

Plan i realizacija funkcionalnog, automatizovanog, API i load testiranja web aplikacije

Članovi tima:

Mirnes Fehrić (vođa)

Emin Begić

Aldin Velić

Zana Beljuri

Sadržaj

1	Uvod	2
1.1	Opis testirane aplikacije	2
2	Zadatak 1	3
2.1	Funkcionalno testiranje (Black-box)	3
2.1.1	Evidencija u Azure DevOps-u i pristupi (Basic + Test Plans)	3
2.1.2	Identifikovani bugovi (evidentirani u Azure DevOps-u)	3
3	Zadatak 2	5
3.1	Automatizovano testiranje	5
3.1.1	SeleniumIDE testovi	5
3.1.2	Implementacija POM-a i Singleton-a (struktura rješenja)	6
3.1.3	Organizacija automatizovanih testova	6
3.1.4	NUnit testovi	6
3.1.5	Pokretanje NUnit testova	6
3.1.6	Azure DevOps repozitorij	6
4	Zadatak 3	7
4.1	API testiranje	7
4.1.1	Alat i pristup testiranju	7
4.1.2	Napomena o ponašanju API-ja	7
4.1.3	Tipovi validacija	7
4.1.4	Lista implementiranih API testova	8
4.1.5	Rezultati i kratka analiza	8
4.1.6	Pokretanje putem komande	9
4.1.7	Linkovi	9
5	Zadatak 4	10
5.1	Load testiranje	10
5.1.1	Postavka testa	10
5.1.2	Analiza vremena odziva (Response time)	10
5.1.3	TTFB analiza	11
5.1.4	Percentili odziva (stabilnost performansi)	11
5.1.5	Propusnost (Throughput) i mrežni promet	12
5.1.6	Zaključak	13
5.2	Pingdom (Website Speed Test)	13
5.2.1	Opšti rezultati testa	14
5.2.2	HAR timing (front-end milestone)	14
5.2.3	Sadržaj i HTTP zahtjevi	15
5.2.4	Preporuke Pingdom alata	15
5.3	Uporedba: Loadster vs Pingdom	16
5.3.1	Zaključak poređenja	16
6	Zaključak	17

1 Uvod

U okviru trećeg dijela projektnog zadatka izvršeno je sveobuhvatno testiranje javno dostupne web aplikacije **Automation Exercise** (<https://automationexercise.com/>). Aplikacija predstavlja demonstracionu e-commerce platformu (online prodavnicu) i koristi se kao praktičan poligon za učenje i provjeru različitih pristupa testiranju softvera jer sadrži veliki broj tipičnih funkcionalnosti koje se susreću u realnim web sistemima (rad sa korisničkim nalogima, katalog proizvoda, korpa, checkout) te dodatno nudi javno dostupne API endpoint-e za testiranje.

Cilj ovog projekta je primjena i povezivanje više metoda i tehnika testiranja, uključujući:

- manuelno funkcionalno testiranje (black-box) i evidentiranje testnih slučajeva,
- automatizovano UI testiranje (Selenium IDE i NUnit + Selenium),
- API testiranje (Postman),
- nefunkcionalno testiranje performansi (load test) i analiza metrika.

Testiranje je realizovano grupno uz korištenje alata Azure DevOps (Test Plans i repozitorij), Selenium IDE, NUnit Selenium framework, Postman i alata za load testiranje.

1.1 Opis testirane aplikacije

Automation Exercise je web aplikacija namijenjena simulaciji e-commerce sistema, sa jasno definisanim tokovima interakcije korisnika kroz više stranica i modula. Ključne funkcionalnosti aplikacije koje su bile relevantne za testiranje u ovom dijelu projekta uključuju:

- **Registracija i autentikacija:** kreiranje novog naloga, prijava/odjava korisnika, validacije pri unosu podataka.
- **Katalog proizvoda:** pregled liste proizvoda, filtriranje po kategorijama/brendovima, pretraga proizvoda.
- **Detalji proizvoda:** prikaz detalja artikla i dodavanje proizvoda u korpu.
- **Korpa i checkout:** pregled sadržaja korpe, izmjena količina, uklanjanje artikala, prelazak na checkout i završetak kupovine.
- **Kontakt forma:** slanje poruke putem Contact Us stranice.
- **API servis:** endpoint-i za dohvat proizvoda/brendova, pretragu proizvoda, verifikaciju prijave i upravljanje korisničkim računom.

Aplikacija je pogodna za testiranje jer sadrži i UI tokove tipične za web prodavnice, ali i javno dostupne API endpoint-e, što omogućava kombinovanje manualnog, automatizovanog, API i load testiranja u jedinstvenom izvještaju, uz mjerljive rezultate i jasne kriterije verifikacije.

Link na aplikaciju: <https://automationexercise.com/>

2 Zadatak 1

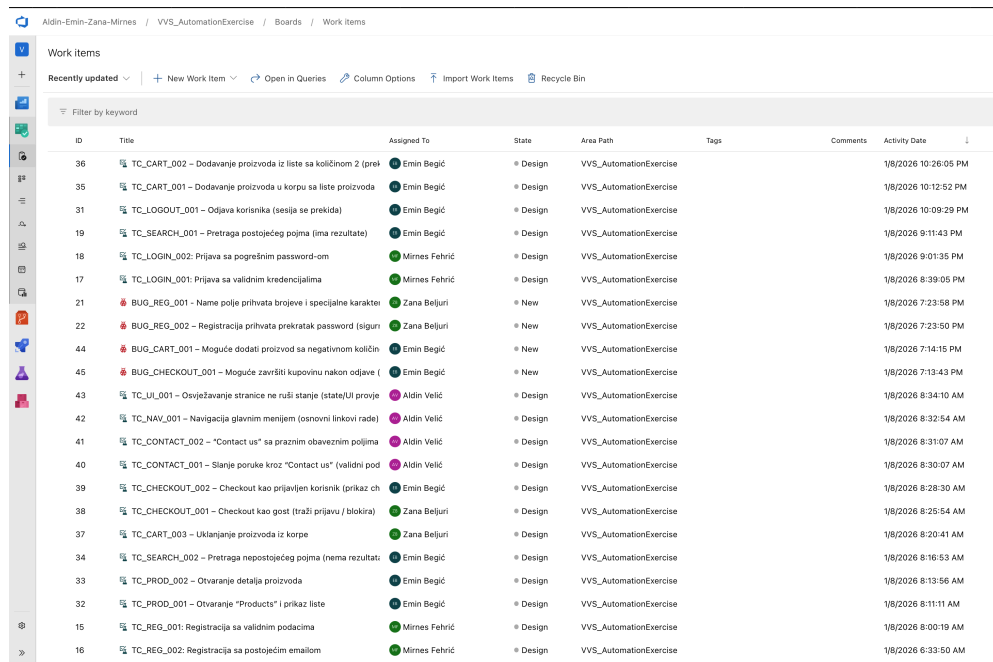
2.1 Funkcionalno testiranje (Black-box)

Za funkcionalno testiranje korištene su black-box tehnike testiranja:

- ekvivalentne klase,
- analiza graničnih vrijednosti,
- testiranje negativnih scenarija.

2.1.1 Evidencija u Azure DevOps-u i pristupi (Basic + Test Plans)

Svi testni slučajevi su uneseni kroz **Test Plans** (test suite-ovi i pojedinačni test case-ovi), sa svim potrebnim informacijama unutar work item-a. U okviru projekta je odgovornoj asistentici/demonstratorici dodijeljen **Basic + Test Plans** pristup, kako bi bio omogućen potpuni uvid u test planove, test case-ove, rezultate izvođenja i prijavljene bagove.



ID	Title	Assigned To	State	Area Path	Tags	Comments	Activity Date
36	TC_CART_002 – Dodavanje proizvoda iz liste sa količinom 2 (prek	Emin Begić	Design	VVS_AutomationExercise			1/8/2026 10:26:05 PM
35	TC_CART_001 – Dodavanje proizvoda u korpu sa liste proizvoda	Emin Begić	Design	VVS_AutomationExercise			1/8/2026 10:12:52 PM
31	TC_LOGOUT_001 – Odjava korisnika (sesija se prekida)	Emin Begić	Design	VVS_AutomationExercise			1/8/2026 10:09:29 PM
19	TC_SEARCH_001 – Pretraga postojećeg pojma (ima rezultate)	Emin Begić	Design	VVS_AutomationExercise			1/8/2026 9:11:43 PM
18	TC_LOGIN_002: Prijava sa pogrešnim password-om	Mirnes Fehrić	Design	VVS_AutomationExercise			1/8/2026 9:01:35 PM
17	TC_LOGIN_001: Prijava sa validnim kredencijalima	Mirnes Fehrić	Design	VVS_AutomationExercise			1/8/2026 8:39:05 PM
21	BUG_REQ_001 - Name polje prihvata brojeve i specijalne karaktere	Zana Beljuri	New	VVS_AutomationExercise			1/8/2026 7:23:58 PM
22	BUG_REQ_002 – Registracija prihvata prekratak password (ispun	Zana Beljuri	New	VVS_AutomationExercise			1/8/2026 7:23:50 PM
44	BUG_CART_001 – Moguće dodati proizvod sa negativnom količin	Emin Begić	New	VVS_AutomationExercise			1/8/2026 7:14:15 PM
45	BUG_CHECKOUT_001 – Moguće završiti kupovinu nakon odjave (Emin Begić	New	VVS_AutomationExercise			1/8/2026 7:13:43 PM
43	TC_UI_001 – Osvježavanje stranice ne ruši stanje (state/UI provje	Aldin Velić	Design	VVS_AutomationExercise			1/8/2026 8:34:10 AM
42	TC_NAV_001 – Navigacija glavnim menijem (osnovni linkovi rade)	Aldin Velić	Design	VVS_AutomationExercise			1/8/2026 8:32:54 AM
41	TC_CONTACT_002 – "Contact us" sa praznim obaveznim poljima	Aldin Velić	Design	VVS_AutomationExercise			1/8/2026 8:31:07 AM
40	TC_CONTACT_001 – Slanje poruke kroz "Contact us" (validni pod	Aldin Velić	Design	VVS_AutomationExercise			1/8/2026 8:30:07 AM
39	TC_CHECKOUT_002 – Checkout kao prijavljen korisnik (prikaz ch	Emin Begić	Design	VVS_AutomationExercise			1/8/2026 8:28:30 AM
38	TC_CHECKOUT_001 – Checkout kao gost (traži prijavu / blokira)	Zana Beljuri	Design	VVS_AutomationExercise			1/8/2026 8:25:54 AM
37	TC_CART_003 – Uklanjanje proizvoda iz korpe	Zana Beljuri	Design	VVS_AutomationExercise			1/8/2026 8:20:41 AM
34	TC_SEARCH_002 – Pretraga nepostojećeg pojma (nema rezultate	Emin Begić	Design	VVS_AutomationExercise			1/8/2026 8:16:53 AM
33	TC_PROD_002 – Otvaranje detalja proizvoda	Emin Begić	Design	VVS_AutomationExercise			1/8/2026 8:13:56 AM
32	TC_PROD_001 – Otvaranje "Products" i prikaz liste	Emin Begić	Design	VVS_AutomationExercise			1/8/2026 8:11:11 AM
15	TC_REQ_001: Registracija sa validnim podacima	Mirnes Fehrić	Design	VVS_AutomationExercise			1/8/2026 8:00:19 AM
16	TC_REQ_002: Registracija sa postojećim emailom	Mirnes Fehrić	Design	VVS_AutomationExercise			1/8/2026 6:33:50 AM

Slika 1: Test cases kao work items na Azure DevOps.

2.1.2 Identifikovani bugovi (evidentirani u Azure DevOps-u)

U toku manualnog funkcionalnog testiranja (Zadatak 1) identifikovane su sljedeće nepravilnosti u ponašanju aplikacije *Automation Exercise*, koje su evidentirane kao **Bug** work item-i na Azure DevOps-u. Bugovi su formulisani kroz standardnu strukturu: *preduslovi*, *koraci za reprodukciju*, *očekivani rezultat* i *stvarni rezultat*.

- **BUG_CART_001 – Moguće dodati proizvod sa negativnom količinom; subtotal/ukupno postaje negativno.**

Opis: Polje **Quantity** na stranici detalja proizvoda prihvata negativne vrijednosti, a sistem takvu vrijednost prosljeđuje u korpu. Posljedično, izračun *subtotal* i *total* može postati negativan, što je neložično i predstavlja funkcionalni defekt (i potencijalni sigurnosni/finansijski rizik).

Koraci: Products → View Product → u **Quantity** unijeti npr. **-2** → Add to cart → View Cart.

Očekivano: Sistem odbija negativnu količinu (minimalno 1) ili automatski koriguje na 1; subtotal/ukupno nikada ne smije biti negativno.

Stvarno: Sistem dozvoljava negativnu količinu i u korpi prikazuje negativan subtotal/ukupno.

- **BUG_CHECKOUT_001 – Moguće završiti kupovinu nakon odjave (direct URL) sa iznosom 0 i praznim podacima.**

Opis: Nakon što korisnik kopira URL checkout stranice i izvrši logout, moguće je direktnim otvaranjem (paste) checkout URL-a nastaviti proces kupovine bez aktivne sesije, pri čemu su podaci o korpi i dostavi prazni, a iznos narudžbe može biti 0. Uprkos tome, sistem omogućava *Place Order* i završetak kupovine.

Preduslov: Korisnik je bio prijavljen i imao barem 1 proizvod u korpi (u trenutku kopiranja checkout linka).

Koraci: Dodati proizvod u korpu → Proceed To Checkout → kopirati URL checkout stranice → Logout → zalijepiti checkout URL i otvoriti → Place Order → Pay and Confirm Order.

Očekivano: Nakon odjave sistem treba onemogućiti checkout (redirect na login ili poruka o isteku sesije); checkout ne smije biti moguć sa praznim podacima i iznosom 0.

Stvarno: Checkout je moguć i kupovina se može završiti nakon odjave putem direktnog URL pristupa.

- **BUG_REG_001 – Polje Name prihvata brojeve i specijalne karaktere.**

Opis: Forma za registraciju dozvoljava kreiranje korisničkog računa sa imenom koje sadrži brojeve i specijalne znakove, bez adekvatne validacije.

Koraci: Otvoriti <https://automationexercise.com/login> → u dijelu “New User Signup” unijeti **Name: 12345!@ # \$ %** i validan email → Signup → završiti registraciju.

Očekivano: Sistem odbija unos i prikazuje validacionu poruku da **Name** treba sadržavati samo slova (ili barem ne smije sadržavati brojeve/specijalne znakove).

Stvarno: Sistem dozvoljava registraciju i kreira račun sa navedenim imenom.

- **BUG_REG_002 – Registracija prihvata prekratak password (sigurnosna ranjivost).**

Opis: Sistem dozvoljava kreiranje korisničkog računa sa lozinkom kraćom od preporučenog minimuma (npr. 3 karaktera), bez validacije minimalne dužine, što predstavlja sigurnosni rizik.

Preduslov: Korisnik nije prijavljen; email mora biti jedinstven (nije ranije registrovan).

Koraci: Signup/Login → unijeti validan Name i jedinstven Email → Signup → u **Password** unijeti **123** → popuniti ostala obavezna polja → Create Account.

Očekivano: Sistem odbija kreiranje naloga i zahtijeva minimalnu dužinu lozinke (npr. ≥ 8 karaktera).

Stvarno: Sistem dozvoljava registraciju sa prekratkom lozinkom.

BUG 44
44 BUG_CART_001 – Moguće dodati proizvod sa negativnom količinom; subtotal/ukupno postaje negativno

Emin Begić

Status: New
Reason: Now
Assignee: VVS_AutomationExercise
Reporter: VVS_AutomationExercise

Repro Steps

Preduslovi:

- Nema (korisnik ne mora biti prijavljen).
- Korpa može biti prazna.

Koraci za reprodukciju:

- Otvoriti <https://automationexercise.com/>
- Kliknuti Products
- Kliknuti View Product na bilo kojem proizvodu
- U polje Quantity unijeti negativan broj (npr. -2)
- Kliknuti Add to cart
- Kliknuti View Cart

Očekivani rezultat:

- Sistem odbija negativnu količinu (min. 1) ili automatski koriguje na 1.
- Subtotal/ukupno nikad ne smije biti negativno.

Stvarni rezultat:

- Proizvod se dodaje sa negativnom količinom i subtotal/ukupno postaju negativni.

System Info

Komponenta/Područje: Cart / Product Details
Okruženje: macOS 10.15.7, Safari 16.6

Slika 2: Primjer evidencije jednog bug-a.

3 Zadatak 2

3.1 Automatizovano testiranje

Svi funkcionalni testni slučajevi su automatizovani korištenjem:

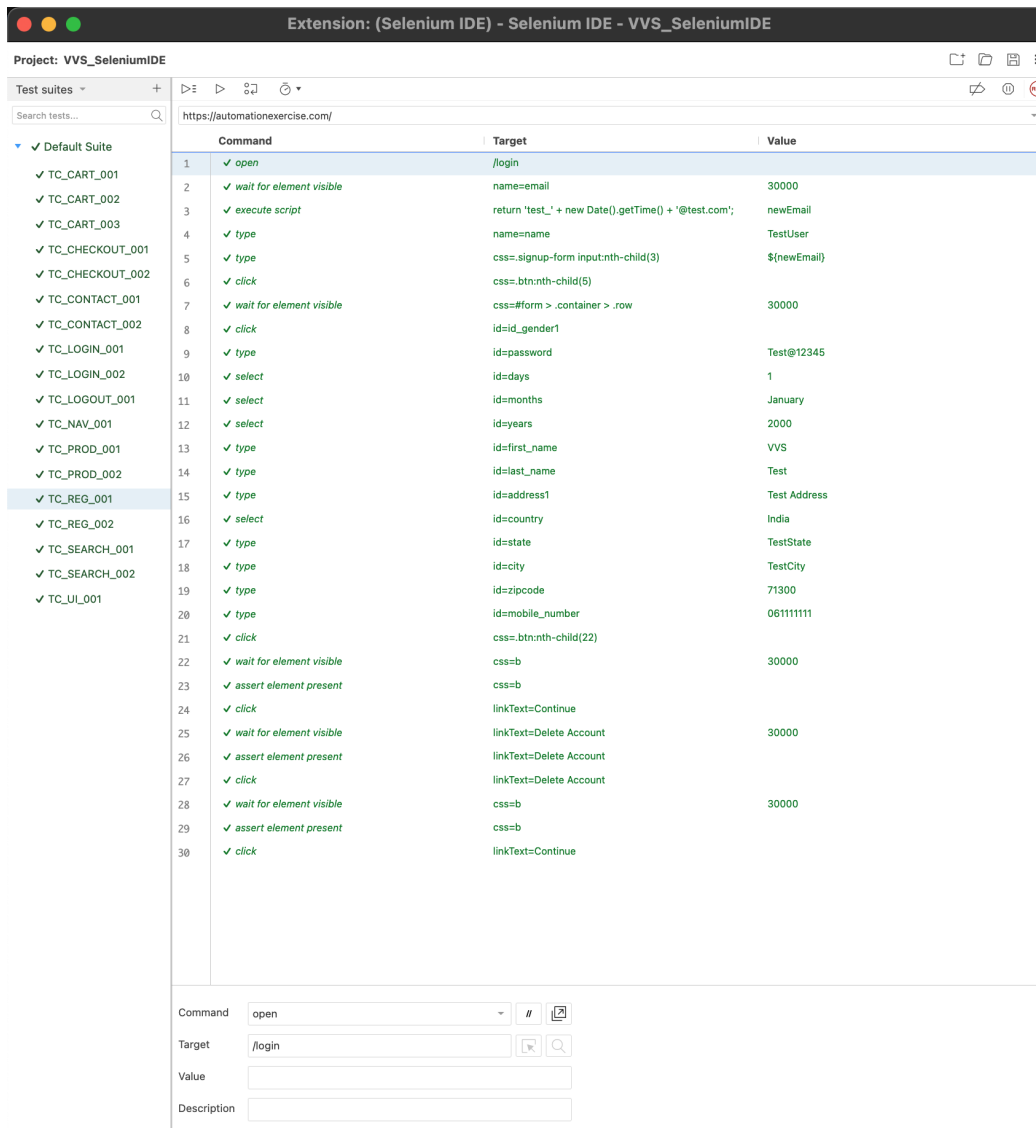
- Selenium IDE,
- NUnit Selenium automation framework.

U okviru testnog projekta primijenjeni su:

- Page Object Model (POM),
- Singleton dizajnerski obrazac.

Selenium testovi i NUnit projekat su pohranjeni na Azure DevOps repozitorij.

3.1.1 SeleniumIDE testovi



Project: VVS_SeleniumIDE

Test suites +

Search tests...

https://automationexercise.com/

Command	Target	Value
1 ✓ open	/login	
2 ✓ wait for element visible	name=email	30000
3 ✓ execute script	return 'test_' + new Date().getTime() + '@test.com';	newEmail
4 ✓ type	name=name	TestUser
5 ✓ type	css=signup-form input:nth-child(3)	\$(newEmail)
6 ✓ click	css=btn:nth-child(5)	
7 ✓ wait for element visible	css=#form > .container > .row	30000
8 ✓ click	id=id_gender1	
9 ✓ type	id=password	Test@12345
10 ✓ select	id=days	1
11 ✓ select	id=months	January
12 ✓ select	id=years	2000
13 ✓ type	id=first_name	VVS
14 ✓ type	id=last_name	Test
15 ✓ type	id=address1	Test Address
16 ✓ select	id=country	India
17 ✓ type	id=state	TestState
18 ✓ type	id=city	TestCity
19 ✓ type	id=zipcode	71300
20 ✓ type	id=mobile_number	061111111
21 ✓ click	css=btn:nth-child(22)	
22 ✓ wait for element visible	css=b	30000
23 ✓ assert element present	css=b	
24 ✓ click	linkText=Continue	
25 ✓ wait for element visible	linkText=Delete Account	30000
26 ✓ assert element present	linkText=Delete Account	
27 ✓ click	linkText=Delete Account	
28 ✓ wait for element visible	css=b	30000
29 ✓ assert element present	css=b	
30 ✓ click	linkText=Continue	

Command open // [A]

Target /login [R] [Q]

Value

Description

Slika 3: Prikaz testova unutar SeleniumIDE ekstenzije i primjer jednog testa.

3.1.2 Implementacija POM-a i Singleton-a (struktura rješenja)

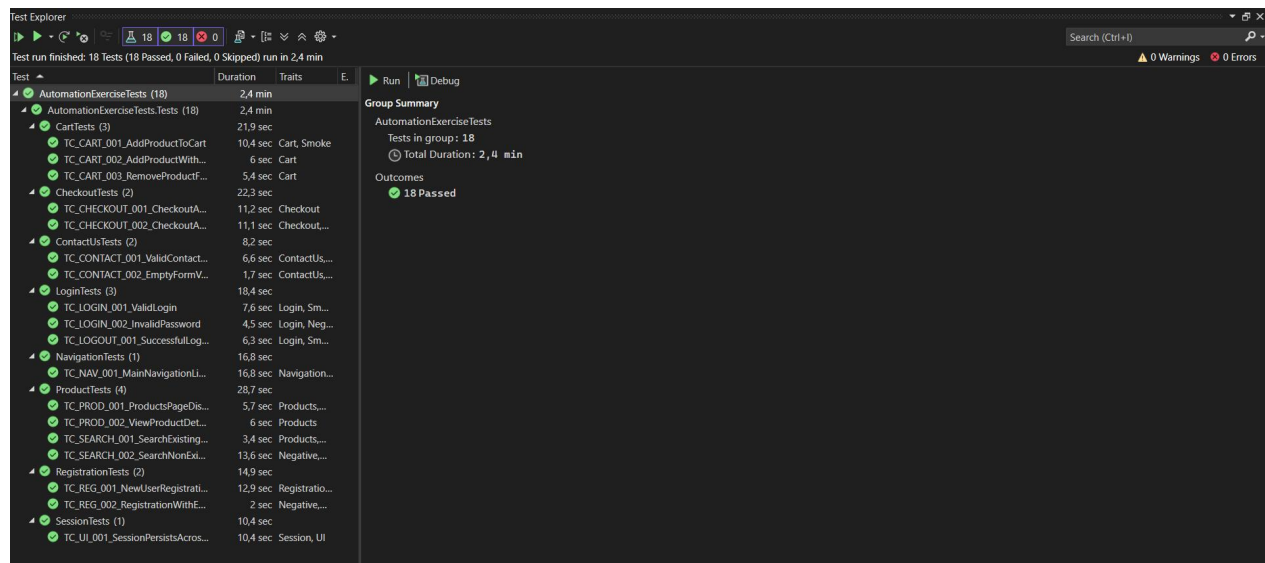
U NUnit projektu je korišten POM pristup kroz odvojene Page klase (npr. **HomePage**, **LoginPage**, **ProductsPage**, **ProductDetailPage**, **CartPage**, **CheckoutPage**, **PaymentPage**, **ContactUsPage**) kako bi selektori i akcije nad stranicama bili centralizovani i održivi. Zajednička funkcionalnost i pomoćne metode (npr. navigacija, čekanja i osnovne akcije) su objedinjene u bazičnim klasama (npr. **BasePage** i **BaseTest**).

Singleton obrazac je primijenjen kroz centralizovano upravljanje WebDriver instancom (npr. **DriverSingleton**), čime se osigurava konzistentno inicijalizovanje i gašenje driver-a, te izbjegavaju konflikti instanci tokom izvođenja testova.

3.1.3 Organizacija automatizovanih testova

Automatizovani testovi su grupisani po funkcionalnostima aplikacije kroz više test klasa (npr. **LoginTests**, **RegistrationTests**, **NavigationTests**, **ProductTests**, **CartTests**, **CheckoutTests**, **ContactUsTests**, **SessionTests**). Na ovaj način je postignuta preglednost (po modulima), lakše održavanje i jasnija veza između manualnih testnih slučajeva (Zadatak 1) i automatizovanih scenarija (Zadatak 2).

3.1.4 NUnit testovi



Slika 4: Prikaz NUnit testova.

3.1.5 Pokretanje NUnit testova

Testovi se mogu pokrenuti standardno kroz Visual Studio Test Explorer ili komandno:

```
dotnet test
```

3.1.6 Azure DevOps repozitorij

Link na Azure DevOps projekat:

https://dev.azure.com/Aldin-Emin-Zana-Mirnes/VVS_AutomationExercise

4 Zadatak 3

4.1 API testiranje

4.1.1 Alat i pristup testiranju

API testiranje je realizovano korištenjem alata **Postman**. Kreirana je Postman kolekcija koja sadrži ukupno **12 API zahtjeva**, pri čemu svaki zahtjev sadrži minimalno **4 validacije** (assertions) u okviru *Tests* (Post-response) skripti.

Testovi su organizovani tako da pokrivaju ključne funkcionalnosti aplikacije:

- pregled proizvoda i brendova,
- pretragu proizvoda (pozitivni i negativni scenariji),
- verifikaciju prijave korisnika (pozitivni i negativni scenariji),
- kreiranje, ažuriranje i brisanje korisničkog računa,
- dohvat detalja korisnika po email adresi.

4.1.2 Napomena o ponašanju API-ja

Tokom testiranja uočeno je da API u više slučajeva vraća **HTTP 200 OK** i kada se radi o grešci (npr. nepodržana metoda ili nedostajući parametar), dok se stvarni status greške nalazi u tijelu odgovora kroz polje **responseCode** (npr. 400, 404, 405). Zbog toga su validacije prilagođene realnom ponašanju sistema: provjerava se i HTTP status, ali je **responseCode** u JSON odgovoru ključni indikator uspjeha/neuspjeha.

4.1.3 Tipovi validacija

Korištene validacije (u kombinacijama, zavisno od endpoint-a):

- provjera HTTP status koda (najčešće 200),
- provjera **responseCode** u body-ju (200/201/400/404/405),
- provjera poruke u polju **message**,
- provjera strukture odgovora (postojanje polja **products**, **brands**, **user**),
- provjera da lista nije prazna (**products.length > 0**),
- provjera vremena odziva (npr. manje od 2000ms ili manje od 3000ms).

4.1.4 Lista implementiranih API testova

Tabela prikazuje testirane endpoint-e, scenario i očekivani rezultat.

#	Metoda	Endpoint	Scenario / očekivanje
1	GET	/api/productsList	Lista proizvoda; responseCode =200, products nije prazna.
2	POST	/api/productsList	Nepodržana metoda; responseCode =405, poruka "not supported".
3	GET	/api/brandsList	Lista brendova; responseCode =200, brands nije prazna.
4	PUT	/api/brandsList	Nepodržana metoda; responseCode =405, poruka "not supported".
5	POST	/api/searchProduct	Pretraga sa parametrom search_product ; responseCode =200, rezultati postoje.
6	POST	/api/searchProduct	Bez parametra search_product ; responseCode =400, poruka o nedostajućem parametru.
7	POST	/api/createAccount	Kreiranje korisnika; responseCode =201, poruka "User created!".
8	PUT	/api/updateAccount	Ažuriranje korisnika; responseCode =200, poruka "User updated!".
9	GET	/api/getUserDetailByEmail	Dohvat user detalja; responseCode =200, user.email odgovara query parametru.
10	POST	/api/verifyLogin	Validan email/password; responseCode =200, poruka "User exists!".
11	POST	/api/verifyLogin	Nevalidan password ili nepostojeći user; responseCode =404, poruka "User not found!".
12	DELETE	/api/deleteAccount	Brisanje korisnika; responseCode =200, poruka "Account deleted!".

4.1.5 Rezultati i kratka analiza

Tokom izvođenja testova potvrđeno je da API vraća očekivane strukture odgovora za pozitivne scenarije (liste proizvoda/brendova i user detalji), dok se greške za negativne scenarije (nedostajući parametri, nepodržane metode, nevalidan login) signaliziraju kroz **responseCode** i **message** u JSON odgovoru. Posebno je značajno uočeno odstupanje u odnosu na dokumentaciju gdje se umjesto HTTP 4xx često dobija HTTP 200, zbog čega je logika validacije bazirana primarno na **responseCode** polju u body-ju.

4.1.6 Pokretanje putem komande

Pored grafičkog interfejsa, testovi su izvršavani i putem **Postman CLI alata**, čime je omogućeno automatizovano pokretanje kompletne kolekcije iz komandne linije. Autentifikacija je izvršena korištenjem API ključa, nakon čega je kolekcija pokrenuta sljedećim komandama:

```
postman login --with-api-key PMAK-6960ff873741ba0001a84890-10ab7dea26ada7d0215839dfb3088266a8
```

```
postman collection run 48033619-be4a2c7a-147f-447c-9f78-9f2cb9e1e266
```

4.1.7 Linkovi

- Link na aplikaciju: <https://automationexercise.com/>
- Postman kolekcija u repozitoriju: <https://velicaldin252-4759827.postman.co/workspace/Aldin-Veli%C4%87's-Workspace~803ff573-83ee-4605-864a-ad8559c8a2de/collection/48033619-be4a2c7a-147f-447c-9f78-9f2cb9e1e266?action=share&creator=48033619>

5 Zadatak 4

5.1 Load testiranje

5.1.1 Postavka testa

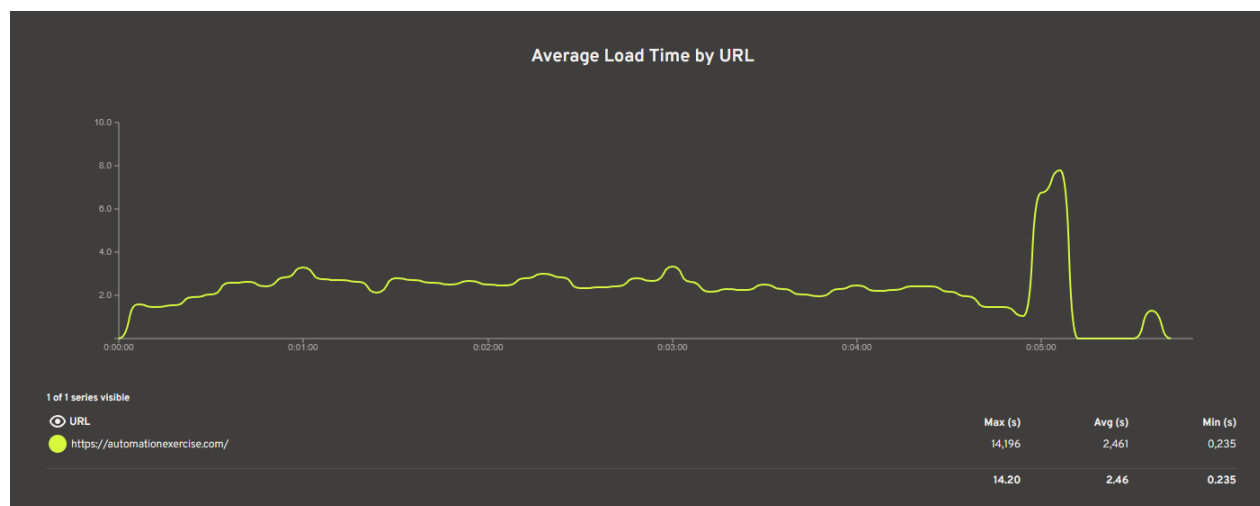
Load testiranje web aplikacije izvršeno je pomoću alata *Loadster* nad URL-om automationexercise.com.

Simulirano je ukupno **25 virtuelnih korisnika (bots)** tokom trajanja od **5 min 47 s**. Test je generisao ukupno **83 291 HTTP zahtjeva (hits)** kroz **1 112 iteracija**, pri čemu je broj grešaka iznosio **0**. Na osnovu ovih vrijednosti, test potvrđuje da je aplikacija tokom testiranog opterećenja ostala stabilna bez padova i bez evidentiranih grešaka.

Tablica 2: Sažetak load testa (Loadster)	
Metrika	Vrijednost
Trajanje testa	0:05:47
Broj korisnika (bots)	25
Ukupno iteracija	1 112
Ukupno zahtjeva (hits)	83 291
Broj grešaka	0
Prosječno vrijeme odgovora	2.46 s
P50 vrijeme odgovora	2.49 s
P90 vrijeme odgovora	3.34 s
Preuzeto (downloaded)	3.4 GB
Poslato (uploaded)	44.0 MB

5.1.2 Analiza vremena odziva (Response time)

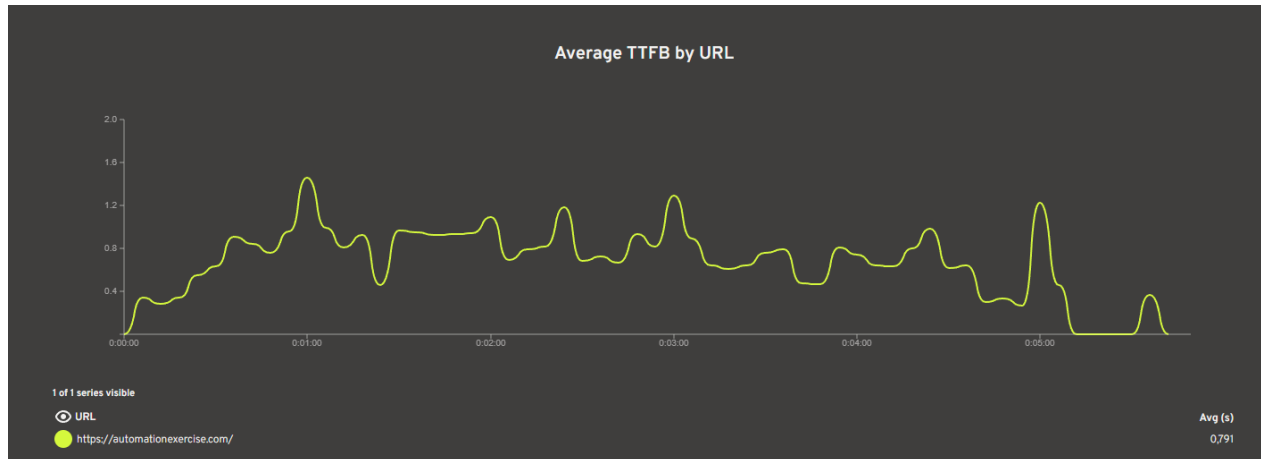
Na slici 5 prikazano je prosječno vrijeme učitavanja stranice (*Average Load Time by URL*). Prosječna vrijednost iznosi **2.46 s**, minimalno vrijeme je **0.235 s**, dok maksimalno doseže **14.20 s**. Tokom većeg dijela testa kriva je relativno stabilna u rasponu približno 2–3 s, što ukazuje na konzistentno ponašanje sistema pri opterećenju od 25 korisnika. Uočen je kratkotrajan skok oko završne faze testa (vrijeme ≈ 5 min), što može biti posljedica trenutne zagušenosti servera, mrežnih oscilacija ili završne faze ramp-down-a (smanjenje broja aktivnih korisnika). Kako nije zabilježena nijedna greška, ovaj skok predstavlja degradaciju performanse u kratkom intervalu, ali ne i funkcionalni problem.



Slika 5: Prosječno vrijeme učitavanja po URL-u (*Average Load Time by URL*).

5.1.3 TTFB analiza

Vrijeme do prvog bajta (*Time To First Byte* – *TTFB*) predstavlja indikator server-side latencije. Prosječna vrijednost iznosi **0.791 s** (slika 6). Tokom testa TTFB uglavnom oscilira oko 0.6–1.0 s, uz povremene pikove (oko 1.4 s), što ukazuje da dio ukupnog vremena odziva otpada na server processing i početno uspostavljanje veze. S obzirom da je ukupno prosječno vrijeme učitavanja 2.46 s, može se zaključiti da ostatak vremena otpada na prijenos sadržaja i renderovanje stranice na klijentu.



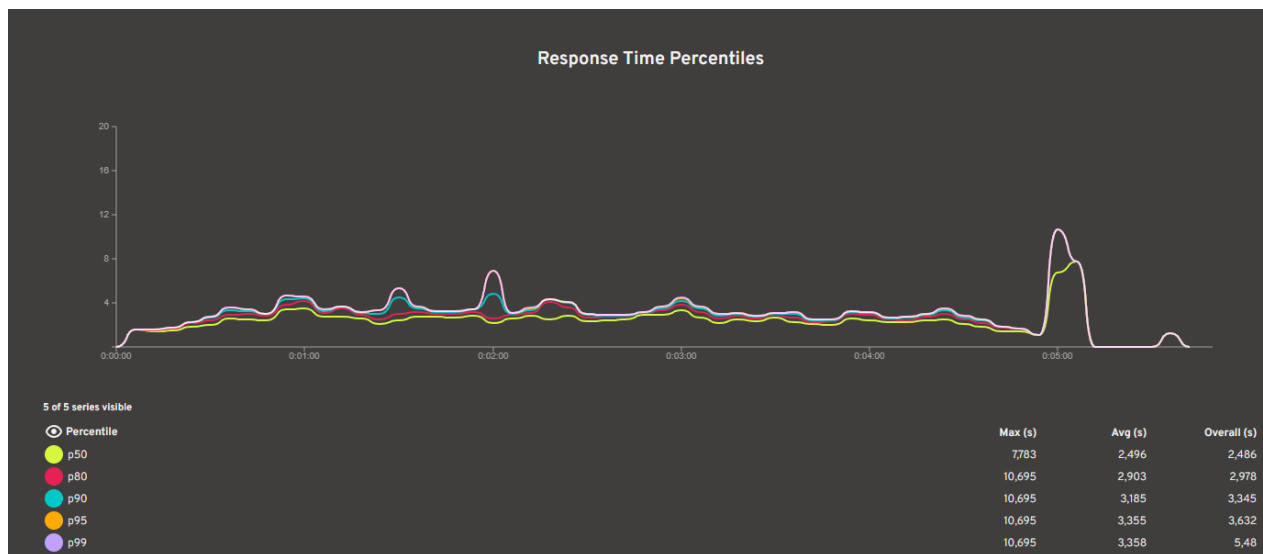
Slika 6: Prosječni TTFB po URL-u (Average TTFB by URL).

5.1.4 Percentili odziva (stabilnost performansi)

Percentili nude realističniji uvid u korisničko iskustvo od same prosječne vrijednosti. Dobijeni rezultati (slika 7) su:

- **P50 = 2.49 s**: polovina zahtjeva završava za 2.49 s ili brže (tipično korisničko iskustvo).
- **P90 = 3.34 s**: 90% zahtjeva završava za 3.34 s ili brže.
- **P95 = 3.63 s** i **P99 = 5.48 s**: mali broj zahtjeva (rep distribucije) ima primjetno veća vremena odziva.

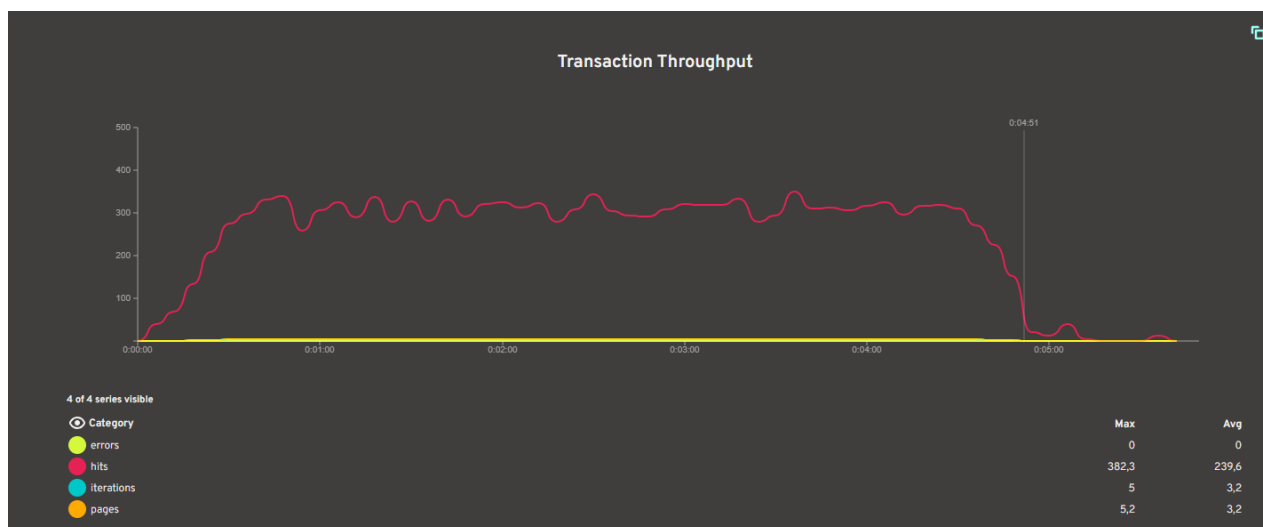
Ovo ukazuje da je većina zahtjeva stabilna i relativno brza, ali da postoje povremeni sporiji odgovori koji značajno podižu maksimalne vrijednosti (npr. max 14.20 s).



Slika 7: Percentili vremena odziva (Response Time Percentiles).

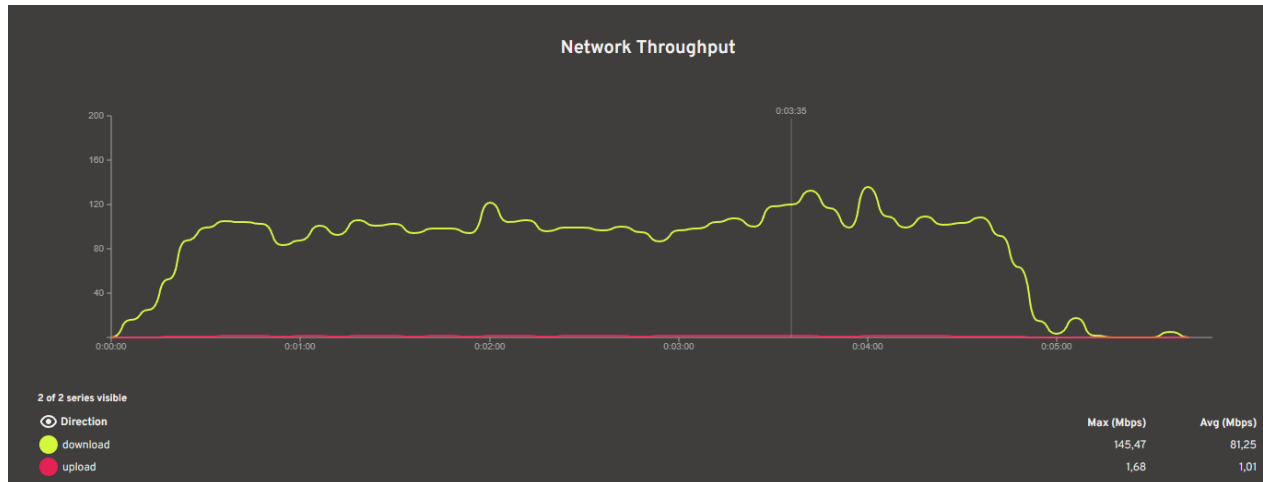
5.1.5 Propusnost (Throughput) i mrežni promet

Na slici 8 prikazana je propusnost (Transaction Throughput). Prosječna vrijednost **hits/s** iznosi **239.6**, dok maksimalna dostiže **382.3 hits/s**. Tokom centralnog dijela testa throughput je relativno stabilan, a opada pri kraju testa, što odgovara ramp-down fazi (smanjenje broja aktivnih korisnika).



Slika 8: Transaction Throughput (hits/s) kroz vrijeme.

Mrežni promet potvrđuje intenzivno preuzimanje sadržaja (dominantan download). Prosječna brzina preuzimanja iznosi **81.25 Mbps**, a slanja **1.01 Mbps** (slika 9). Ukupno je preuzeto približno **3.4 GB**, a poslato **44 MB**. Ovakav odnos (download \gg upload) je očekivan za web aplikacije gdje klijent većinom preuzima resurse.



Slika 9: Network Throughput (download i upload) tokom testa.

5.1.6 Zaključak

Load test sa 25 virtuelnih korisnika pokazuje da aplikacija automationexercise.com u posmatranom intervalu radi stabilno i bez grešaka (**0 errors**). Prosječno vrijeme učitavanja iznosi **2.46 s**, dok percentili (P90 = 3.34 s, P95 = 3.63 s) pokazuju da većina korisnika ima konzistentno iskustvo. Postoji manji broj sporijih zahtjeva (P99 = 5.48 s i maksimalno 14.20 s), što ukazuje na povremene oscilacije performansi, ali bez funkcionalnih problema. Ukupno posmatrano, sistem se ponaša stabilno za ovakav nivo opterećenja, uz povremene pikove koji bi se u realnom okruženju mogli dodatno analizirati povećanjem broja korisnika ili produženjem trajanja testa.

5.2 Pingdom (Website Speed Test)

Pingdom alat korišten je za izvođenje **single-user sintetičkog testa** nad automationexercise.com.

Za razliku od Loadster testa koji simulira paralelno opterećenje više korisnika, Pingdom mjeri performanse iz perspektive jednog korisnika i fokusira se prvenstveno na **front-end metrike**, strukturu resursa i potencijalne optimizacije.

Pingdom test pruža sljedeće ključne informacije:

- performansnu ocjenu stranice (performance grade),
- ukupno vrijeme učitavanja stranice (load time),
- veličinu stranice i broj HTTP zahtjeva,
- preporuke za optimizaciju (kompresija, caching, DNS, broj zahtjeva),
- detaljan waterfall dijagram i breakdown po domenima i tipovima sadržaja.

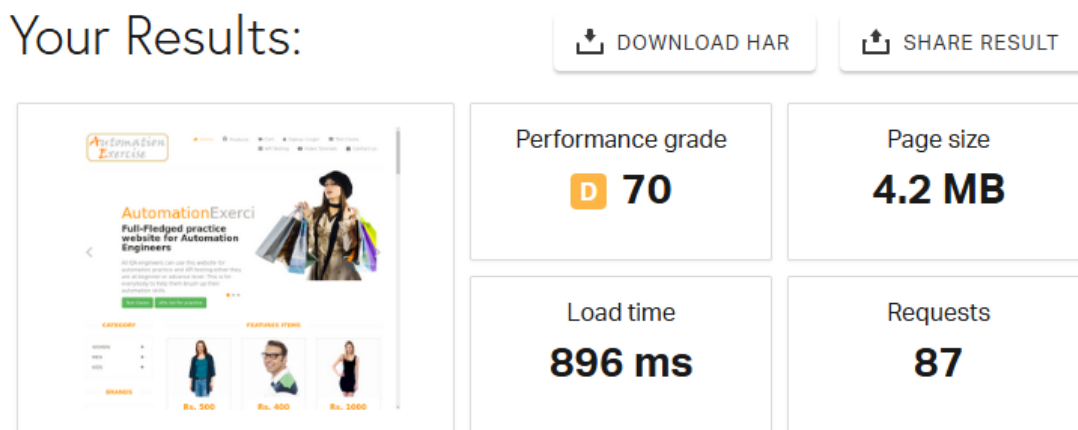
Rezultati Pingdom testa prikazani su u Tabeli 3, dok je vizuelni prikaz ključnih metrika dat na Slici 10.

Tablica 3: Pingdom sažetak

Metrika	Vrijednost
Performance grade	D (70)
Load time	896 ms
Page size	4.2 MB
Requests	87

5.2.1 Opšti rezultati testa

Pingdom mjerenje pokazuje da je osnovno vrijeme učitavanja stranice relativno kratko (**896 ms**), što ukazuje na brzu inicijalnu isporuku sadržaja za jednog korisnika. Međutim, performansna ocjena (D – 70) sugerise postojanje značajnih prostora za optimizaciju.



Slika 10: Pingdom: performansna ocjena, vrijeme učitavanja, veličina stranice i broj zahtjeva.

5.2.2 HAR timing (front-end milestone)















Iz HAR zapisa izdvojeni su ključni vremenski događaji koji opisuju tok učitavanja stranice:

- **DOMContentLoaded (DCL)** = 537 ms, što znači da je DOM struktura brzo dostupna,
- **onLoad** = 896 ms, kada su glavni resursi učitani,
- **Fully Loaded** = 1808 ms, što ukazuje na kasnije učitavanje dodatnih ili asinhronih resursa.

Ovi rezultati pokazuju da korisnik relativno brzo dobija osnovni sadržaj stranice, dok se dio resursa (npr. slike, skripte trećih strana) učitava i nakon inicijalnog prikaza.

5.2.3 Sadržaj i HTTP zahtjevi

Struktura sadržaja i raspodjela zahtjeva po tipu resursa i domenima prikazana je na Slici 12.

Content size by content type			Requests by content type		
CONTENT TYPE	PERCENT	SIZE	CONTENT TYPE	PERCENT	REQUESTS
 Image	82.06%	3.5 MB	 Image	47.67%	41
 Script	11.15%	471.0 KB	 Script	20.93%	18
 Font	4.16%	175.6 KB	 XHR	12.79%	11
 CSS	1.08%	45.8 KB	 CSS	9.30%	8
 XHR	0.86%	36.4 KB	 HTML	5.81%	5
 HTML	0.64%	27.0 KB	 Font	2.33%	2
 Error	0.05%	2.3 KB	 Error	1.16%	1
Total	100.00%	4.2 MB	Total	100.00%	86

Content size by domain			Requests by domain		
CONTENT TYPE	PERCENT	SIZE	CONTENT TYPE	PERCENT	REQUESTS
automationexercise.com	86.29%	3.6 MB	automationexercise.com	64.37%	56
pagead2.googlesyndication.com	7.44%	308.3 KB	fundingchoicesmessages.google...	16.09%	14
fonts.gstatic.com	3.11%	128.9 KB	pagead2.googlesyndication.com	6.90%	6
fundingchoicesmessages.google...	2.03%	84.2 KB	ep2.adtrafficquality.google	3.45%	3
ep1.adtrafficquality.google	0.34%	13.9 KB	googleads.g.doubleclick.net	2.30%	2
ep2.adtrafficquality.google	0.31%	12.9 KB	ep1.adtrafficquality.google	2.30%	2
other	0.48%	19.8 KB	other	4.60%	4
Total	100.00%	4.1 MB	Total	100.00%	87

Slika 11: Pingdom: breakdown po tipu sadržaja i domenima (veličina i broj zahtjeva).

Analiza pokazuje da **slike dominantno utiču na performanse stranice**, jer čine **82.06% ukupne veličine** (3.5 MB) i generišu **41 HTTP zahtjev**. Skripte (JavaScript) čine dodatnih 11.15% (471 KB), dok su CSS i fontovi relativno mali, ali povećavaju broj paralelnih zahtjeva.

Po domenima, većina resursa dolazi sa osnovnog domena **automationexercise.com**, međutim prisutan je i značajan broj zahtjeva prema third-party servisima (npr. Google Ads, DoubleClick i srodni domeni). Ovakvi eksterni zahtjevi tipično povećavaju ukupni broj HTTP konekcija, DNS lookup vrijeme i varijabilnost odziva.

5.2.4 Preporuke Pingdom alata

Pingdom automatska analiza identifikovala je nekoliko kritičnih tačaka optimizacije:

- **Compress components with gzip** (F),
- **Add Expires headers** (F),
- **Make fewer HTTP requests** (F),
- **Use cookie-free domains** (F),
- **Reduce DNS lookups** (C).

Improve page performance

GRADE	SUGGESTION	
F 0	Compress components with gzip	▼
F 0	Use cookie-free domains	▼
F 0	Add Expires headers	▼
F 48	Make fewer HTTP requests	▼
C 75	Reduce DNS lookups	▼
A 100	Avoid empty src or href	▼
A 100	Put JavaScript at bottom	▼

Slika 12: Pingdom: Preporuke

Ove preporuke ukazuju da bi se performanse mogle značajno unaprijediti uvođenjem agresivnijeg caching-a, kompresije statičkih resursa i smanjenjem broja zahtjeva, naročito za slike i eksterni JavaScript sadržaj.

5.3 Uporedba: Loadster vs Pingdom

Ovo poređenje je često traženo jer alati mjere različite stvari.

Tablica 4: Loadster vs Pingdom (ključne razlike i interpretacija)

Kriterij	Pingdom (speed test)	Loadster (load test)
Tip testa	Jedan klijent (sintetički), fokus na page load	Više paralelnih korisnika (25 botova), fokus na skaliranje
Šta mjeri	Učitavanje stranice + waterfall + preporuke	Odziv pod opterećenjem: latencije, throughput, greške
Rezultat za ovu stranicu	onLoad \approx 0.896 s, Fully Loaded \approx 1.808 s	Avg response 2.46 s, P90 3.34 s, P99 (overall) 5.48 s
Kako tumačiti	Stranica je brza u idealnom scenariju (single-user) ali ima puno requestova (87) i težak image payload	Pod opterećenjem latencija raste 2–3.5 s za većinu zahtjeva, a repovi (P95/P99) idu preko 3.6–5.5 s
Praktični zaključak	Optimizacije (gzip, caching, manje requestova) će ubrzati i single-user i pomoći pod opterećenjem	Ako ciljamo realne korisnike, treba smanjiti repove (P95/P99), posebno kroz caching i redukciju third-party poziva

5.3.1 Zaključak poređenja

Pingdom pokazuje da je stranica *u jednom mjerenju* sposobna da se učita ispod 1 s (onLoad), dok Loadster pokazuje da *u uslovima opterećenja* prosječni odziv raste na oko 2.46 s, a sporiji percentili (P95/P99) postaju značajni. Dakle, oba rezultata su istovremeno tačna, ali opisuju različite scenarije.

Na osnovu oba alata:

- Stranica se u idealnom scenariju učitava brzo (Pingdom onLoad < 1 s), ali ima dosta zahtjeva i relativno velik payload (4.2 MB, dominiraju slike).
- Pod paralelnim opterećenjem (25 botova) prosječne latencije rastu (avg 2.46 s) i pojavljuju se sporiji repovi (P99 overall 5.48 s), što može negativno uticati na korisničko iskustvo.

- Najisplativije optimizacije su: gzip/brotli, cache headers (Expires/Cache-Control), smanjenje broja requestova, optimizacija slika (kompresija + moderni formati), te kontrola third-party resursa (ads/trackers).

6 Zaključak

U ovom projektu uspješno su primijenjene različite tehnike testiranja softvera. Korištenjem kombinacije manualnog, automatizovanog, API i load testiranja ostvarena je dobra pokrivenost funkcionalnosti aplikacije i identifikovani su potencijalni problemi.