver.get("https://onlineshop.ba/")
9     driver.maximize_window()
10
11    cookie_consent_button = driver.find_element(By.CLASS_NAME, 'pjAcceptCookieBarBtn')
12    cookie_consent_button.click()
13
14    login_link = driver.find_element(By.LINK_TEXT, "Prijava")
15    login_link.click()
16
17    username_field_tuple = (By.ID, "Email")
18    username_field = wait.until(EC.element_to_be_clickable(username_field_tuple))
19    username_field.click()
20    username_field.clear()
21    username_field.send_keys("neispravanemail@example.com") # Unesite neispravne podatke za prijavu
22
23    lozinka_field = driver.find_element(By.ID, "Lozinka")
24    lozinka_field.click()
25    lozinka_field.clear()
26    lozinka_field.send_keys("neispravnaLozinka")
27
28    loginButton = driver.find_element(By.XPATH, "//input[@value='Login' and @class='woocommerce-Button button']")
29    loginButton.click()
30
31    neispravna_prijava_text_element_touple = (By.XPATH, "//div[@class='text-danger']")
32    neispravna_prijava_element = wait.until(EC.visibility_of_element_located(neispravna_prijava_text_element_touple))
33    neispravna_prijava_text = neispravna_prijava_element.text.strip()
34    assert neispravna_prijava_text == "Login podaci netačni ili nepotpuni"
35
```

Test remove product from the wish list

```
from selenium.webdriver.common.by import By
from selenium.webdriver.support.wait import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC

def test_remove_product_from_wishlist(driver):
    wait = WebDriverWait(driver, timeout=60)

    driver.get("https://onlineshop.ba/")
    driver.maximize_window()

    cookie_consent_button = driver.find_element(By.CLASS_NAME, 'pjAcceptCookieBarBtn')
    cookie_consent_button.click()

    categories_tuple = (By.XPATH, "//a[text()='Kategorije']")
    categories_field = wait.until(EC.element_to_be_clickable(categories_tuple))
    categories_field.click()

    category_field = driver.find_element(By.XPATH, "//a[text()=' Igračke ']")
    category_field.click()

    product_tuple = (By.XPATH, "//ul[@id='listList']/li[1]/div[@class='media']/div[@class='media-left']/a")
    product_field = wait.until(EC.element_to_be_clickable(product_tuple))
    product_field.click()

    add_product_tuple = (By.XPATH, "//div[@class='summary entry-summary']/div[@class='action-buttons']/a[@class='add_to_wishlist']")
    add_product_field = wait.until(EC.element_to_be_clickable(add_product_tuple))
    add_product_field.click()

    swal_tuple = (By.CLASS_NAME, 'swal2-container')
    wait.until(EC.visibility_of_element_located(swal_tuple))
    wait.until(EC.invisibility_of_element_located(swal_tuple))

    search_button_tuple = driver.find_element(By.CLASS_NAME, "ec-favorites")
    search_button = wait.until(EC.element_to_be_clickable(search_button_tuple))
    search_button.click()

    brisi_iz_liste_tuple = (By.CLASS_NAME, "deleteWL.delete.remove.remove_from_wishlist")
    brisi_iz_liste = wait.until(EC.element_to_be_clickable(brisi_iz_liste_tuple))
    brisi_iz_liste.click()

    text_in_list_touple = (By.CLASS_NAME, "lead-title.text-center.cart-empty")
    text_in_list_element = wait.until(EC.visibility_of_element_located(text_in_list_touple))
    list_text = text_in_list_element.text
    expected_text = "Vaša lista želja je trenutno prazna"
    assert list_text == expected_text
```

Sorting products by their prices

```
test_invalid_login.py X test_sort_products_by_prices.py X test_remove_product_from_wishlist.py
1 import time #implicit wait
2 from selenium.webdriver.common.by import By
3 from selenium.webdriver.support.wait import WebDriverWait
4 from selenium.webdriver.support import expected_conditions as EC
5
6 def test_sort_products(driver):
7     wait = WebDriverWait(driver, timeout=60)
8
9     driver.get("https://onlineshop.ba/")
10    driver.maximize_window()
11
12    cookie_consent_button = driver.find_element(By.CLASS_NAME, 'pjAcceptCookieBarBtn')
13    cookie_consent_button.click()
14
15    categories_tuple = (By.XPATH, "//a[text()='Kategorije']")
16    categories_field = wait.until(EC.element_to_be_clickable(categories_tuple))
17    categories_field.click()
18
19    category_field = driver.find_element(By.XPATH, "//a[text()=' Mobitel ']")
20    category_field.click()
21
22    sortiranje_field = driver.find_element(By.ID, "sortiranje")
23    sortiranje_field.click()
24
25
26    sortiranje_po_cijeni = driver.find_element(By.ID, "priceAsc")
27    sortiranje_po_cijeni.click()
28    time.sleep(5)
29    product_elements = wait.until(EC.presence_of_all_elements_located((By.XPATH, "//ul[@id='listList']/li")))
30    prices = []
31    for i, product_element in enumerate(product_elements):
32        price_element = product_element.find_element(By.XPATH, ".//span[@class='amount']")
33        price_text = price_element.text
34        if price_text:
35            price_value = float(price_text.replace('KM', '').replace('.', ''))
36            prices.append(price_value)
37
38
39    is_sorted = all(prices[i] <= prices[i + 1] for i in range(len(prices) - 1))
40    if is_sorted:
41        sorted_prices = sorted(prices)
42        assert prices == sorted_prices, "Products are not sorted by price in ascending order"
43    else:
44        print("Prices are not sorted, skipping assertion.")
45
```

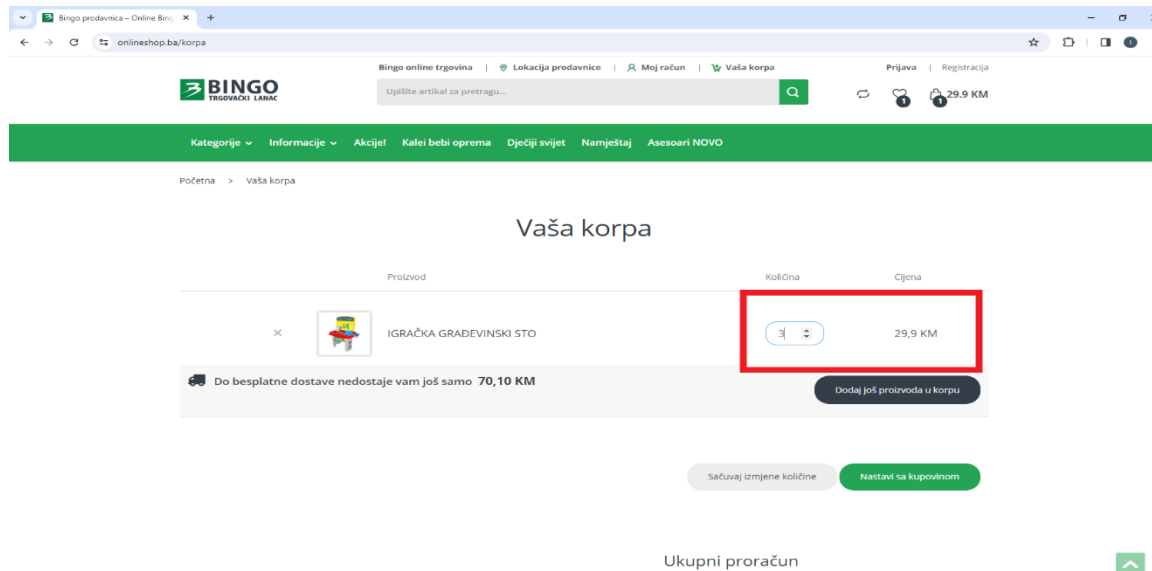
Zadatak 5.

Sve defekte (bugove) koje pronađete na web aplikaciji prijavite korištenjem šablona za prijavu defekata sa predavanja. Ako na datoj aplikaciji ne pronađete nijedan bug, istestirajte neku drugu aplikaciju po svojoj volji i za istu prijavite najmanje 5 bugova korištenjem šablona za prijavu defekata sa predavanja.

Napomena: U nastavku se nalaze kratki opisi i slike zaslona na kojima su prikazane navedene greške. U mailu odvojeno prilažemo i excel fajl sa Defect Report-om.

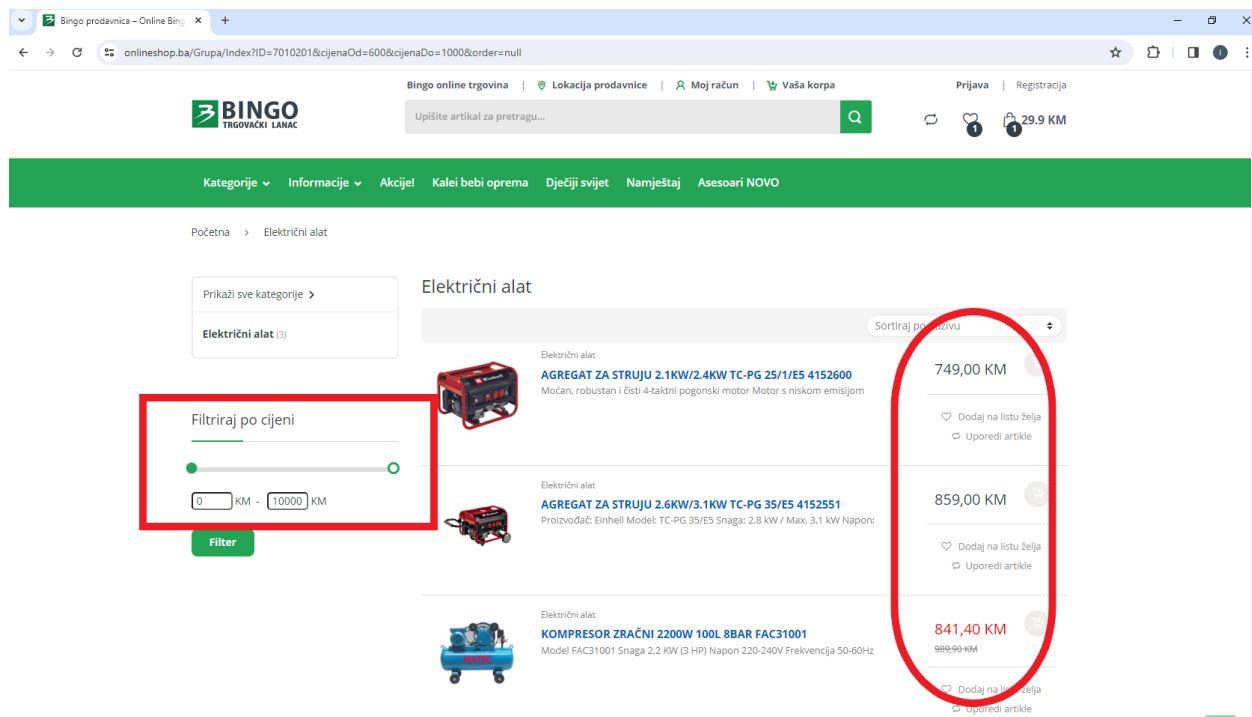
Bug 1.:

Kada dodamo proizvod u korpu i povećamo mu količinu cijena pored proizvoda ostaje ista:



Bug 2.:

Unijeli smo raspon cijene od 600-100 i prikazali su nam se proizvodi u tom rasponu, međutim filter se vratio na defaultne vrijednosti.



Bug 3.:

Kada sortiramo proizvod po cijeni od najniže do najviše i unesemo raspon cijene 100-300. Proizvodi se ne sortiraju samo se prikazuju proizvodi u rasponu od 100-300. i filteri se vrate na defaultne vrijednosti.

← → ↻ onlineshop.ba/Grupa/Rasprodaja5 ☆

Bingo online trgovina | Lokacija prodavnice | Moj račun | Vaša korpa | Prijava | Registracija

Upišite artikal za pretragu... 🔍

🔄 📌 29.9 KM

Kategorije ▾ Informacije ▾ Akcije! Kalei bebi oprema Dječiji svijet Namještaj Asesoari NOVO

Početna > Namještaj


Filtriraj po cijeni

0 KM - 10000 KM

Filter

Namještaj


Sortiraj po cijeni (najniža - najviša) ▾



Namještaj
STOLIĆ POMOĆNI CHAMPO 2U1
Stolić pomoćni "CHAMPO" 2u1, Bijela visoki sjaj/Orah - Novo u ponudi!

71,20 KM
~~89,00 KM~~


♡ Dodaj na listu želja
🔄 Uporedi artikle



Namještaj
KOMODA ZA TV CORNELIA
TV komoda "Cornelia" 120cm, Bijela visoki sjaj/Mermer - Novo u ponudi!

95,20 KM
~~119,00 KM~~

♡ Dodaj na listu želja
🔄 Uporedi artikle



Namještaj
POLICE OTVORENE LONGWAY
Otvorene police "LONGWAY" 80cm, Bijela visoki sjaj/Crna - Novo u ponudi!

103,20 KM
~~129,00 KM~~

♡ Dodaj na listu želja
🔄 Uporedi artikle

← → ↻ onlineshop.ba/Grupa/Rasprodaja?cijenaOd=100&cijenaDo=300&order=null&cijenaOd=100&cijenaDo=300&order=null ☆

Bingo online trgovina | Lokacija prodavnice | Moj račun | Vaša korpa | Prijava | Registracija

Upišite artikal za pretragu... 🔍

🔄 📌 29.9 KM

Kategorije ▾ Informacije ▾ Akcije! Kalei bebi oprema Dječiji svijet Namještaj Asesoari NOVO

Početna > Rasprodaja


Filtriraj po cijeni

0 KM - 10000 KM

Filter

Rasprodaja


Sortiraj po nazivu ▾



Kolica i oprema za bebe
HRANILICA BEN PANDA BLUE FREEON
Hranilica Ben odličan je izbor za hranjenje i igru djeteta. Kada je ne koristite.

122,95 KM
~~159,90 KM~~


♡ Dodaj na listu želja
🔄 Uporedi artikle



Kolica i oprema za bebe
HRANILICA BEN BEAR ORANGE FREEON
Hranilica Ben odličan je izbor za hranjenje i igru djeteta. Kada je ne koristite.

122,95 KM
~~159,90 KM~~

♡ Dodaj na listu želja
🔄 Uporedi artikle



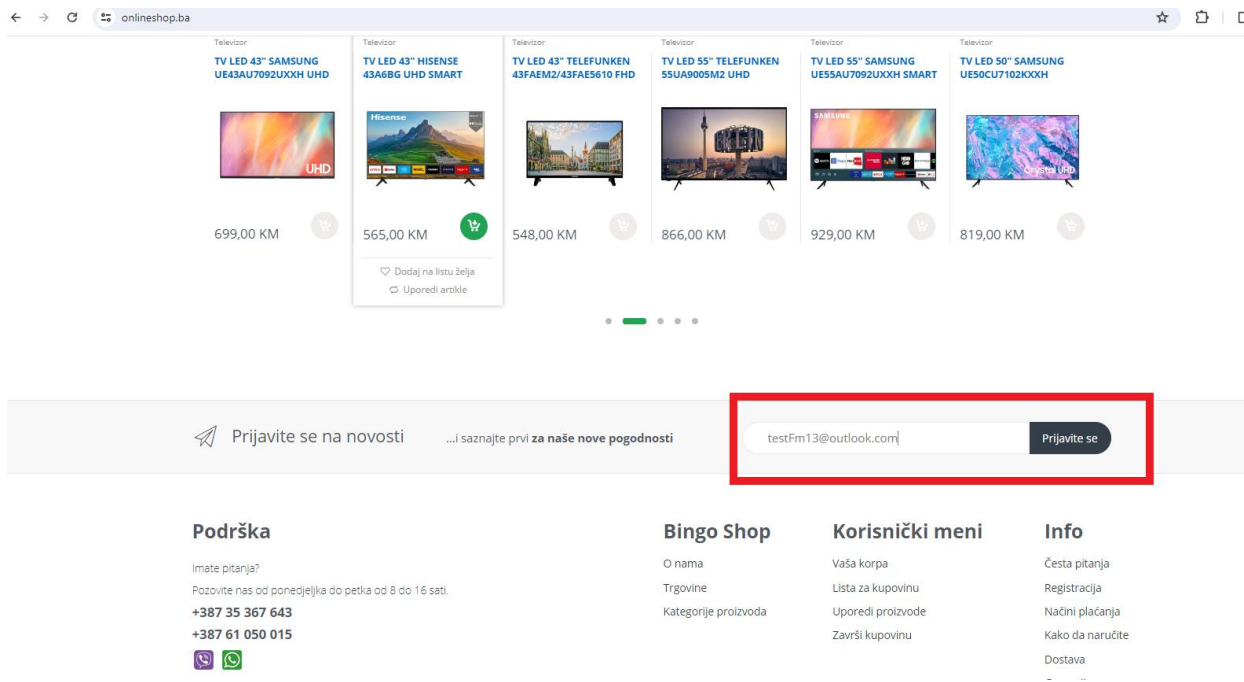
Kolica i oprema za bebe
HODALICA LITTLE DRIVER FREEON
- Udobno obloženo, skidivo i prvo sjedalo - Mogućnost postavljanja na tri

109,95 KM
~~139,90 KM~~

♡ Dodaj na listu želja
🔄 Uporedi artikle

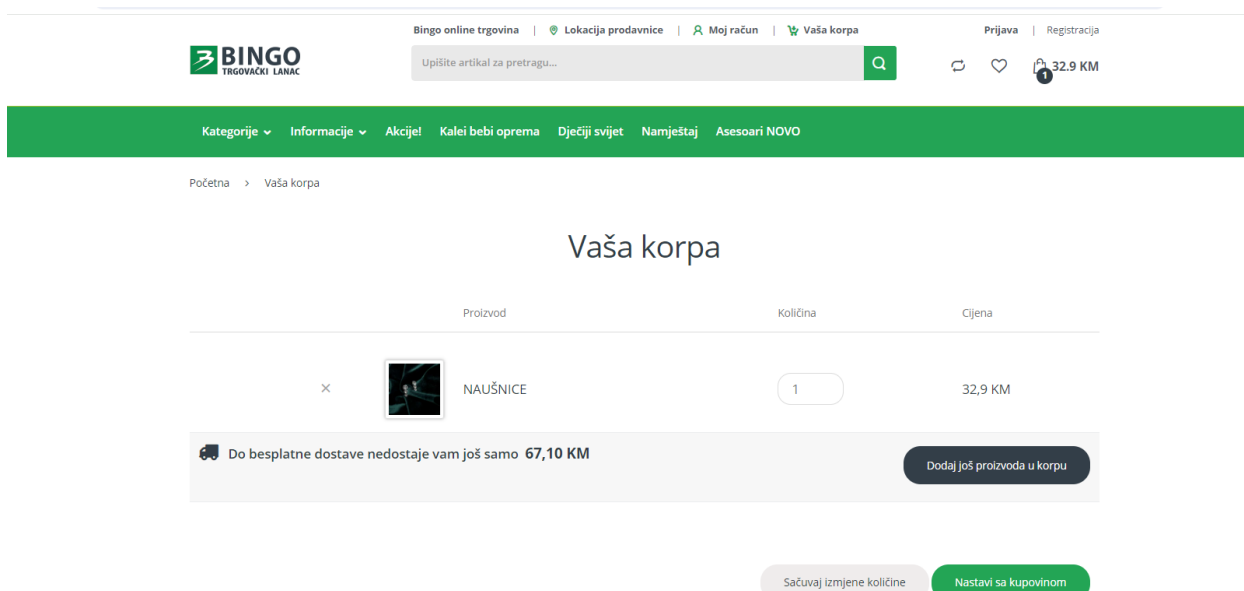
Bug 4.:

Unijeli smo email u text box na kraju stranice i kliknuli dugme „Prijavi se“, međutim nismo dobili nikakvu povratnu informaciju.



Bug 5.:

Kada odaberemo proizvod i kliknemo na njega da bismo ušli u detalje proizvoda, zatim pritisnemo dugme „Dodaj u korpu“ ne dobijemo nikakvu poruku da smo proizvod dodali u korpu.



Bug 6.:

Kada prvo dodamo jedan proizvod, zatim opet dodamo isti proizvod u korpu ne povećava se količina proizvoda u korpi za taj proizvod. Taj proizvod će stajati posebno u korpi dva puta.

← → ↻ onlineshop.ba/korpa ☆ 📄 📱

Bingo online trgovina | 📍 Lokacija prodavnice | 🧑 Moj račun | 🛒 Vaša korpa | Prijava | Registracija

BINGO
TRGOVACKI LANAC



Upišite artikal za pretragu... 🔍

🔄 ❤️ 📦 65.8 KM

Kategorije ▾ Informacije ▾ Akcije! Kalei bebi oprema Dječiji svijet Namještaj Asesoari NOVO

Početna > Vaša korpa

Vaša korpa

	Proizvod	Količina	Cijena
×	 NAUŠNICE	1	32,9 KM
×	 NAUŠNICE	1	32,9 KM

📦 Do besplatne dostave nedostaje vam još samo **34,20 KM**

[Dodaj još proizvoda u korpu](#)

[Sačuvaj izmjene količine](#) [Nastavi sa kupovinom](#)

Bug 7.:

Kada ostavimo recenziju, unesemo ocjenu i u mjesto za poruku recenzije unesemo space (pošto je obavezno polje). Nećemo dobiti poruku o grešci.

← → ↺ onlineshop.ba/home/artikal?artikal=SN0225 ☆ 📄 📱



Opis **Recenzije**

0,00
prosječna ocjena

★★★★★

★★★★☆

★★★☆☆

★★★☆☆

★★☆☆☆

★☆☆☆☆

0

0

0

0

0

0

Ostavi recenziju za "STO KLUB REETA"

Vaša ocjena

★★★★★

Vaša recenzija

Ostavi recenziju

Opis **Recenzije**

5,00
prosječna ocjena

★★★★★

★★★★☆

★★★☆☆

★★★☆☆

★★☆☆☆

★☆☆☆☆

1

0

0

0

0

0

Ostavi recenziju za "STO KLUB REETA"

Vaša ocjena

☆☆☆☆☆

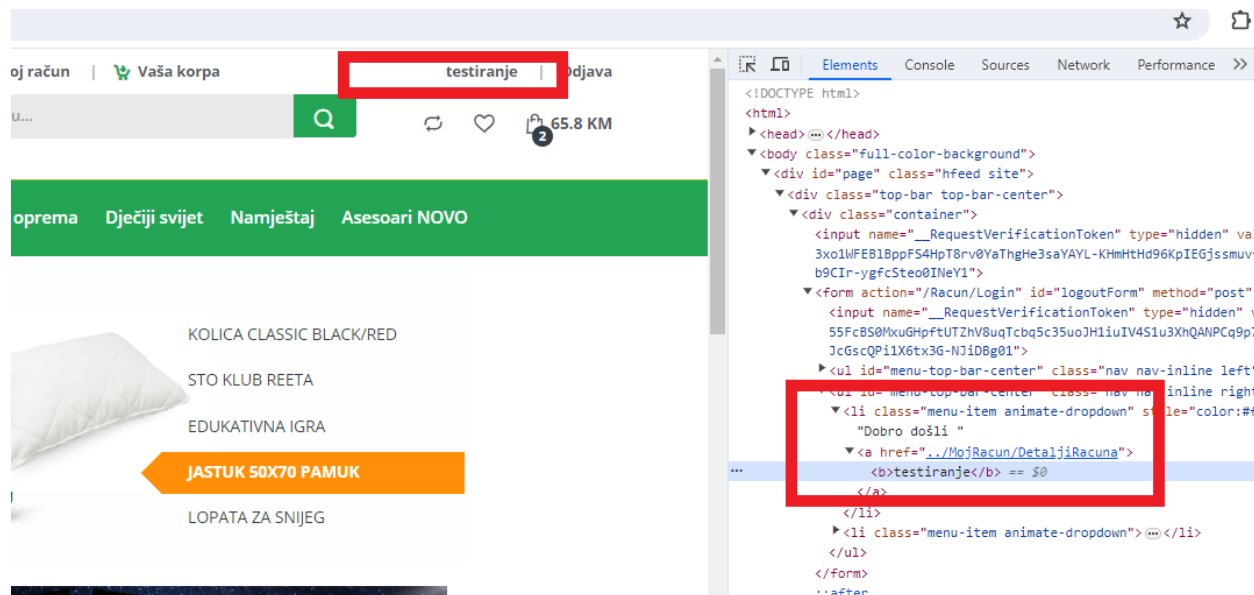
Vaša recenzija

Ostavi recenziju

★★★★★

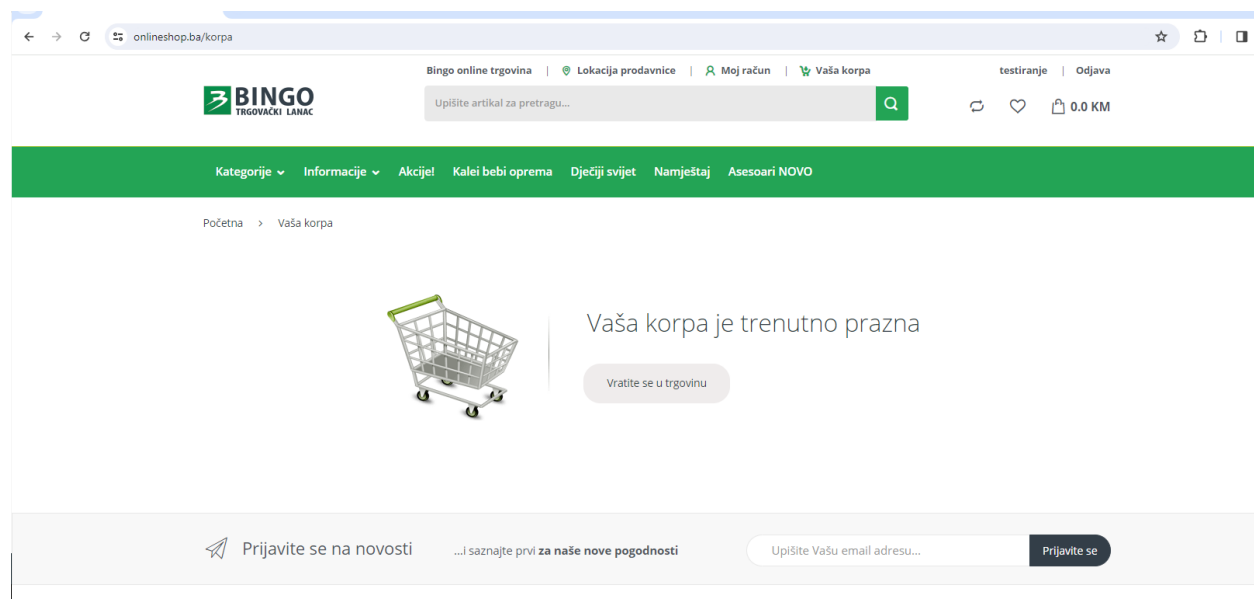
Bug 8.:

Kada se logiramo na stranicu u gornjem desnom ćošku stoji samo korsničko ime, ne ispisuje se poruka „Dobro došli“.



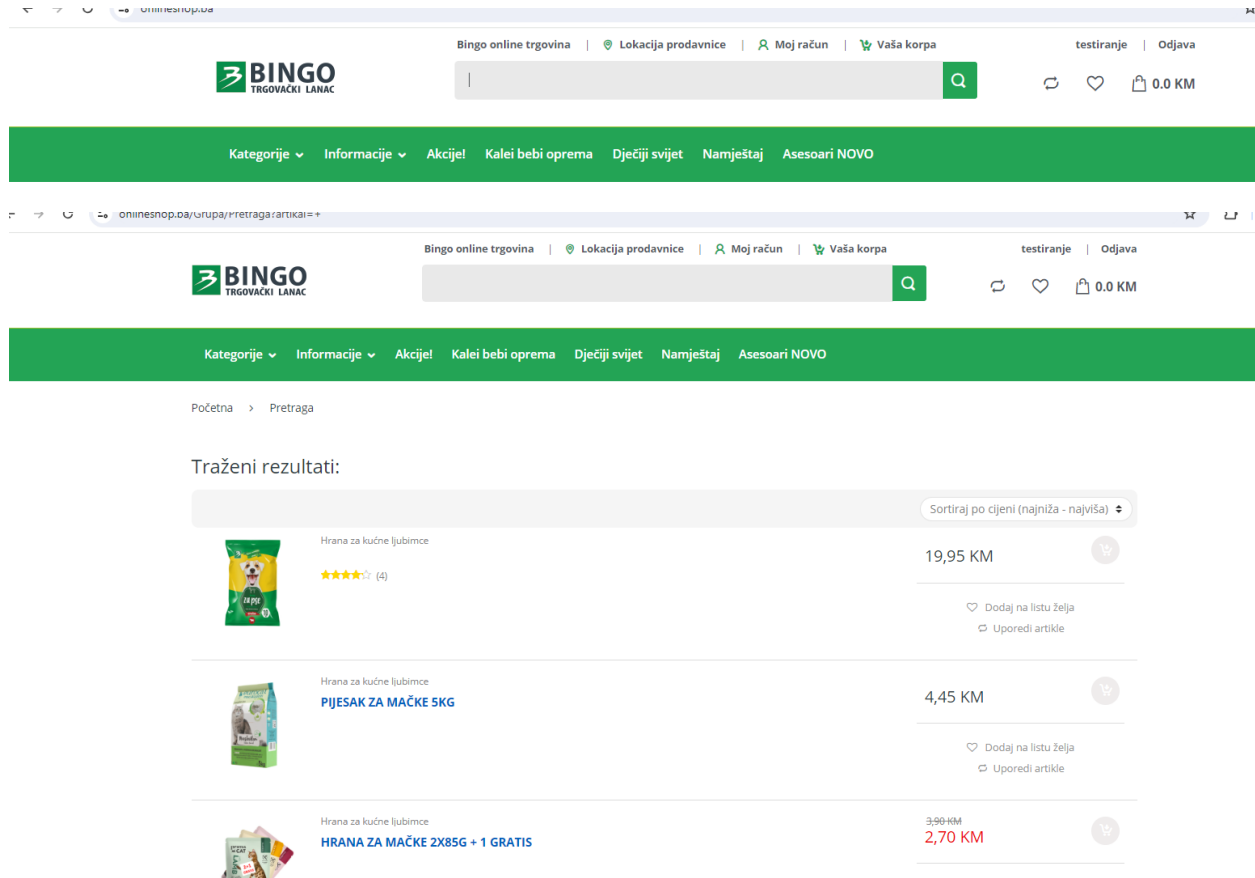
Bug 9.:

Kada izbrisemo proizvod iz korpe ne dobijamo nikakvu poruku o tome.

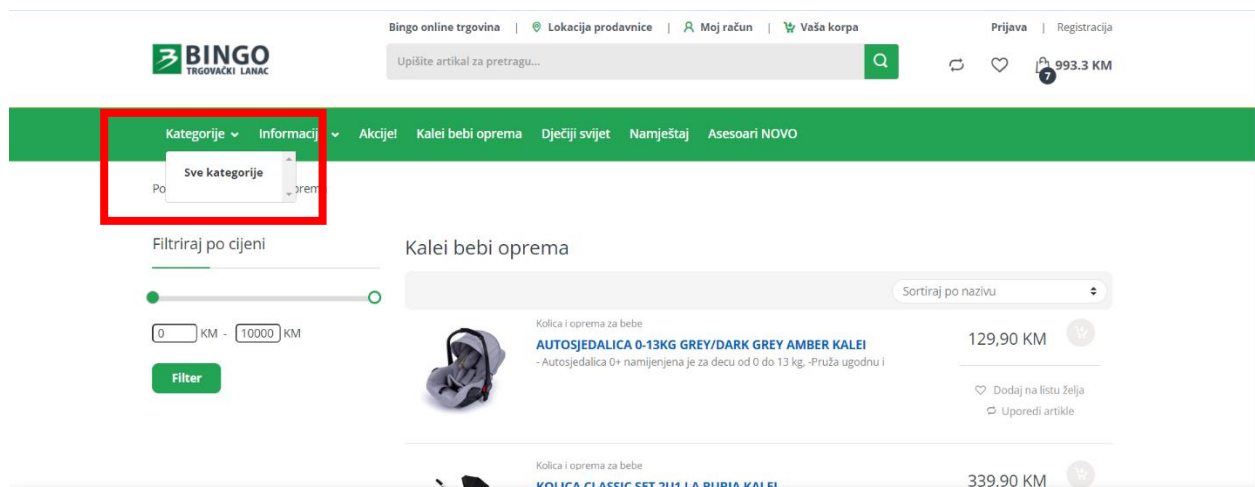


Bug 10.:

Kada ukucamo space u filter za pretragu po nazivu, dobijamo listu nekih proizvoda



Bug 11.: Kada kliknemo na "Kalei bebi oprema" pa kliknemo na "Kategorije" dropdown, ne prikažu nam se kategorije



Bug 12.: Kada kliknemo na bilo koju kategoriju (u ovom slučaju je kategorija “Mobiteli” i unesemo tekst (“iPhone”) u filter za cijenu, ne prikaže se bilo kakav error. Filter bude primjenjen na listu artikala

Bingo online trgovina | Lokacija prodavnice | Moj račun | Vaša korpa | Prijava | Registracija

Upišite artikal za pretragu...

Kategorije ▾ Informacije ▾ Akcije! Kalei bebi oprema Dječiji svijet Namještaj Asesoari NOVO

Početna > Mobitel

Prikaži sve kategorije >

Mobitel (39)

Filtriraj po cijeni

iPhone KM 10000 KM

Filter

Sortiraj po nazivu ▾

Mobitel

MOBITEL APPLE IPHONE 13 128GB
Procesor (CPU) Apple A# / Hexa-Core / 3.2 GHz Memorija (RAM) 4GB Ekran
1.449,00 KM
~~1.999,00 KM~~

♥ Dodaj na listu želja
🔄 Uporedi artikle

Mobitel

MOBITEL CUBOT NOTE 21 6GB/128GB DUAL SIM CRNI
Ekran Dijagonala: 6.56 inča Rezolucija: 720 x 1612 90Hz refresh rate Procesor
215,00 KM
~~259,00 KM~~

♥ Dodaj na listu želja

Bingo online trgovina | Lokacija prodavnice | Moj račun | Vaša korpa | Prijava | Registracija

Upišite artikal za pretragu...

Kategorije ▾ Informacije ▾ Akcije! Kalei bebi oprema Dječiji svijet Namještaj Asesoari NOVO

Početna > Mobitel

Prikaži sve kategorije >

Mobitel (39)

Filtriraj po cijeni

0 KM - 10000 KM

Filter

Sortiraj po nazivu ▾

Mobitel

MOBITEL APPLE IPHONE 13 128GB
Procesor (CPU) Apple A# / Hexa-Core / 3.2 GHz Memorija (RAM) 4GB Ekran
1.449,00 KM
~~1.999,00 KM~~

♥ Dodaj na listu želja
🔄 Uporedi artikle

Mobitel

MOBITEL CUBOT NOTE 21 6GB/128GB DUAL SIM CRNI 53681
Ekran Dijagonala: 6.56 inča Rezolucija: 720 x 1612 90Hz refresh rate Procesor
215,00 KM
~~259,00 KM~~

♥ Dodaj na listu želja

Bug 13.: Kada uđemo na bilo koju kategoriju, dropdown na stranici koji prikazuje kategorije sadrži višak koda

