Univerzitet „Džemal Bijedić“ u Mostaru,

Fakultet informacijskih tehnologija, Softverski inženjering

**Testiranje web aplikacije „Online Bingo prodavnica“**

Predmet: Formalne metode

Autori: Lamija Babović IB230398, Asistent: mr. Ahmet Mulalić

Mustafa Isić IB190022,

Ismihana Međedović IB190022,

Inas Bajrektarević IB200007

# 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 zahtijeva 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 il 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 ommogućavaju filtriranja 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 | Navažeća | -922337203685488 |
|  | Vrijednost > 922337203685477 | Navažeća | 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.*

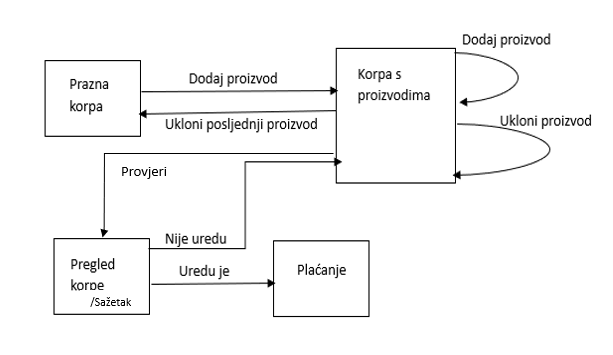
Tabela odluke za testiranje funkcionalnosti besplatne dostave na ovoj stranici ima za cilj proveru ispravnosti sistema u skladu s definisanim poslovnim pravilima.Ova stranica nudi besplatnu dostavu na adresu koju korisnik unese. Da bi dobili besplatnu dostavu, iznos korisnikove narudžbe treba biti 100KM 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 slučaje 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 uredu | / | / | Korpa s proizvodima | / |
| Uredu je | / | / | Plaćanje | / |

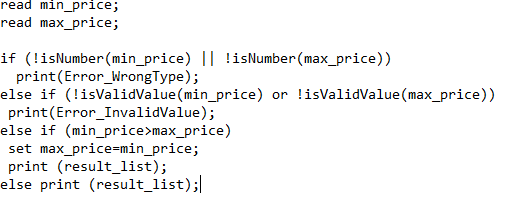
### 5. Testiranje izjava (statements) i pokrivenost

### 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 .*

*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).



|  |  |
| --- | --- |
| **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 .*

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

A diagram of a flowchart

Description automatically generated

|  |  |
| --- | --- |
| **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 defekate 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 | Otiđi 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 testra. 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 proceduara, 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 apspekata. 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**

A screenshot of a document

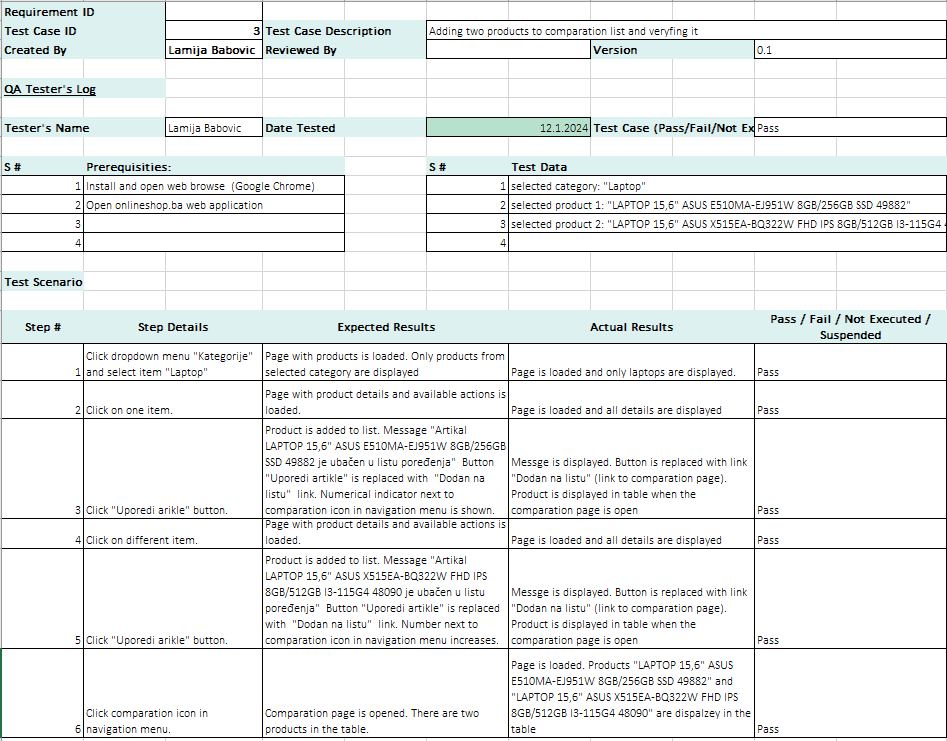
Description automatically generated

**Test slučaj na niskom nivou 2**

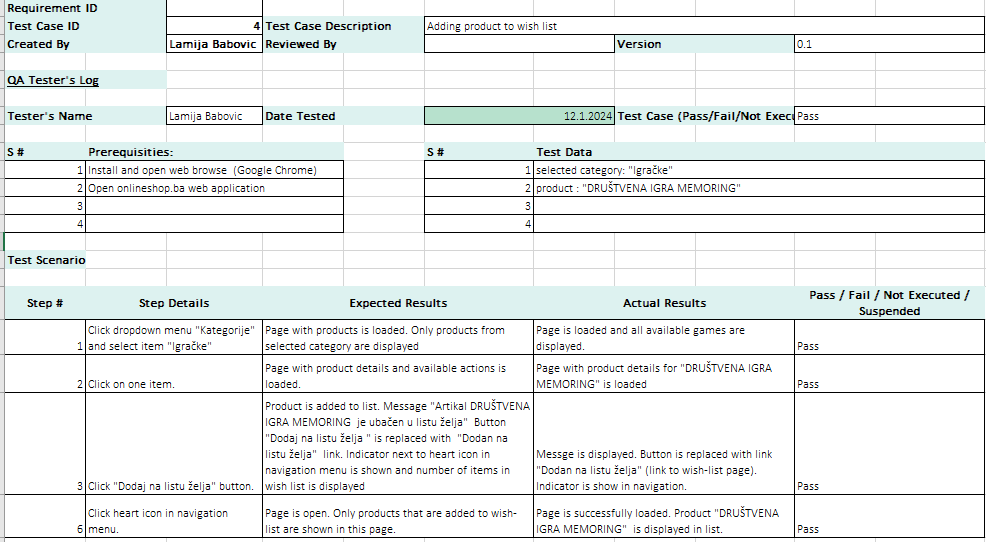
A document with text and numbers

Description automatically generated

**Test slučaj na niskom nivou 3**



**Test slučaj na niskom nivou 4**



**Test slučaj na niskom nivou 5**

A close-up of a document

Description automatically generated

**Test slučaj na niskom nivou 6**

A close-up of a document

Description automatically generated

**Test slučaj na niskom nivou 7**

A close-up of a document

Description automatically generated

**Test slučaj na niskom nivou 8**

A close-up of a document

Description automatically generated

**Test slučaj na niskom nivou 9**

A document with text and numbers

Description automatically generated

**Test slučaj na niskom nivou 10**

A screenshot of a document

Description automatically generated

# Zadatak 4.

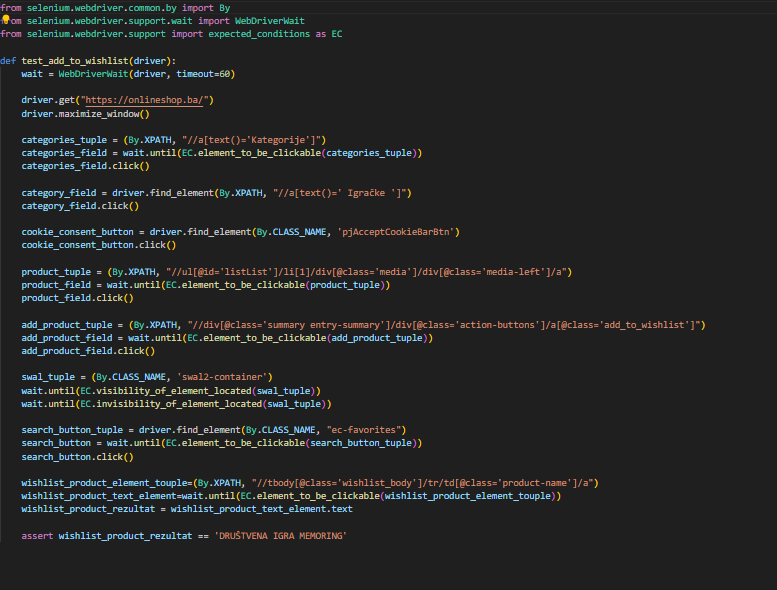
## Automatizirajte prethodno napisane testne slučajeve na niskom nivou korištenjem Selenium-a, Python-a i Pytest-a. Svaki testni slučaj mora biti zaseban test i mora imati određenu validaciju (assert) koja će utvrditi da li je test uspješno izvršen ili ne. Nemojte zaboraviti napraviti setup i teardown 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**

A screen shot of a computer program

Description automatically generated

**Test add to wish list**



**Test correct search**

A screen shot of a computer program

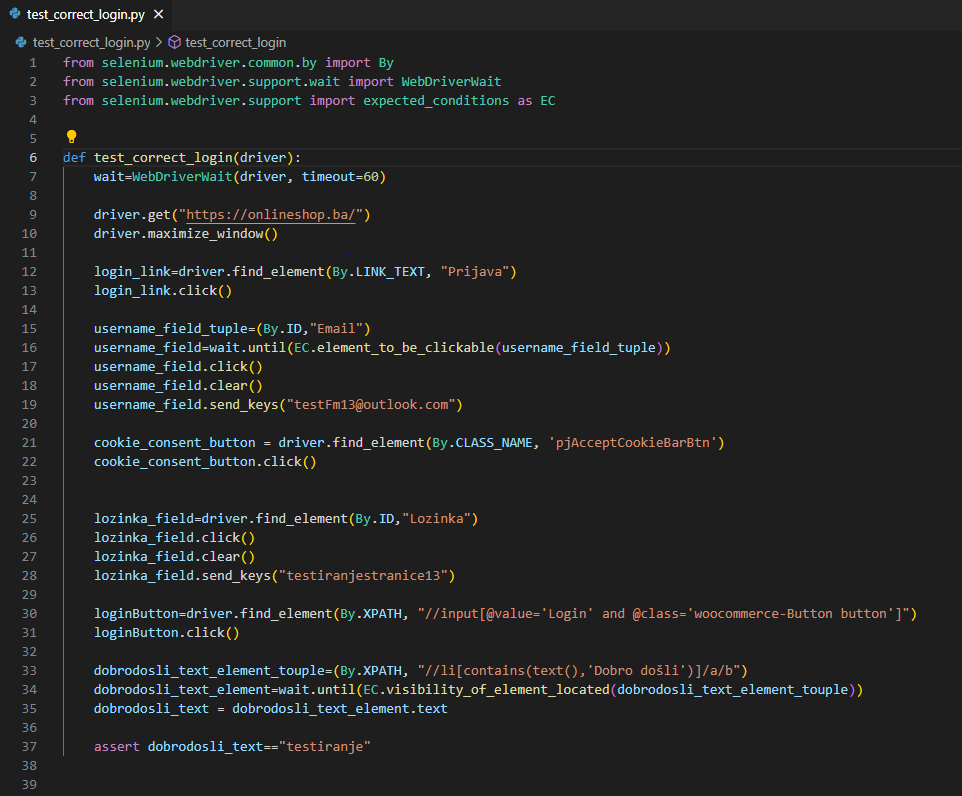
Description automatically generated

**Test incorrect search**

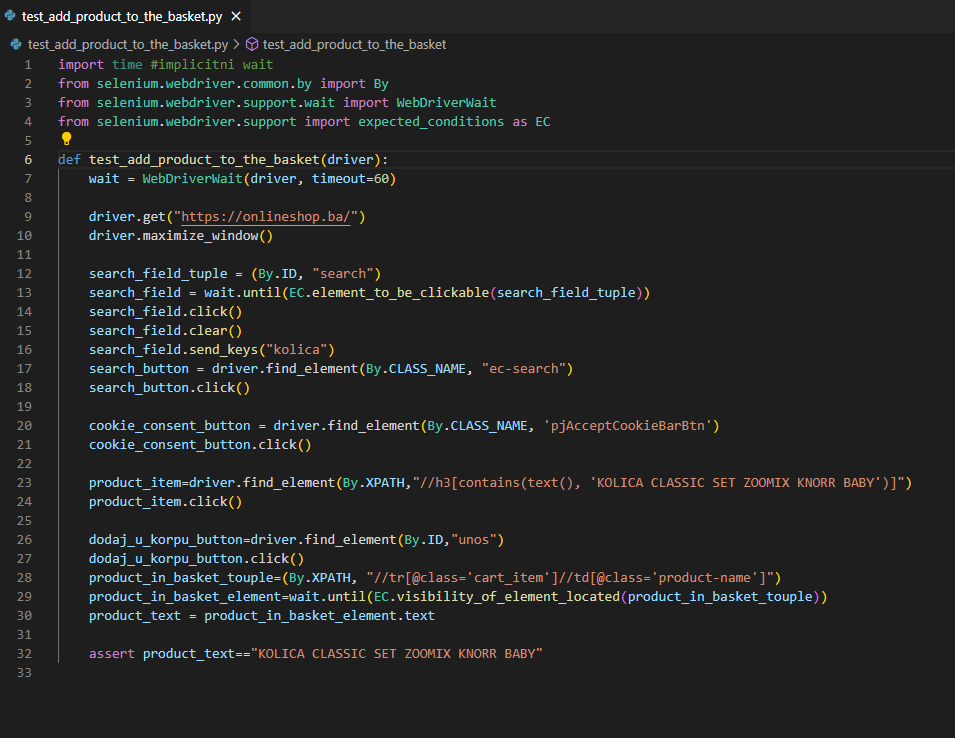
A screen shot of a computer program

Description automatically generated

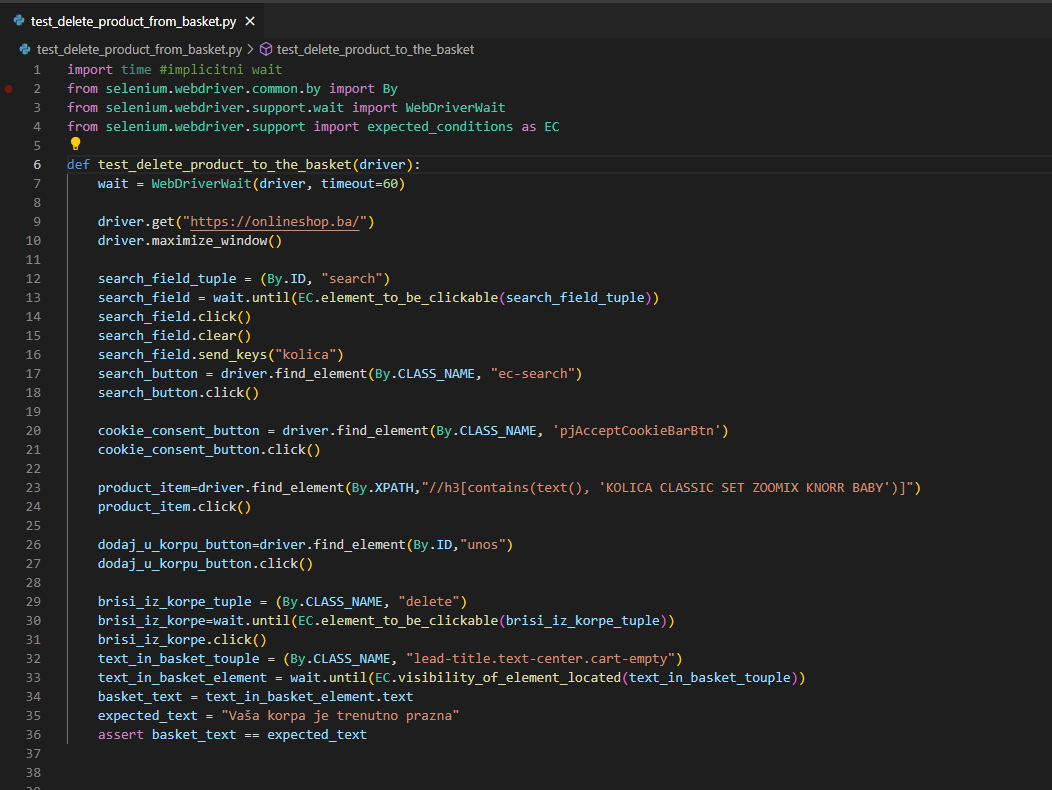
**Test correct login**



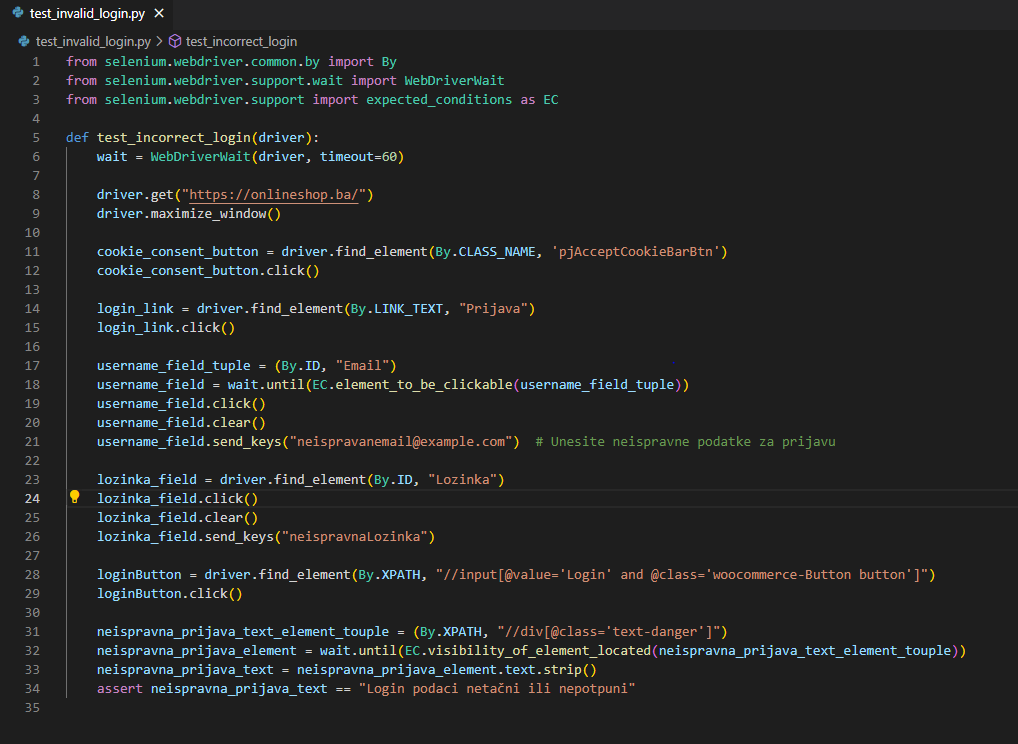
**Test add product to the basket**



**Test delete product from the basket**



**Test invalid login**

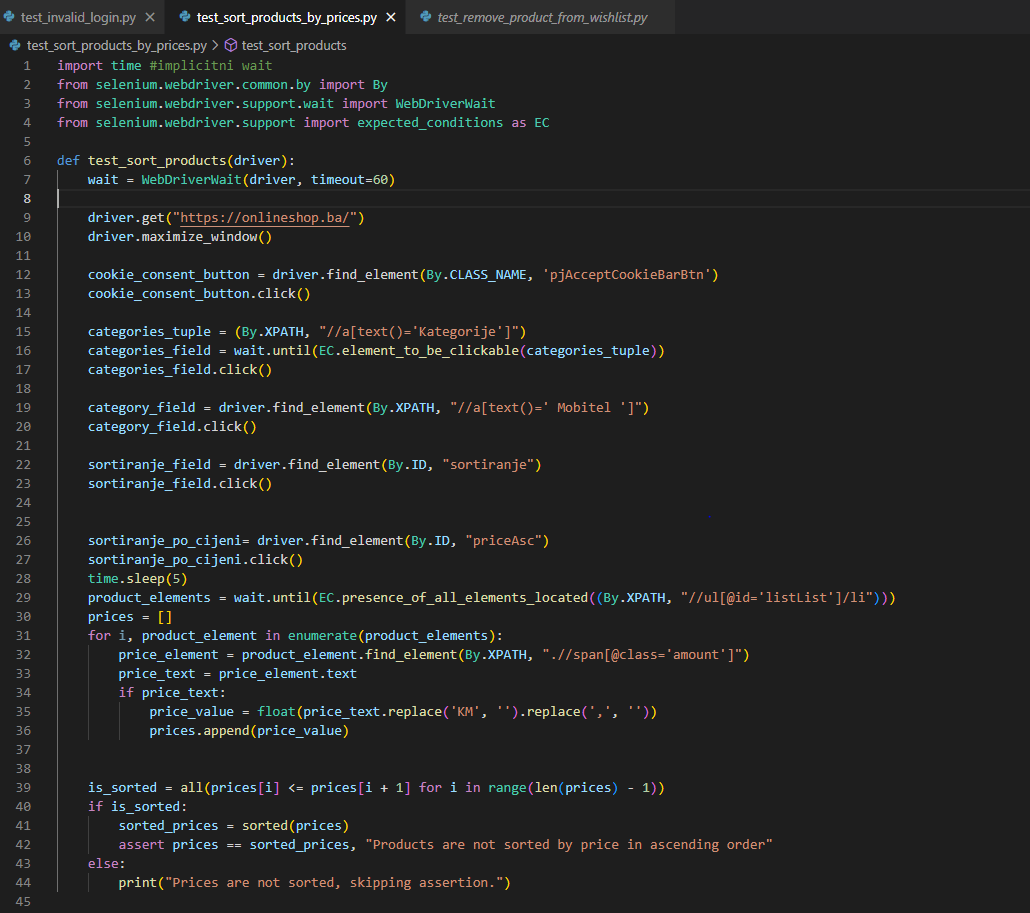


**Test remove product from the whish list**

A computer screen shot of text

Description automatically generated

**Sorting products by their prices**



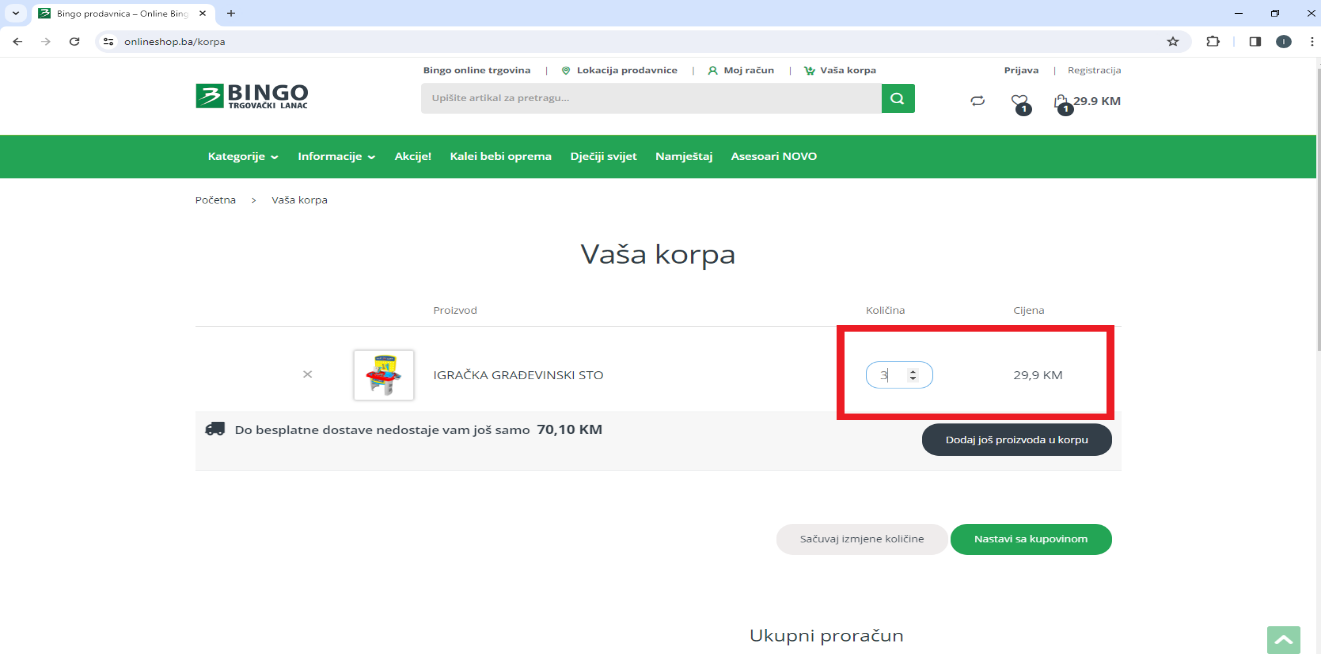
# Zadatak 5.

## 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.

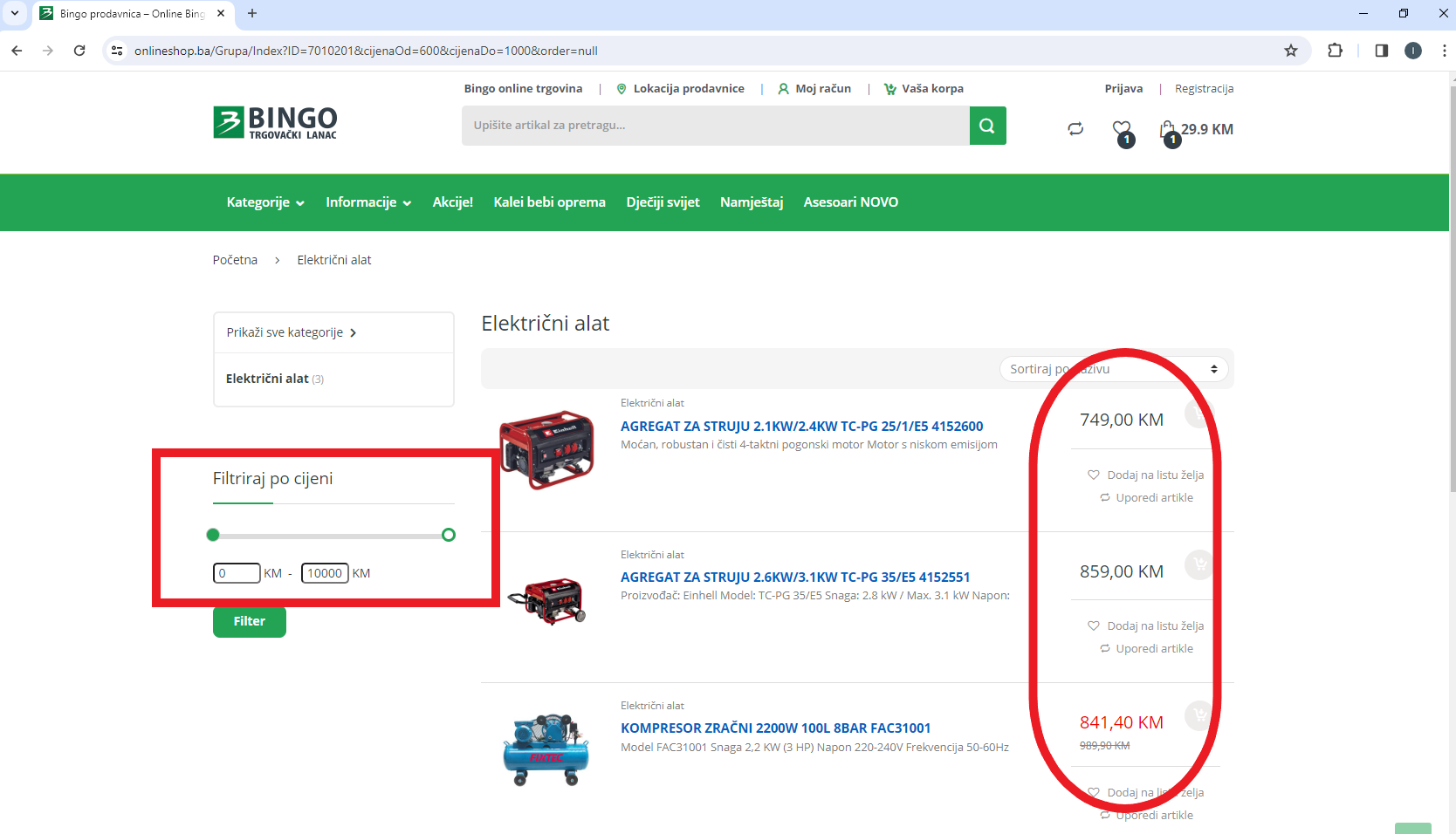
**Bug 1.:**

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 prikažu proizvodi u rasponu od 100-300. i filteri se vrate na defaultne vrijednosti.

A screenshot of a web page

Description automatically generated

A screenshot of a web page

Description automatically generated

**Bug 4.:**

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

A screenshot of a computer

Description automatically generated

**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.

A screenshot of a computer

Description automatically generated

**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.

A screenshot of a computer

Description automatically generated

**Bug 7.:**

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

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Bug 8.:**

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

**A screenshot of a computer

Description automatically generated**

**Bug 9.:**

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

A screenshot of a computer

Description automatically generated

**Bug 10.:**

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

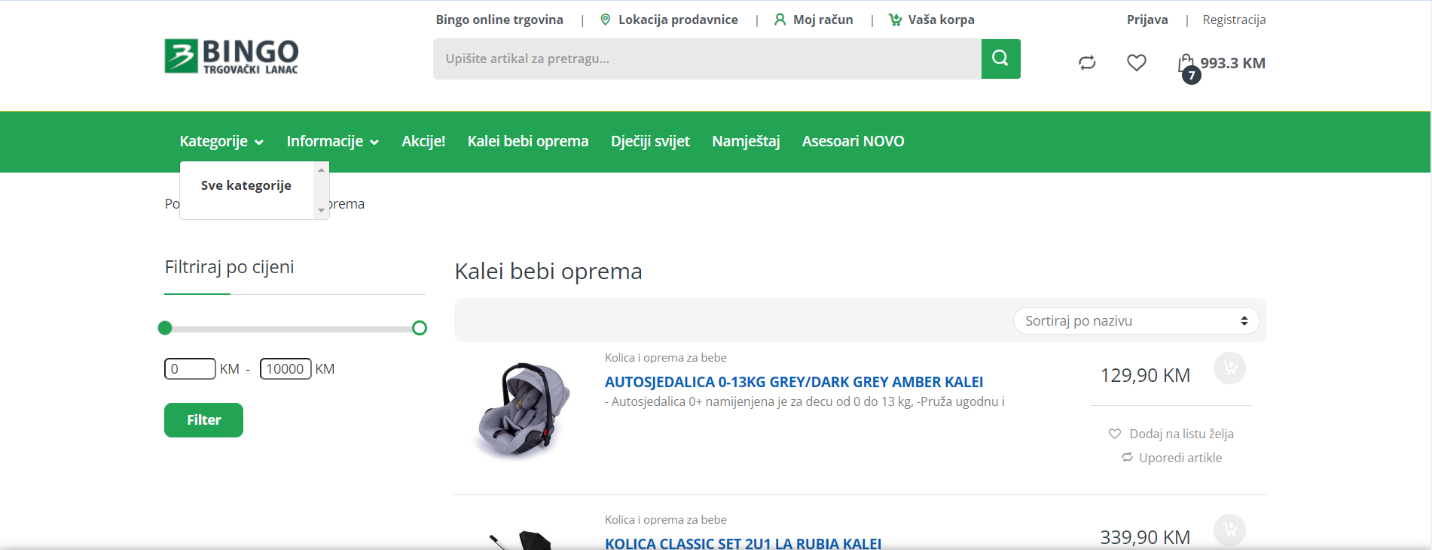
A screenshot of a computer

Description automatically generated

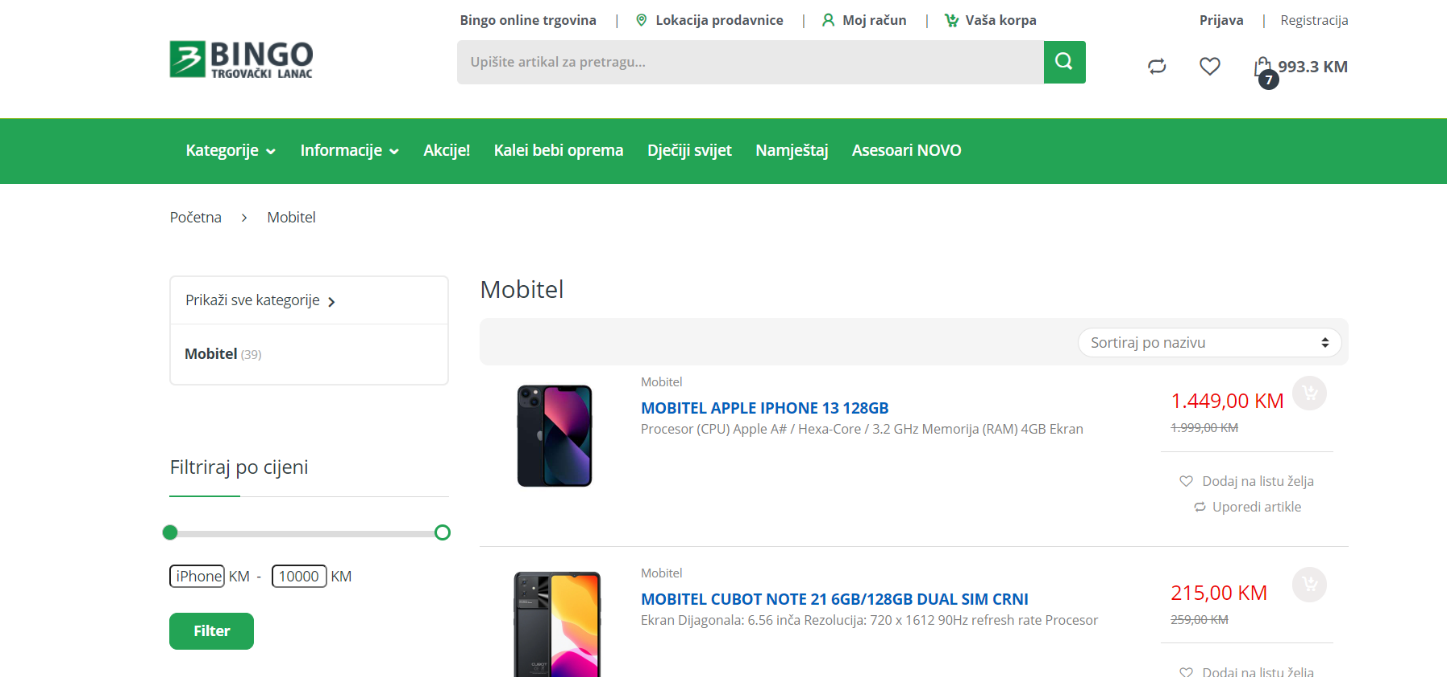
A screenshot of a computer

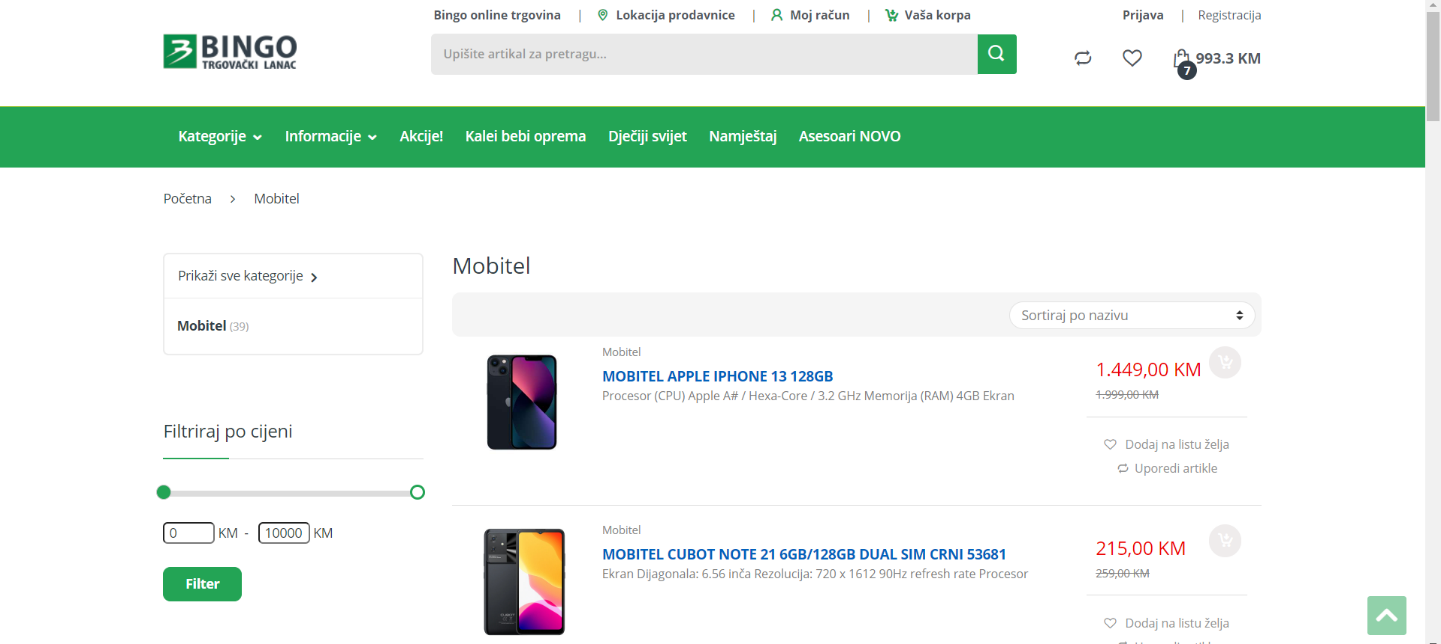
Description automatically generated

**Bug 11.:** Kada klinemo 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





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

