

# Funkcionalno testiranje i testiranje opterećenja web sustava prikaza vijesti

---

**Maras, Ines**

**Master's thesis / Diplomski rad**

**2020**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:200:903047>

*Rights / Prava:* [In copyright](#)/[Zaštićeno autorskim pravom.](#)

*Download date / Datum preuzimanja:* **2024-11-24**

*Repository / Repozitorij:*

[Faculty of Electrical Engineering, Computer Science and Information Technology Osijek](#)



**SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU**

**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I  
INFORMACIJSKIH TEHNOLOGIJA**

**Sveučilišni diplomski studij**

**FUNKCIONALNO TESTIRANJE I TESTIRANJE  
OPTEREĆENJA WEB SUSTAVA PRIKAZA VIJESTI**

**Diplomski rad**

**Ines Maras**

**Osijek, 2020.**

## SADRŽAJ

1. UVOD.....	1
1.1. Zadatak diplomskog rada .....	2
2. PREGLED POSTOJEĆIH RIJEŠENJA FUNKCIONALNOG TESTIRANJA I TESTIRANJA OPTEREĆENJA WEB SUSTAVA .....	3
2.1. Pregled postojećih rješenja funkcionalnog testiranja web sustava .....	3
2.2. Pregled postojećih rješenja nefunkcionalnog testiranja - testiranja opterećenja web sustava ...	4
3. TESTIRANJE PROGRAMSKE PODRŠKE WEB SUSTAVA .....	7
3.1. Tipovi testiranja programske podrške.....	9
3.1.1. Ručno testiranje programske podrške.....	9
3.1.2. Automatsko testiranje programske podrške.....	10
3.2. Vrste testova za web sustav.....	12
3.2.1. Funkcionalno testiranje web sustava.....	13
3.2.2. Nefunkcionalno testiranje -test opterećenja web sustava .....	14
4. WEB SUSTAV ZA PRILAGODLJIVI VIŠEKRITERIJSKI PRIKAZ VIJESTI IZ RAZLIČITIH IZVORA .....	16
4.1. Dizajn web sustava prikaza vijesti.....	16
4.2. Svrha web sustava prikaza vijesti .....	17
4.3. Arhitektura web sustava prikaza vijesti .....	17
4.4. Rješenje web sustava prikaza vijesti.....	18
4.4.1. Prednosti web sustava prikaza vijesti .....	18
4.4.2. Nedostaci web sustava prikaza vijesti .....	18
5. KORIŠTENI ALATI ZA TESTIRANJE WEB SUSTAVA PRIKAZA VIJESTI .....	20
5.1. JMeter.....	20
5.1.1. Proxy poslužitelj.....	21
5.1.2. BlazeMeter .....	22
5.2. Visual Studio .....	22
6. PROVOĐENJE FUNKCIONALNOG TESTIRANJA .....	24
6.1. Priprema za funkcionalno testiranje.....	24
6.1.1. Definiranje testnog plana.....	24
6.1.2. Definiranje testnih scenarija i testnih slučajeva .....	25
6.1.3. Definiranje očekivanih rezultata .....	27
6.2. Provedba funkcionalnog testiranja.....	28
7. PROVOĐENJE TESTA OPTEREĆENJA WEB SUSTAVA .....	32
7.1. Priprema za provođenje testa opterećenja web sustava .....	32
7.1.1. Definiranje testnih scenarija.....	32

7.1.2. Definiranje testne grupe.....	33
7.1.3. Definiranje očekivanih rezultata .....	34
7.2. Provedba testa opterećenja .....	35
7.2.1. Provedba testa opterećenja u alatu JMeter.....	35
7.2.2. Provedba testa opterećenja u alatu Visual studio .....	42
8. REZULTATI FUNKCIONALNOG TESTIRANJA I TESTIRANJA OPTREĆENJA WEB SUSTAVA PRIKAZA VIJESTI.....	48
8.1. Rezultati funkcionalnog testiranja web sustava prikaza vijesti .....	48
8.2. Rezultati testa opterećenja web sustava prikaza vijesti .....	49
8.2.1. Rezultati dobiveni u programskom alatu JMeter i Proxy .....	49
8.2.2. Rezultati dobiveni u programskom alatu JMeter i BlazeMeter .....	56
8.3. Rezultati dobiveni u programskom alatu Visual Studio .....	64
8.4. Usporedba korištenih alata .....	65
9. ZAKLJUČAK .....	67
LITERATURA .....	68
SAŽETAK.....	71
ABSTRACT .....	72
ŽIVOTOPIS.....	73
PRILOZI.....	74

## 1. UVOD

Programska podrška je postala svakodnevni dio ljudskih života i većina ljudi danas donosi svoje odluke na temelju pretraživanja podataka na internetu. Korisnici svakodnevno koriste razne web sustave koji im olakšavaju posao, pomažu oko organizacije i putem interneta doznaju razne informacije, a uz to pregledavaju vijesti kako bi bili upućeni u događaje u svijetu i svojoj zemlji. Kako bi se omogućilo svakodnevno korištenje web sustava i osigurali korisnicima njegov kvalitetan rad, osim izrade web sustava potrebno je provesti testiranje tj. provjeriti ispravnost, funkcionalnosti, sigurnost, opterećenje i dizajn web sustava, a takav postupak provjere naziva se web testiranje.

Zadatak ovog diplomskog rada je provesti funkcionalno i nefunkcionalno testiranje na diplomskom radu „Programsko rješenje web sustava za prilagodljivi višekriterijski prikaz vijesti iz različitih izvora“ [1]. Provedbom funkcionalnog testiranja na ovom web sustavu dobivaju se rezultati ispravnosti rada funkcija te se analiziraju dobivenih rezultati i uspoređuju s očekivanim rezultatima. U slučaju da rezultati nisu jednaki očekivanim rezultatima, potrebno je prijaviti pogrešku i paziti da se što prije ispravi kako ne bi došla do korisnika. U ovom slučaju potrebno je provjeriti učitavanje stranice i zadanih portala, otvaranje svakog portala pojedinačno, ispravnost rada komponenti web sustava, ispravnost rada prikazivanja i pretraživanja vijesti. Osim provjere funkcionalnosti, potrebno je provjeriti i opterećenje sustava tj. potrebno je provjeriti performanse web sustava prikaza vijesti, jer će ga koristiti više korisnika u isto vrijeme. Potrebno je opteretiti web sustav sa što više korisnika, korisnicima zadati različite zadatke, a zatim je potrebno dobro i pravilno analizirati rezultate kako bi se došlo do ispravnih zaključaka o radu web sustava prikaza vijesti.

U poglavlju 2 predstavljen je pregled postojećih rješenja za funkcionalno testiranje i pregled rješenja za nefunkcionalno testiranje tj. za provedbu testa opterećenja. Osim toga, opisano je i postupak provedbe jednog i drugog teksta. Poglavlje 3 sadrži metodologiju testiranja, opis što je testiranje, vrste i tipove testiranja. U poglavlju 4 objašnjen je web sustav na kojem će se testiranje provoditi, objašnjen je dizajn, arhitektura, svrha i rješenje samog web sustava. Korišteni alati za provedbu testiranja opisani su u poglavlju 5, dok su u poglavljima 6 i 7 provedeni testovi tj. u poglavlju 6 je opisana provedba funkcionalnog testiranja dok je u poglavlju 7 opisana provedba nefunkcionalnog testiranja. Poglavlje 8 sadrži analizu dobivenih rezultata testiranja te usporedbu rezultata i zaključak testiranja.

## **1.1. Zadatak diplomskog rada**

U diplomskom radu treba opisati metodologiju, razine i vrste testiranja s posebnim naglaskom na funkcionalno i testiranje opterećenja programske podrške. Također, potrebno je opisati specifičnosti i zahtjeve navedenih testiranja za web sustav prikaza vijesti, uključujući pripremu, provedbu i analizu rezultata testiranja, alate koji će se koristiti za testiranje i razradu testnih koraka.

U praktičnom dijelu rada potrebno je provesti funkcionalno testiranje i testiranje opterećenja navedenog web sustava, a rezultate testiranja prikladno analizirati, interpretirati i navesti zahtjeve za potrebnim izmjenama programskog rješenja web sustava.

## **2. PREGLED POSTOJEĆIH RIJEŠENJA FUNKCIONALNOG TESTIRANJA I TESTIRANJA OPTEREĆENJA WEB SUSTAVA**

Prije provedbe testiranja na web sustavu prikaza vijesti, potrebno je pregledati postojeća rješenja testiranja web sustava, kako je već spomenuto, u ovom diplomskom radu provodit će se dvije vrste testova, a to su funkcionalno testiranje i nefunkcionalno testiranje tj. testiranje opterećenja web sustava. Funkcionalno testiranje provodi se nešto češće od testiranja opterećenja jer kod bilo kojeg tipa web sustava potrebna je provjera funkcionalnosti, dok se test opterećenja provodi u slučaju kad web sustav koristi više korisnika u isto vrijeme pa se mjere performanse sustava i prati ponašanje sustava pri opterećenju.

U nastavku je navedeno nekoliko postojećih rješenja provedbe funkcionalnog testiranja i testiranja opterećenja. Neka rješenja će biti primijenjena u provedbi testiranja web sustava prikaza vijesti, dok će neka biti samo spomenuti za dobitak ideje. Također, navedeno je i nekoliko rješenja za provedbu testa opterećenja te će se većina ideja uzeti u obzir za razliku od situacije kod funkcionalnog testiranja.

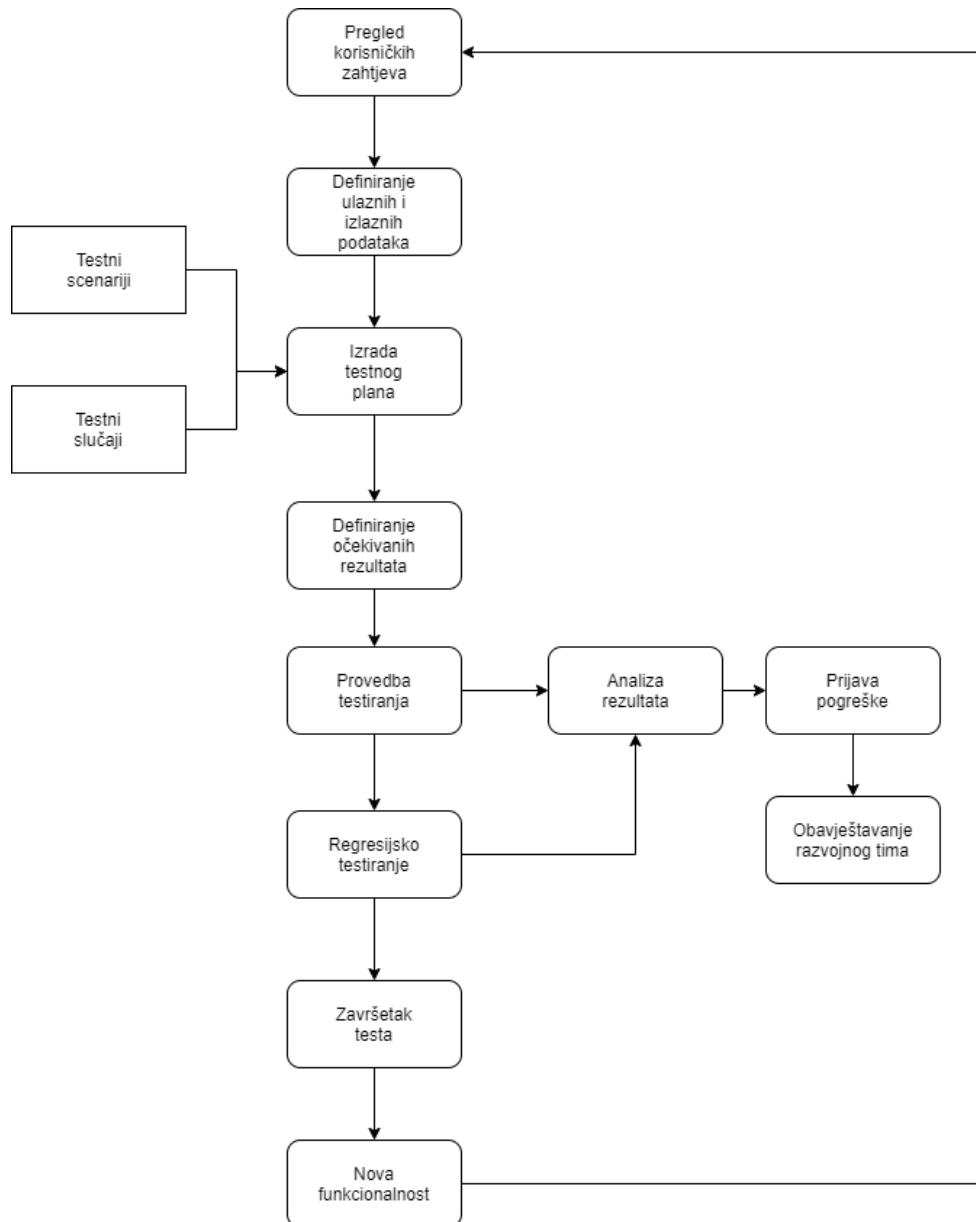
### **2.1. Pregled postojećih rješenja funkcionalnog testiranja web sustava**

Funkcionalno testiranje je jedno od najbitnijih testiranja web sustava. Prije pokretanja postupka testiranja potrebno je imati definirane zahtjeve. Zahtjevi web sustava se razlikuju od sustava do sustava, dok je postupak funkcionalnog testiranja praktički jednak za sve web sustave. U knjigama kao i na internetu, postupak funkcionalnog testiranja započinje pregledom zahtjeva pa slijedi raspis test plana, a zatim raspis testnih scenarija, testnih slučajeva, testnih koraka i očekivanih rezultata testiranja.

Korake provedbe funkcionalnog testiranja mogu se vidjeti na slici 2.1. Prije svega potrebno je razumjeti definirane funkcionalne zahtjeve, ulazne i izlazne podatke, raspis testnih slučajeva, kreiranje ulaznih podataka i definiranje očekivanih rezultata, izvršavanje testnih slučajeva pa sve do provedbe regresijskog testiranja i slanja testnog izvještaja razvojnom timu. Također iste korake provedbe funkcionalnog testiranja web sustava mogu se pronaći na online web aplikaciji za testiranje Guru99 gdje se funkcionalno testiranje može provesti online [2].

U većini literature opisana su pitanja koja razvojni tim treba postaviti kako bi uspješno proveo funkcionalno testiranja. Neka od pitanja su: Radi li to stvarno? Može li korisnik raditi ono što smo očekivali? Postoje li načini kako bi nešto radilo, a da nismo uzeli u obzir? [3]. Postavljanjem tih pitanja i još mnogih kao što su: Je li bi korisnik išao ovom putanjom po web sustavu? Što može sve vidjeti i dirati te što će dobiti za odgovor? Kad bi bilo odgovora na sva pitanja i vodilo se po

koracima navedenih u literaturi, za rezultata bi bila dobivena jako detaljna provedba funkcionalnog testiranja web sustava te bi se obuhvatila sva funkcionalnost web sustava. Teško je navesti već postojeća rješenja funkcionalnog testiranja kada se ono provodi na većini web sustava i jedan je od najvažnijih testova web sustava današnjice.



Sl. 2.1. Postupak provedbe funkcionalnog testiranja

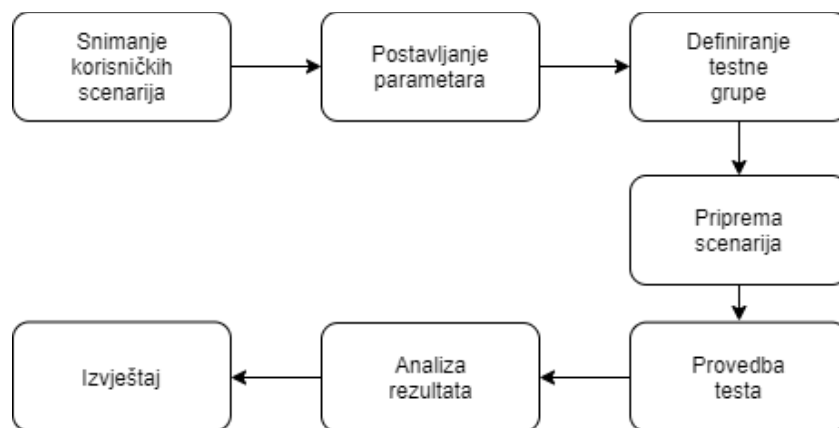
## 2.2. Pregled postojećih rješenja nefunkcionalnog testiranja - testiranja opterećenja web sustava

Nefunkcionalno testiranje tj. test opterećenja web sustava jedan je od rjeđih testova koji se provode pa zbog toga ima i manje postojećih rješenja po kojima se može voditi provedba takvog testa. Dvije su bitne web stranice po kojima se može voditi za provedbu testa opterećenja, a to su:



„Quickstart: Create a load test project“ na Microsoftovoj stranici gdje je opisan detaljan postupak provođenja testa opterećenja u alatu Visual Studio [4]. Također na stranici „Guru99: How to Use JMeter for Performance & Load Testing“ gdje opisan detaljan postupak provođenja testa opterećenja pomoću alata JMeter [5].

U ovom diplomskom radu, provedba testiranja će se voditi po oba postupka i pristup testu će biti jednak kao pristupi na spomenutim web stranicama. Još puno više rješenja testiranja opterećenja web sustava postoje međutim dovoljno je ih je samo spomenuti jer test se provodi po istim koracima, samo njegovi parametri se mijenjaju. Na slici 2.2 prikazani su koraci provođenja testa opterećenja kojih se drže test inženjeri prilikom njegovoj provođenja kako bi na kraju dobili realne i očekivane rezultate.



Sl. 2.2. Postupak provedbe testa opterećenja

Danas je jako bitno provoditi test opterećenja zbog opterećenosti sustava. Jedan od primjera potrebe korištenja ove vrste testa je rad web sustava koji su inače opterećeni na Crni petak. Prije samog Crnog petka, potrebno je provjeriti koliko korisnika sustav može podnijeti, kako bi se smanjila mogućnost padanja poslužitelja. Također je svaki sustav dobro opteretiti zbog provjere vremena odziva kako korisnik ne bi dugo mogao čekati odgovor poslužitelja.

Provedba testa opterećenja je težak zadatak i zahtjeva dobro razumijevanje web sustava na kojem se provodi test prema Visseru. Kako bi se test opterećenja pravilno proveo potrebno je imati ispravni web sustav, a rezultati se trebaju pažljivo analizirati kako bi se otkrile pogreške u web sustavu tijekom provođenja testa opterećenja [6].

Tijekom provođenja testa opterećenja potrebno je navesti metrike koje će se mjeriti kao što je mjerenje vremena odziva koje uključuje: ukupno vrijeme potrebno nakon što klijent pošalje zahtjev i dok ne dobije odgovor, vrijeme uključuje učitavanje stranice, vrijeme primanja prvog

bajta poslužitelja, vrijeme primanja zadnje bajta poslužitelja, broj proslijeđenih i neuspjelih transakcija [7].

Testiranje opterećenja prestaje kada nisu vidljive nove pogreške u radu pri opterećenju sustava [8]. Također, provođenjem testa opterećenja može se podići alarm za velike probleme koji nastaju pri opterećenjem i prije nego što web sustav dođe u ruke klijenata (korisnika), problemi se mogu riješiti [9].

### **3. TESTIRANJE PROGRAMSKE PODRŠKE WEB SUSTAVA**

U ovom poglavlju bit će objašnjeno što je testiranje programske podrške web sustava i od čega se sastoji, tipove i vrste testova koje se provode za testiranje programske podrške web sustava, a bit će objašnjeni i testovi koji će provoditi za testiranje web sustava za prilagodljivi višekriterijski prikaz vijesti iz različitih izvora.

Testiranje je proces pronalaženja greški tj. postupak vrednovanja sustava ili njegove komponente s ciljem utvrđivanja zadovoljavaju li navedene zahtjeve ili ne. Testiranjem se pokreće sustav radi prepoznavanja nedostataka, pogrešaka ili zahtjeva koji nedostaju u odnosu na stvarne zahtjeve. Kako programi ili bilo koji drugi digitalni sustavi nisu kontinuirani, testiranje graničnih vrijednosti nije dostatno da garantira ispravnost. Sve moguće vrijednosti moraju biti testirane i verificirane, ali cjelovito testiranje najčešće nije moguće izvršiti [10].

Testiranje je integralni dio razvoja programske podrške web sustava i ono se primjenjuje u svim fazama razvoja programske podrške web sustava, a na testiranje se potroši nekad i do 50% vremena i nakon toliko testiranja nije sigurno hoće li web sustav ispravno raditi jer nije moguće pronaći sve pogreške rada sustava. Kako se testiranjem smanjuju pogreške web sustava, tako se smanjuju i troškovi, dok se kvaliteta povećava zbog toga postoje razni razlozi testiranja.

Testiranje web sustava mora se učiniti iz više razloga, a neki od njih su: provjera ispravnosti, poboljšanje kvalitete i procjena pouzdanosti. Kako bi se testiranje provelo što bolje i jednostavnije, potrebno je i dizajnirati programsku podršku web sustava što jednostavnije kako bi se kasnije lakše održavala. Zbog toga zadataka svakog programera je što jednostavnije izraditi web sustav i napisati jednostavan kod, kako bi testni inženjer s lakoćom mogao testirati ispravnost, kvalitetu i pouzdanost sustava te se s lakoćom snaći u kodu.

Testiranjem ispravnosti web sustava testira se ispravnost funkcionalnosti programske podrške te se provjerava odgovora li programirani dio zatraženom zahtjevu od strane korisnika. Koliko god testova rabili, konačan broj testova ne može potvrditi da program radi u svim slučajevima [10]. Pouzdanost programske podrške ovisi o više čimbenika, a neki od njih su: uređaj na kojem se program vrti i njegov kapacitet jer neće program isto raditi na dva različita sustava, kompleksnosti programa i korištenju od strane korisnika. Kako bi se procijenila pouzdanost sustava prije samog testiranja definiraju se rezultati koji se očekuju, a zatim nakon testiranja uspoređuju se dobiveni rezultati s očekivanim. Kvaliteta se ne može izravno testirati, ali smanjenjem broja grešaka u sustavu i poboljšanjem sustava i ispunjavanjem svih zahtjeva od strane korisnika poboljšat će se i kvaliteta sustava.

Testiranje se provodi u različitim fazama razvoja programske podrške web sustava. Provođenje testiranja ovisi o više čimbenika koji se mogu grupirati na: čimbenike firmi i njenih načela, čimbenike projekta i njegove veličine te čimbenike metodologije. Testiranje web sustava se može provoditi nakon svake faze razvoja npr. slijedi definiranje zahtjeva pa testiranje tj. pregled zahtjeva koji se mogu ispuniti, a koji ne mogu i pisanje testnog plana u kojem se definiraju testni scenariji i testni slučajevi po kojima će se testirati web sustav. Zatim slijedi razvoj dizajna sustava pa testiranje dizajna sustava, tu se većinom provodi Usability test tj. test iskoristivosti, onda slijedi sam razvoj web sustava pa testiranje nakon svake faze razvoja tj. nakon svakog napravljenog ciklusa.

Nakon završetka razvoja web sustava još jednom se provodi testiranje radi provjere ispravnosti svih funkcionalnosti i radi provjere pouzdanosti (regresijsko testiranje). Ovakav način provođenja testiranja koristi se na velikim projektima razvoja web sustava dok na malim projektima nije potrebno testirati nakon svake faze već samo na kraju kada se projekt cijeli završi.

Različite faze razvoja web sustava uvjetuju različite faze testiranja, a to su:

- faza pripreme za provođenje testiranja web sustava,
- faza definiranja testnog plana, testnih slučajeva i testnih scenarija,
- faza provedbe testiranja web sustava,
- i faza prikupljanja, analiziranja i uspoređivanja rezultata.

Faza pripreme testiranja je najvažnija faza testiranja. O njoj ovisi ishod i uspješnost cijelog proces testiranja web sustava. U toj fazi definiraju se testovi koji će se provoditi na tom projektu razvoja programske podrške, zatim slijedi raspis testnog plana u kojem se definiraju svi testni slučajevi koji će se testirati i koji obuhvaćaju definiranje svih funkcionalnosti aplikacije, a tako se definiraju testni scenariji koji sadrže korake po kojima će odviti testiranje pojedinih funkcionalnosti.

Pošto je testiranje veliki proces i dugo traje pa su iz tih razloga nastale i neke zablude testiranja. Neke od važnijih zabluda su da je testiranje preskupo, oduzima puno vremena, kako se testiraju samo do kraja razvijeni proizvodi, da je potpuno testiranje moguće, kako testirani program nema pogreške i da propušteni nedostaci su nastali zbog testnog inženjera, a oni su odgovorni za kvalitetu proizvoda, također automatiziranje testova treba se koristiti uvijek kako bi se dobilo na vremenu i da bilo tko može testirati program i da je zadaća testnog inženjera pronaći pogreške [11]. Iz ovog se može zaključiti da je testiranje bitan proces za razvoj programske podrške i kako programeri tako i testni inženjeri imaju veliku ulogu u razvoju programske podrške. Također testiranje je proces koji se može izvoditi beskrajno, ali u jedno trenutku treba prestati jer nikad sa

sigurnošću se ne može reći da su otkriveni i uklonjeni svi mogući defekti [11]. Najbolja odluka za prekid testiranja je kad su ispunjeni svi zahtjevi klijenta (korisnika), kad su rezultati na te zahtjeve prolazni, odlukom razvojnog tima kao i klijenta (korisnika).

### **3.1. Tipovi testiranja programske podrške**

U tipove testiranja programske podrške ubrajaju se dva načina testiranja, a to su manualno ili ručno testiranje i automatizirano testiranje. Ručno ili manualno testiranje još se naziva i istraživačko testiranje te se provodi ručno na više načina.

Prvi način je da testni inženjer provodi tehniku testiranja bez ikakvog saznanja o unutarnjem načinu rada aplikacije i usmjereno je samo na ulaz i izlaz, ako su dobiveni očekivani rezultati, test je prošao, ova tehnika zove se „Testiranje pristupom crne kutije“. Drugi način je da testni inženjer zna unutarnji način rada aplikacije pri čemu treba pronaći koji dio koda ne radi kako treba, a takav pristup zove se „Testiranje pristupom bijele kutije“.

Automatiziranje testova provodi se na velikim projektima tj. dugoročnim projektima radi lakšeg provođenja testiranja i radi lakšeg održavanja kvalitete i sigurnosti projekta. Prije automatiziranja potrebno je provjeriti mogućnost automatiziranja jer nije moguće sve dijelove web sustava automatizirati. Također nije potrebno provoditi automatiziranje na malim projektima jer projekt brzo završava i nije potrebno vrijeme potrošiti i uložiti na automatiziranje testova. Za automatiziranje testova potrebno je raspoloživo vrijeme, stabilan program s gledišta ručnog testiranja i definiranje zahtjeva koji se ne mijenjaju često.

#### **3.1.1. Ručno testiranje programske podrške**

Ručno testiranje je tip testiranja programske podrške bez upotrebe posebnih testnih alata ili skripti. Testni inženjer preuzima ulogu krajnjeg korisnika i testira programsku podršku očekujući neobično ponašanje ili pogreške programske podrške. Ovaj tip testiranja je najprimitivniji tip testiranja i pomaže u pronalaženju pogrešaka.

Svaka nova aplikacija tj. programska podrška mora se prvo ručno tj. manualno testirati kako bi se prošlo kroz web sustav i pregledalo što se može automatizirati, a što ne. Ručno testiranje zahtjeva puno više vremena i više napora, ali je potrebno provjeriti izvedivost automatiziranja testova. Također ručno testiranje ne zahtijeva poznavanje testnih alata [12].

Prije samog provođenja ručnog testiranja programske podrške, testni inženjer raspisuje testni plan po razinama testiranja programske podrške. U razine testiranja spadaju: testiranje baze podataka programske podrške, testiranje strane poslužitelja i testiranje strane korisnika programske podrške.

Testni plan sastoji se od testnih scenarija i testnih slučajeva. Testni slučajevi obuhvaćaju funkcionalnosti koje će se testirati i testni inženjer ih sam raspisuje, dok testni scenarij je skup testnih slučajeva i raspisuje se u koracima po redoslijedu. Za raspis testnih planova koriste se testni programi i alati kao što su: Jira, TestRail, StrongQA i itd., a neke manje firme za raspis korite i Microsoft Office Excel. Nakon što je raspisan testni plan, testni inženjer priprema vrste i metode kojima će testirati programsku podršku.

Svrha provođenja ručnog testiranja je osigurati da programska podrška ne sadrži pogreške i utvrđuje se sklad s navedenim funkcionalnim zahtjevima. Postoje mitovi o provođenju ručnog testiranja, a jedan od njih je kako svatko može ručno testirati web sustava, što nije istina jer za provedbu ručnog testiranja potrebne su određene vještine i određeni način razmišljanja. Također, jedan od mitova kaže da ručno testiranje daje programsku podršku bez pogrešaka, što nije istina, nitko ne može pronaći sve, baš sve pogreške. Postoji još jedan mi koji kaže kako je automatiziranje testova bolje i sigurnije od ručnog testiranja što također nije istina jer nije moguće sve automatizirati i ručno testiranje je tu neophodno.

Kako bi se ručno testiranje pravilno provelo, najbitnije je da testni inženjer počne razmišljati na način kako nešto srušiti i pokidati, a ne kako nešto implementirati da dobro radi. Također mora biti sposoban sjetiti se što više testnih slučajeva kako bi obuhvatio testiranje cijele programske podrške web sustava. Testni inženjer ima zadatak razmišljati i djelovati s krajnjim korisnikom na umu te mora biti strpljiv, kreativan i otvorenog uma. U tablici 3.1 prikazane su prednosti i nedostaci ručnog testiranja, a neki od njih su već navedeni u prijašnjem tekstu.

**Tab. 3.1.** *Prednosti i nedostaci ručnog testiranja*

<b>Prednosti</b>	<b>Nedostaci</b>
Cijene alata su niže	Vrijeme izvršavanja je duže
Test inženjer se koncentrira na jedan test i izvršava ga više puta	Vrijeme izvršavanja je duže
Test inženjer posjeduje intelekt i uočava nedostatke koje automatiziranim testiranjem ne bi mogli biti uočeni	Nepreciznost pri testiranju vremenske učinkovitosti programske podrške

### **3.1.2. Automatsko testiranje programske podrške**

Automatsko testiranje programske podrške obuhvaća pisanje testnih skripti i mini programa za automatizirano testiranje programske podrške. Najčešće se koristi za ponovo pokretanje testnog scenarija koji se izvodi ručno, ali brzo i ponavljajuće [13].

Kako je već navedeno u prethodnom potpoglavlju 3.1.1, nije moguće sve automatizirati i zato je prvo bitno proći kroz programsku podršku ručno te pregledati što se može automatizirati, a što ne. U većini slučajeva moguće je automatizirati prijavu korisnika na web sustav, istovremeni pristup velikog broja korisnika na web sustav, neki dijelovi korisničkog sučelja te validacija polja. Automatiziranje se koristi na dugoročnim projektima s velikim funkcijama tj. na projektima koji zahtijevaju često testiranje istih dijelova, a zahtjevi se ne mijenjaju i moguća je raspoloživost vremena. Također automatiziranjem testova smanjuju su troškovi i vrijeme testiranja što olakšava posao testnom inženjeru, a poboljšava i ubrzava rad na programskoj podršci.

Automatiziranje se obavlja u jezicima koji podržavaju automatiziranje i u automatiziranoj programskoj podršci. Niz je alata koji omogućuju pisanje testnih skripti, a neki od njih su Katalon studio, Selenium, Visual Studio, TestComplete, IntelliJ, JMeter i itd.. Testne skripte mogu biti napravljene na dva načina, a to su: pisanje koda od strane testnog inženjera ili snimanje koraka u programu koji služi za automatiziranje testova. Oba načina ovise o pristupu i ideji testnog inženjera i o veličini programske podrške.

Tehnologije koje služe za automatiziranje testova nisu lako dostupne i takve tehnologije uglavnom su namijenjene velikim kompanijama s kompleksnim projektima zbog svoje cijene, a za male projekte ponekad nije isplativo koristiti automatiziranje testova, ali je moguće je pronaći kompromis između ručnog i automatiziranog testiranja [14].

Automatskim testiranjem mogu se izvršavati testovi koje je teško ili nemoguće provoditi ručno. Primjer takvog testa bit će prikazan u ovom diplomskom radu i služi radi provjere rada web sustava koji koristi nekoliko stotinjaka ljudi. Ručnim testiranjem je gotovo nemoguće testirati ovu situaciju, ali preko automatskih testova moguće je dobiti simulaciju željene situacije [15].

Također automatskim testiranjem povećava se sigurnost sustava, ako je dobro definiran skup automatiziranih testova i testovi prođu bez problema, povećava se sigurnost da u sustavu neće biti neugodnih iznenađenja nakon isporuke [16].

Kao i kod ručnog testiranja tako i kod automatskog testiranja postoje određeni problemi, a neki od njih su nerealna očekivanja, npr. očekuje se da će se testiranje moći izvršiti u što kraćem roku, loša praksa testiranja, očekivanje da će se pronaći veliki broj pogrešaka, lažni osjećaj sigurnosti jer test inženjer smatra kako je pokrio sve testove programske podrške. U problemima se mogu pronaći i održavanje napisanih testova kao i tehnički problemi zbog verzija rada tehnologija te organizacijski problemi s kojima se svakodnevno susreću test inženjeri.

Rezultati automatiziranih testova dosta ovise i računalnim resursima, potrebno je dobro računalo i dobar procesor koji može izdržati veliko opterećenje prilikom provedbe testova. U tablici 3.2 prikazane su prednosti i nedostaci automatskog testiranja, dok su tablici 3.3 prikazane razlike ručnog i automatiziranog testiranja.

**Tab. 3.2.** *Prednosti i nedostaci automatskog testiranja*

<b>Prednosti</b>	<b>Nedostaci</b>
Testiranje se izvodi brže	Cijena alata je skupa
Nisu potrebni dodatni ljudski resursi	Automatizirani testovi ne mogu razmišljati kao ljudi
Testovi se mogu izvršavati u bilo koje doba	Ne može se sve automatizirati
Potrebno je samo jedno računalo	Problem održavanja testova
Testovi se mogu snimiti i doraditi	
Rezultati su lako prikazani u jednom dokumentu te su razumljivi svima	

**Tab. 3.3.** *Razlika ručnog i automatskog testiranja [8]*

<b>Ručno testiranje</b>	<b>Automatsko testiranje</b>
Zahtijeva ljudsku intervenciju za provođenje testa	Zahtijeva testne alate za provođenje testa
Ova vrsta ispitivanja zahtijeva vrijeme od obučanih testnih inženjera, a to povlači velike troškove	Štedi vrijeme, troškove i ljudske resurse, samo jednom implementirano pa se onda ponavlja
Svaka vrsta programske podrške može se ručno ispitati, a određene vrste pogodnije su za ručno izvršenje	Preporučuje se samo za stabilne sustave i obično se koristi za regresijsko testiranje
Više posla, više vremena i više ponavljanja	Obraduje dosadni dio provođenja istih slučajeva iznova i iznova

### **3.2. Vrste testova za web sustav**

Testiranje se može podijeliti na dvije glavne vrste testiranja, a to su: funkcionalno testiranje i nefunkcionalno testiranje. Funkcionalno testiranje je proces testiranja sustava prema zahtjevima [17]. Nefunkcionalno testiranje za razliku od funkcionalnog testiranja provjerava karakteristike sustava kao što je brzina rada sustava [18]. U ovom diplomskom radu dogovorena je provedba za obje vrste testova te će se uspoređivati očekivani i dobiveni rezultati, a uz to će se usporediti i rezultati dobiveni u različitim alatima.

Uz funkcionalne i nefunkcionalne vrste testova postoji još puno različitih vrsta testova, u ovom radu jedni će biti spomenuti, a nešto detaljnije obradit će se oni koje će se koristiti za testiranje programske podrške web sustava prikaza vijesti News. Neke od boljih vrsta testova su: test



pristupačnosti, test granične vrijednosti, test kompatibilnosti preglednika, test kompatibilnost, test sukladnosti, istraživački test, test gorile, test sretnog puta, test opterećenja, test performansi, test ispravnosti, test sigurnosti, stres test, test upotrebljivosti i itd. [19]. Navedeni testovi su unaprijed definiraju na testiranju programske podrške, a zatim se po njima raspisuju testni slučajevi u testnom planu.

### **3.2.1. Funkcionalno testiranje web sustava**

Funkcionalno testiranje programske podrške web sustava je jedno od najbitnijih i najkorištenijih testova današnjice. Ono se provodi na svakom projektu i od velike je važnosti. Funkcionalno testiranje je testiranje programske podrške prema zahtjevima korisnika. Takvo testiranje kao rezultat daje rezultate koji se uspoređuju s očekivanim rezultatima, također za ovaj način provedbe testa potrebno je jasno definiranje funkcionalnih zahtjeva kako bi se mogao detaljno raspisati testni plan i detaljno testirati svaku funkcionalnost programske podrške.

Cilj funkcionalnog testiranja je provjerava programske podrške web sustava i je li programska podrška ispunjava sve definirane zahtjeve. Ono se može se provesti na dva načina, a to je kao pozitivno testiranje ili kao negativno testiranje. Pozitivnim testiranjem obuhvaćaju se ispravnosti rada funkcija programske podrške kako je definirano u zahtjevima, dok negativno provođenje funkcionalnog testa koristi se kada je potrebno pronaći pogrešku u radu programske podrške.

U funkcionalno testiranje spadaju:

- Jedinično testiranje - testiranje jedinica programske podrške kako bi se provjerilo da li zasebno ispravno funkcioniraju (eng. *Unit testing*),
- Integracijsko testiranje - testiranje nekoliko jedinica programske podrške kao grupe (eng. *Integration testing*),
- Sustavno testiranje - testiranje cijelog sustava kako bi se potvrdilo zadovoljstvo zahtjeva (eng. *System testing*) [20].

Funkcionalno testiranje obavlja se u sljedećim koracima:

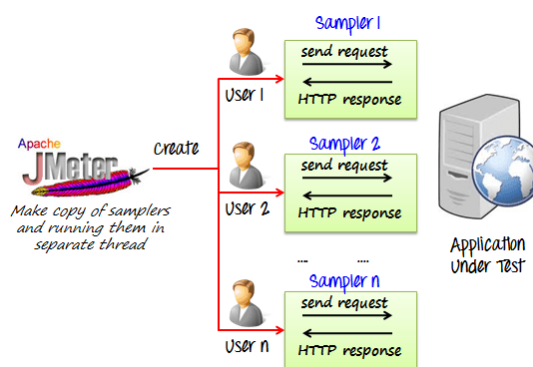
- Identifikacija funkcionalnosti koje će programska podrška sadržavati,
- Napraviti ulazne podatke na temelju specifikacija funkcije,
- Odrediti izlaz na temelju specifikacije funkcije,
- Izvršiti testni slučaj,
- Usporediti stvarne i očekivane rezultate [21].

Funkcionalno testiranje učinkovitije je kada su uvjeti ispitivanja stvoreni izravno iz zahtjeva korisnika tj. iz dokumentacije sustava (Zahtjevi sustava programske podrške). Nedostaci u toj dokumentaciji neće se otkriti testiranjem i to može biti uzrok nezadovoljstva krajnjeg korisnika kada konačno upotrijebi naručenu programsku podršku [21].

### 3.2.2. Nefunkcionalno testiranje -test opterećenja web sustava

Test opterećenja spada u nefunkcionalne testove i s tim testom se utvrđuje hoće li web sustav koji se testira zadovoljiti zahtjeve za opterećenjem. Može se koristiti za analizu ukupnih performansi poslužitelja pod velikim utjecajem. Potpomognuto je alatima za automatsko testiranje, virtualnim korisnicima i skriptama [22]. Uz njega se provodi i stres test kojim se testira ponašanje sustava u neuobičajenim uvjetima kao što su ponovo pokretanje, isključivanje baze podataka, pokretanje zahtjevnih procesa te dovođenje sustav iznad opterećenja. Opteretiti se mogu i API metode i samo korisničko sučelje.

Test opterećenja (eng. *Load testing*) je modeliranje očekivane upotrebe simuliranjem višestrukog pristupa korisnicima web-usluga istodobno, dok Stres Test (eng. *stress testing*) je svaki web poslužitelj ima maksimalni kapacitet učitavanja. Kada se teret prekorači, web poslužitelj započinje polako reagirati i proizvesti pogreške. Svrha testiranja stresa je pronaći maksimalno opterećenje koju web poslužitelj može podnijeti. Na sljedećoj slici 3.1 prikazano je opterećenje sustava pomoću alata JMeter [22].



Sl. 3.1. Prikaz opterećenja sustava pomoću alata JMeter [22]

Postupak provođenja testa opterećenja detaljnije će biti prikazan u nastavku, a sada su samo navedeni koraci po kojima se pridržava test inženjer pri provođenju. Test opterećenja obavlja se u sljedećim koracima:

- Pregled sustava na kojem se provodi test,
- Pronalazak alata u kojem će se provoditi test opterećenja,
- Definiranje testne grupe (npr. Na stranicu će se prijaviti 100 korisnika u isto vrijeme),

- Definiranje stavki koje će se mjeriti ovim testom,
- Definiranje izgleda rezultata,
- Definiranje očekivanih rezultata,
- Usporedba rezultata,
- Prikaz rezultata.

Test opterećenja ne provodi se na svakom projektu, odnosno nije potrebno ga provoditi na sustavima kojima se služi jedan korisnik jer takav sustav neće biti opterećen, ali ovaj test potrebno je provesti na velikim sustavima i sustavima koje će koristiti više korisnika u isto vrijeme.

U provedbi ove vrste testa, potrebno je imati dobar računalni sustav i dobar procesor koji može izdržati provedbu testova jer se većinom provodi na lokalnom poslužitelju dok se web sustav ne postavi na vanjski poslužitelj s više resursa. Opterećivanjem sustava s dosta podataka i virtualnih korisnika i računalno mora moći obraditi sve te podatke i podnijeti tu opterećenost. Najlakše je to objasniti preko primjera pa naprimjer jedno računalno sadrži jako dobar procesor, definira se testna grupu od 500 virtualnih korisnika i pri provedbi testa vidljivo je koliko se procesor i memorija računala troši pri provedbi i koliko može izdržati što znači da ovaj test neće imati jednake rezultate na dva različita računala, mogu biti blizu, ali rijetko da će biti isti.

## 4. WEB SUSTAV ZA PRILAGODLJIVI VIŠEKRITERIJSKI PRIKAZ VIJESTI IZ RAZLIČITIH IZVORA

Web sustav za prilagodljivi višekriterijski prikaz vijesti iz različitih izvora, diplomski rad kolege Filipa Marasa s Fakulteta elektrotehnike, računarstva i informacijskih tehnologija namijenjen je širem krugu korisnika [1]. Kako bi privukao što više korisnika, dizajn trebao bi ti biti što jednostavniji, prilagodljiviji i s jasno definiranim putanjama kako bi se korisnici s lakoćom snalazili na web sustavu. Također osim definiranih putanja bitno je prilagoditi boje kako bi korisnika privuklo na stranicu, kako ne bi naštetilo korisnikovom vidu i kako bi sam prikaz bio uredan.

Osim prikaza i funkcionalnosti također su bitni izvori s kojih se vijesti dohvaćaju jer je bitno što korisnika zanima i koji portali su najviše posjećeni kako bi što više korisnika privuklo te kako bi nastavili koristiti ovaj tip web sustava prikaza vijesti.

### 4.1. Dizajn web sustava prikaza vijesti

Dizajn web sustava prikaza vijesti News je jednostavan i prilagodljiv za sve uzraste. Dizajn se sastoji od početne stranice na kojoj se nalazi naslov web sustava, šest portala jednostavnog prikaza te komponenti potvrde, prikaz svih portala i prikaz vijesti, prikaz na slici 4.1.



Sl. 4.1. Početna stranica web sustava News

Na slici 4.1 također se može vidjeti koje sve portale korisnik može odabrati, a to su: Index.HR, RTL, BBC News, CNN, The New York Times i Sport News. Ako korisnik želi pregledati sve vijesti sa svih portala, nema potrebe da ih odabire ručno jer postoji jednostavno rješenje kao što je odabir komponente „ALL“ koji vodi na drugu stranicu s prikazom svih vijesti sa svih portala kao što je prikazano na slici 4.2. Korisnik može prelistati sve vijesti, a može i pretraživati što mu je potrebno.



**Sl. 4.2.** *Prikaz svih vijesti*

Osim odabira prikaza svih vijesti, postoji mogućnost odabira svih sportskih vijesti koji povlači sve sportske vijesti s portala Index.hr i portala RTL. Ako korisnik želi pročitati obične vijesti i sportske vijesti tada odabire portale s kojih želi vijesti i portal SPORT NEWS te potvrđuje svoj odabir. Takav pristup čini razliku odabira sportskih vijesti putem komponente „Sport“ i odabira portala SPORT NEWS. Na drugoj stranici vidljive su nove dvije komponente što se također može vidjeti na slici 4.2, komponenta „Home“ je komponenta koja korisnik vraća na početnu stranicu ako ju odabere, a komponenta „Search“ je komponenta pomoću koje korisnik ima mogućnost pretraživanja vijesti. Unos bilo koje riječi u komponentu pretraživanja, korisniku se filtriraju vijesti koje sadrže traženu riječ u naslovu. Kako bi korisnik otvorio vijest potrebno je kliknuti na sliku jer ona označava odlazak na portal i vijest, dok klikom na naslov vijesti nema događaja. Povratkom na početnu stranicu korisnik ponovo ima mogućnost odabira portala, prikaza vijesti ili prikaza sporta, ako korisnik želi izaći s web sustava prikaza vijesti News, potrebno je gašenje putem opcije izlaz na internet pregledniku.

## **4.2. Svrha web sustava prikaza vijesti**

Svrha web sustava prikaza vijesti News je prikaz vijesti s više portala, omogućiti korisniku lakše korištenje web sustava prikaza vijesti, pretraživanje vijesti, pregled sportskih vijesti, mogućnost povratka i odabir drugih portala za pregled vijesti.

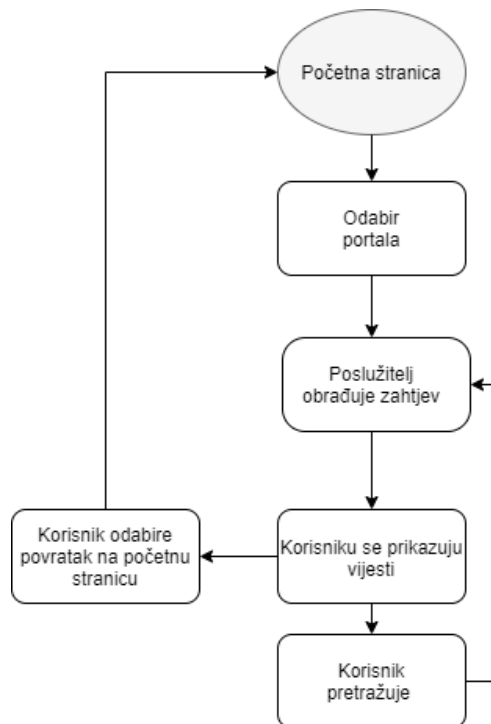
Također svrha ovog web sustava je prikaz vijesti u različitim preglednicima kao što su Chrome, Firefox, Opera, Edge, itd. Kako bi pristup bio omogućen za više korisnika. Web sustav je prilagodljiv svim uzrastima i boje su prilagođene kako bi putanje bile jasno definirane.

## **4.3. Arhitektura web sustava prikaza vijesti**

Arhitektura samog web sustava prikaza vijesti News vidljiva je na slici 4.3. Dolaskom na početnu stranicu, korisnik ima mogućnost odabir portala, kada korisnik odabere portal, poslužitelj obrađuje

zahtjev te kao povratnu informaciju, korisniku vraća prikaz vijesti s odabranog ili odabranih portala.

Nakon prelistavanja i čitanja vijesti, korisnik ima mogućnost pretraživanja ili povratka na početnu stranicu. Kada korisnik upiše riječ koju želi pretražiti, poslužitelj ponovo obrađuje zahtjev i kao povratnu informaciju vraća prikaz vijesti s pretraženom riječi. Ako korisnik želi novi odabir portala ima mogućnost povratka na početnu stranicu i ponovi odabir mogućnosti koje su mu dane.



Sl. 4.3. Dijagram toka web sustava prikaza vijesti News

## 4.4. Rješenje web sustava prikaza vijesti

### 4.4.1. Prednosti web sustava prikaza vijesti

Prednosti ovakvog web sustava prikaza vijesti je pregled vijesti s više portala na jednom mjestu, sortiranje vijesti, pretraživanje vijesti. Također u prednosti spada i mogućnost povratka na početnu stranicu gdje korisnik može ponovo odabrati portale s kojih želi prelistati tj. pročitati vijesti. Osim što nudi mogućnost prikaza vijesti s više portala, u prednost spada i prikaz sa svih vijesti ponuđenih portala te ponuda za pregled sporta. Korisnik s lakoćom može pretraživati vijesti te lako pristupiti onoj koja ga zanima. Ovakav prikaz je vrlo jednostavan i vrlo privlačan korisnima jer se na stranici mogu s lakoćom snalaziti i kretati, pregledno je te korisnik može s lakoćom odabrati što mu treba.

### 4.4.2. Nedostaci web sustava prikaza vijesti

Nedostaci se mogu podijeliti u dvije grupe, a to su: vizualni nedostaci i funkcionalni nedostaci. Funkcionalni nedostaci su nedostaci funkcionalnosti na stranici, na web sustavu prikaza vijesti

News može još dodati portala s kojih bi se mogle vijesti povlačiti. Nakon odabira portala i prikaza vijesti, trebala bi postojati mogućnost filtriranja i sortiranja vijesti, npr. filtriranje podataka po portalu u slušaju da su prikazane sve vijesti ili filtriranje podataka po datum ili nazivu, a što se tiče sortiranja također se može napraviti sortiranje po nazivu ili datum.

Također, korisniku bi bilo lakše pregledavati vijesti ako postoje brojevi stranice, ne bi morao listati vijesti skroz do dna već bi odabirom broja stranice lakoćom došao do vijesti koja ga zanima. Komponenta pretraživanja postoji i njena svrha je pomoći korisniku pronaći pretraženu riječ, ali ako korisnik uđe u vijest i pročita ju, prilikom izlaza iz vijesti očekuje povratak na listu vijesti koju je dobio pretraživanjem riječ te da pretraživana riječ ostane u komponenti pretraživanja.

Osim navedenih funkcionalnih nedostataka, navedeni su i vizualni nedostaci. U vizualne nedostatke pripada vizualni elementi kao što su boje, prikazani elementi, itd. U web sustavu prikaza vijesti News na početnoj stranici vizualno je sve jasno, jasno su definirane putanje, prikazani portali i elementi kojima se olakša ostatak pristupa i prikaza.

Na drugoj stranici, tj. stranici na kojoj se nalazi lista vijesti sadrži naslov News i komponentu Home, klikom na naslov News korisnik očekuje povratak na početnu stranicu, međutim u ovom slučaju ne događa se ništa. Komponentu Home potrebno je premjestiti na neko drugo i vidljivije mjesto gdje bi ju korisnik odmah uočio. Također boje bi trebalo uskladiti i prilagoditi svim korisnicima kako bi mogli razlikovati komponente. Funkcionalnim testiranjem i testom opterećenja postoji mogućnost prikaza još nekih nedostataka kao i prednosti web sustava prikaza vijesti News.

## **5. KORIŠTENI ALATI ZA TESTIRANJE WEB SUSTAVA PRIKAZA VIJESTI**

U ovom poglavlju obrađeni su alati koji će se koristiti prilikom provođenja testa opterećenja. Pošto test opterećenja je automatizirani tip testiranja potrebno je koristiti i alate koji služe za automatizaciju provedbe testa opterećenja. U ovom radu bit će prikazan rad u dva alata, a to su: JMeter i Visual Studio. Usporedit će se alati i donijeti zaključak koji alat je bolji za provedbu ove vrste testa.

Za provođenje funkcionalnog testiranja koristit će se online program „TestMonitor“ za raspis testnog plana koji je prikazan u nastavku, ali nije potreban testni alat za provedbu funkcionalnog testiranja jer u ovom diplomskom radu ono će se provoditi ručno.

Alat JMeter bit će prikazan u radu na dva načina, a to su: rad s Proxy poslužiteljem i rad s alatom BlazeMeter gdje će se također usporediti rezultati i vidjeti koji je alat bolji i lakši za korištenje, koje su razlike ovih sustava i kako se provodi test opterećenja pomoću njih.

Visual Studio bit će prikazan samo na jedan način, prikazat će se postupak provođenja testa opterećenja i kad se koristi, što se s njim dobije te će se usporediti s alatom JMeter.

### **5.1. JMeter**

Apache JMeter je aplikacija tj. programska podrška otvorenog tipa. Aplikacija je dizajnirana za provedbu testa opterećenja i mjerenje performansi testiranog sustava. Prvobitno je napravljena za testiranje web sustava, ali onda se proširio i na ostale testne funkcije [23].

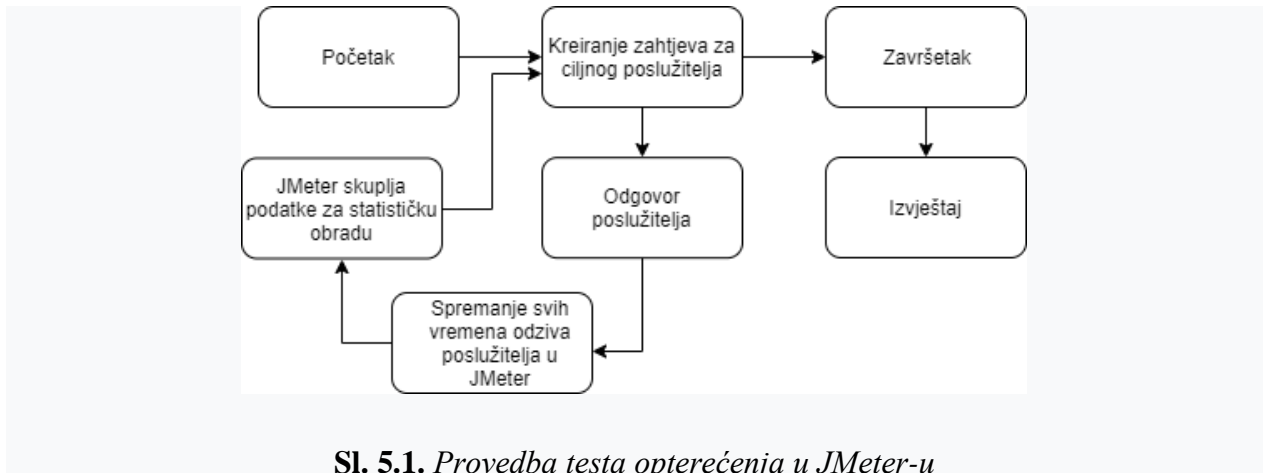
JMeter koristi se za testiranje performansi web dinamičkih aplikacija, statičkih i dinamičkih resursa te se koristi za simulaciju velikog opterećenja na poslužitelju, grupi poslužitelja, mreži ili objektu radi testiranja njegove snage ili za analizu performansi pod različitim opterećenjima.

Alat JMeter je besplatan dostupan je svima te omogućava korištenje izvornog koda za razvoj sustava. Ima jednostavno korisničko sučelje koje se lako koristi i nije potrebno puno vremena za upoznavanjem s ovim alatom. Neovisna je aplikacija i 100% čista desktop aplikacija [24]. Može se izvoditi na više platformi i omogućava istovremeni rad različitih funkcija odvojenih testnih grupa.

Rezultati u ovoj aplikaciji mogu se prikazati u različitim formatima, a neki od njih su stablo, grafikon, tablica i datoteka zapisa. Također mogu se pisati vlastiti testovi i sadrži osnovne protokole kao što su HTTP, JDBC, LDAP, SOAP, JMS i FTP. U aplikaciji JMeter također postoji mogućnost snimanja/reprodukcije testova.



Na slici 5.1 prikazan je dijagram kako JMeter radi. Prije svega simulira testne grupe koji šalju zahtjeve poslužitelju, a zatim vraća statistiku koje prikazuju performanse i funkcionalnosti poslužitelja na kojem se testiranje izvodi, preko grafikona, tablica, stabla, itd. što nudi sama aplikacija JMeter.



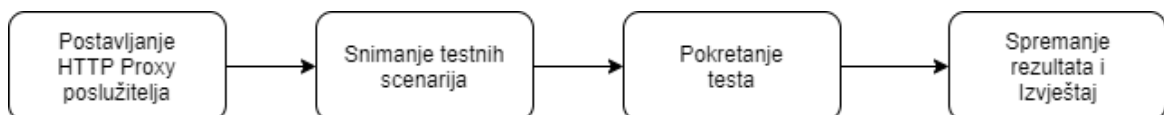
**Sl. 5.1.** *Provedba testa opterećenja u JMeter-u*

Putem aplikacije JMeter snimaju se testovi koji će se testirati, a prilikom snimanja dohvaćaju se API metode. API metode mogu se dohvaćati putem Proxy poslužitelja ili putem ekstenzije na pregledniku Chrome pod nazivom „BlazeMeter“ pa u nastavku su objašnjena oba alata te će se oba koristiti prilikom provedba testa opterećenja u JMeter-u.

### 5.1.1. Proxy poslužitelj

Proxy poslužitelj, odnosno posrednički poslužitelj je računalo koje stoji između klijenta i glavnog poslužitelja kao posrednik [25]. Koristi se za posluživanje web sustava. Omogućuje aplikaciji JMeter gledanje i snimanje korisničkih aktivnosti dok pregledavaju web sustav s uobičajenim preglednikom te dohvaća API metode i sprema [26].

Na slici 5.2 prikazan je postupak korištenja Proxy poslužitelja u aplikaciji JMeter pomoću kojeg se dohvaćaju API metode prilikom snimanja testa. Snimanjem skripti snimaju se koraci koje će virtualni korisnici proći i na temelju toga dobiti rezultate performansi sustava. Detaljniji prikaz provođenja testa opterećenja pomoću Proxy poslužitelja opisan je u poglavlju Provođenje testa opterećenja web sustava.



**Sl. 5.2.** *Koraci snimanja testa pomoću Proxy poslužitelja*

### **5.1.2. BlazeMeter**

BlazeMeter je komercijalna platforma za testiranje osnovana 2011. godine koja je kompatibilna s aplikacijom JMeter. Pruža alate pomoću Amazon Elastic Compute Cloud usluge kao što su testiranje mobilnih uređaja, web aplikacija, web stranica, web usluga ili baza podataka pomoću simulacije tisuće korisnika koji istovremeno posjećuju web sustave [27].

BlazeMeter je web aplikacija otvorenog tipa pa je dostupan svim i lagana je za koristiti i svatko može uživati u njenim uslugama te naveliko olakšava posao test inženjerima. Također BlazeMeter je napredni SaaS alat koji povećava sposobnost JMeter-a tako što može simulirati do 10 milijuna korisnika, daje naprednu analizu i suradnju testova, a ima funkcionalne i API mogućnosti testiranja.

Preglednik Chrome sadrži ekstenziju BlazeMeter koja bilježi sve HTTP/S zahtjeve i stvarne interakcije korisnika koje preglednik šalje, stvara skripte JMeter ili Selenium i automatski ih šalje u BlazeMeter aplikaciju gdje se skripta može izvršiti samo jednim klikom. Ekstenzija tj. proširenje također može omogućiti uklanjanje pogrešaka i testiranje snimljene sesije u Seleniumu na lokalnom pregledniku Chrome, a pomoću JMeter-a lokalno se mogu ukloniti pogreške i poboljšati JMX datoteka kreiranu Chromeovom ekstenzijom kako bi osigurali da je skripta spremna [28]. U ovom radu bit će prikazan rad s ekstenzijom te će se dobivena snimljena skripta pomoću BlazeMeter ekstenzije uređivati u aplikaciji JMeter.

### **5.2. Visual Studio**

Visual Studio je programerski alat kojim se služe programeri i test inženjeri za razvoj aplikacija, web sustava, sustava, igrica, itd. U ovom diplomskom radu programerski alat Visual Studio koristit će se za provođenje testa opterećenja web sustava prikaza vijesti. Ova vrsta testa ne može se provoditi u bilo kojoj verziji alata Visual Studio već je potrebna točno određena verzija, a to je Visual Studio Enterprise 2015.

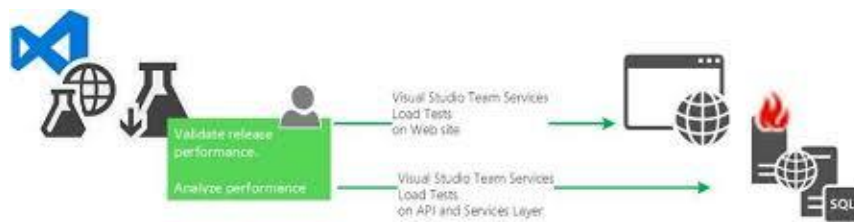
U Visual Studio Enterprise potrebno je dodati komponentu web performance and load testing tools i primijeniti ju. Napokon primjene komponente može se kreirati novi projekt u C# te kreirati test opterećenja. Za snimanje skripti tj. koraka kroz koje će virtualni korisnici prolaziti koristi se Internet Explorer preglednik. Za razliku od JMeter-a koji ima mogućnost korištenja različitih protokola, Visual Studio koristi samo HTTP protokol.

Visual Studio nudi mogućnost spajanja uz plaćanje s Azure DevOps platformom pomoću koje se može koristiti puno više virtualni korisnika, testovi se puno brže vrte jer se vrte na vanjskim

poslužiteljem, a ne na lokalnom računalu i zbog toga ima više mogućnosti za iskoristiti a manje za opteretiti računalo i daje realniju sliku.

Microsoft je 10. prosinca 2018. godine objavio svoje bilješke o izdanju Visual Studio 2019. u kojima navodi uklanjanje značajki testa opterećenja u novoj verziji. Kao alternativu navodi Apache JMeter i BlazeMeter. Također Microsoft navodi kako su alati otvorenog tipa dobri i za programere i test inženjere jer su dostupni svima i imaju jake resurse za iskoristiti te nemaju ograničenja [29].

Postupak testiranja opterećenja u alatu Visual studio ukratko je prikazan na slici 5.3 te su prikazana oba načina provođenja tj. provođenje na lokalno računalu i na Azure DevOps platformi [30].



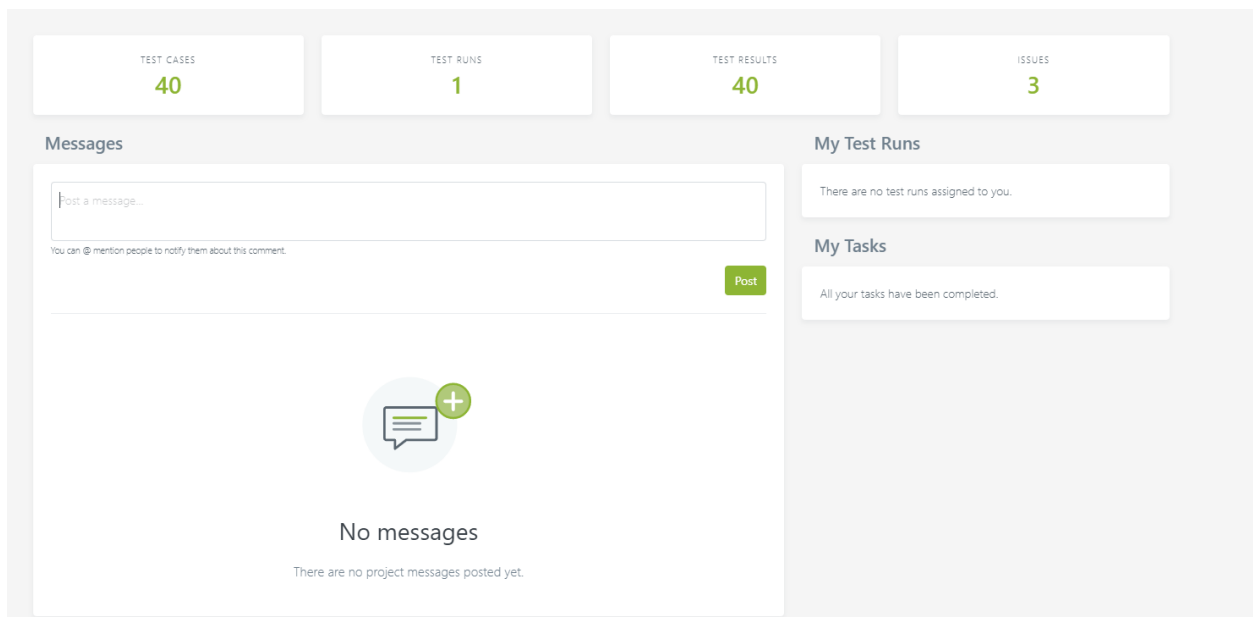
**Sl. 5.3.** Prikaz provođenja testa opterećenja s alatom Visual Studio [26]

## 6. PROVOĐENJE FUNKCIONALNOG TESTIRANJA

U ovom poglavlju prikazano je provođenje funkcionalnog testiranja koje je bitno za web sustave. Prije samog početka provođenja funkcionalnog testiranja potrebno je definirati testni plan, testne scenarije i testne slučajeve. Uz definiranje testnog plana također je poželjno definirati očekivane rezultate, ali nije obavezno.

U ovo diplomsku radu za provođenje funkcionalnog testiranja koristit će se online alat „TestMonitor“ koji sadrži sve, od samog definiranja, dizajniranja i izrade plana testiranja, provođenje testiranja i praćenje rezultata testove do osiguranja kvalitete web sustava [31].

Na slici 6.1 prikazana je početna stranica aplikacija TestMonitor ako je definiran projekt unutar aplikacije. Na početnoj stranici nalazi se broj raspisanih testnih slučajeva, broj koji označava koliko puta se pokrenulo testiranje, rezultate i broj pogrešaka. Također mogu biti prikazani zadaci dodijeljeni članu tima koji radi na razvoju projekta te broj testova za pokrenuti i sandučić za poruke kako bi tim mogao lakše komunicirati dok radi na razvoju.



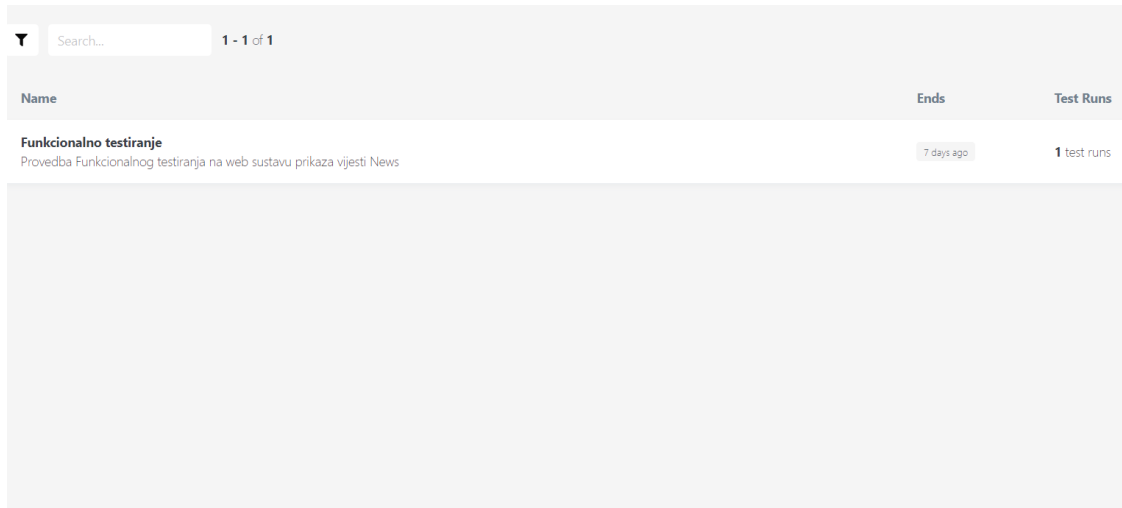
Sl. 6.1. Prikaz prve stranice unutar aplikacije TestMonitor

### 6.1. Priprema za funkcionalno testiranje

#### 6.1.1. Definiranje testnog plana

Zadatak testnog plana je definirati aktivnosti prilikom testiranja, rizike i kontigencije [32]. Testni plan je bitan dokument svakog testnog inženjera i sadrži plan po kojem se testiranje provodi te

pomože oko vođenja informacija i prijave pogrešaka u sustavu. Dokument je dinamičan što znači da je u svakom trenutku dostupan za uređivanje, a pod uređivanje misli se na nadopunjavanje, brisanje ili izmjenjivanje trenutnih testnih slučajeva ili testnih scenarija. Na slici 6.2 prikazan je testni plan Funkcionalno testiranje u aplikaciji TestMonitor. Također na slici prikazano je koliko puta se vrtio testni plan i kada je završilo testiranje.



The screenshot shows a web interface for TestMonitor. At the top, there is a search bar with the text 'Search...' and a dropdown menu showing '1 - 1 of 1'. Below this is a table with the following structure:

Name	Ends	Test Runs
<b>Funkcionalno testiranje</b> Provedba Funkcionalnog testiranja na web sustavu prikaza vijesti News	7 days ago	1 test runs

The table content is displayed in a light gray box, and the rest of the interface below it is a large empty gray area.

**Sl. 6.2.** *Prikaz testnog plana Funkcionalno testiranje*

### **6.1.2. Definiranje testnih scenarija i testnih slučajeva**

Testni scenarij je skup testnih slučajeva, vrsta scenarija koji određuju pozitivne i negativne aspekte projekta kako bi se ocijenili mogući ishodi i kako bi se identificirali potencijalni nedostaci u programu. Manje su opisni kako bi dali više vrijednosti na testne slučajeve koji čine detaljniji dokument i sastoje se od skupa varijabli i uvjeta za utvrđivanje je li program ili aplikacija u skladu s poslovnim zahtjevima i radi kako je izvorno ustanovljeno [33].

Na slici 6.3 prikazani su testni scenariji koji će se provoditi u testiranju web sustava prikaza vijesti, tablica testnih scenarija sadrži naziv testnog scenarija i broja testnih slučajeva koji sadrži testni scenarij npr. testni scenarij Prikaz vise vijesti zajedno sadrži šesnaest testnih slučajeva.

Name	Test Cases
<b>Pocetna Stranica</b> Testiranje pocetne stranice web sustava prikaza vijesti pod nazivom News	8 test cases
<b>Prikaz jedne stranice vijesti</b> 1. Otvaranje web sustava News 2. Odabir web sustav Index.hr 3. Pregled vijesti 4. Odabir web sus...	6 test cases
<b>Prikaz vise web sustava prikaza vijesti zajedno</b> 1. Otvaranje web sustava prikaza vijesti News 2. Odabir vise mogucnosti web sustava 3. Potvrda (S...	16 test cases
<b>Oznacavanje svih vijesti pomocu buttona All</b> 1. Otvaranje web sustava prikaza vijesti News 2. Pomocu buttona All odabrate sve vijesti 3. Pregl...	3 test cases
<b>Oznavanje prikaza vijesti samo za sport (button Sport)</b> 1. Otvaranje web sustava prikaza vijesti News 2. Odabir buttona Sport 3. Pregled svih sportskih v...	3 test cases
<b>Preglednici</b> Pregled stranice u: 1. Firefoxu 2. Chrome 3. Opera 4. Edge	4 test cases

**Sl. 6.3.** Prikaz testnih scenarija

Na slici 6.4 prikazana je tablica koja sadrži raspisane testne slučajeve za testni scenarij „Prikaz više vijesti zajedno“. Svaki testni slučaj ima svoj broj pod kojim se identificira, naziv te mogućnost za postavljanje zahtjeva i uređivanja.

Code	Name	Requirements	Draft
TC12	Odabir BBC News i CNN		...
TC13	Odabir BBC News i Index.hr		...
TC14	Odabir BBC News i RTL		...
TC15	Odabir BBC News i The New York Times		...
TC16	Odabir BBC News i Sport News		...
TC17	Odabir Index.hr i RTL		...
TC18	Odabir Index.hr i CNN		...

**Sl. 6.4.** Prikaz testnih slučajeva scenarija Prikaz više vijesti zajedno

Nakon što su raspisani testni scenariji i testni slučajevi, također je potrebno raspisati i testne korake za svaki testni slučaj kako bi se svi sudionici mogli koristiti dokumentom testni plan, a ne samo testni inženjer koji je izradio testni plan za web sustav. Testni koraci sastoji se od detalja tj. detaljnih koraka što se točno testira i kako se provodi te od ulaznih i izlaznih podataka.

Primjer jednih od testnih koraka prikazan je na slici 6.5. Ovim postupkom svi članovi projekta znat će što se koristilo tijekom testiranja i što trebaju unositi u polja ako se tamo nalazi pogreška.

**Instructions**

1. Otvaranje stranice News
2. Odabir portala BBC News i CNN
3. Pregled vijesti
4. Povratak na pocetnu

**Sl. 6.5. Prikaz testnih koraka slučaja Odabir BBC News i CNN****6.1.3. Definiranje očekivanih rezultata**

Definiranje očekivanih rezultata može se provoditi na dva načina, a to su odmah pri definiranju testnih slučajeva ili na kraju kreiranje tablice i definiranjem očekivanih rezultata. Aplikacija TestMonitor nudi definiranje testnih slučajeva tijekom raspisivanja testnih slučajeva pa nije bilo potrebe za kreiranjem tablice. Na slici 6.6 nalazi se detaljan raspis očekivanih rezultata za testni slučaj „komponenta Submit“. Definiranje očekivanih rezultata definira točno što se treba dobiti da komponenta radi ili kako da se stranica ponaša u određenom trenutku. Nakon definiranja očekivanih rezultata slijedi provedba funkcionalnog testiranja te se tijekom provedbe rezultati uspoređuju i lako se dolazi do zaključka što je pogreška i može se lako prijaviti razvojnom timu kako bi što prije ispravili pogrešku.

**Instructions**

1. Otvaranje web sustava prikaza vijesti News
2. Bez odabira vijesti, odmah kliknuti na button Submit
3. Odabir jedne vijesti pa button Submit
4. Odabir više vijesti pa button Submit
5. Odabir svih vijesti pa button Submit

**Preconditions**

—

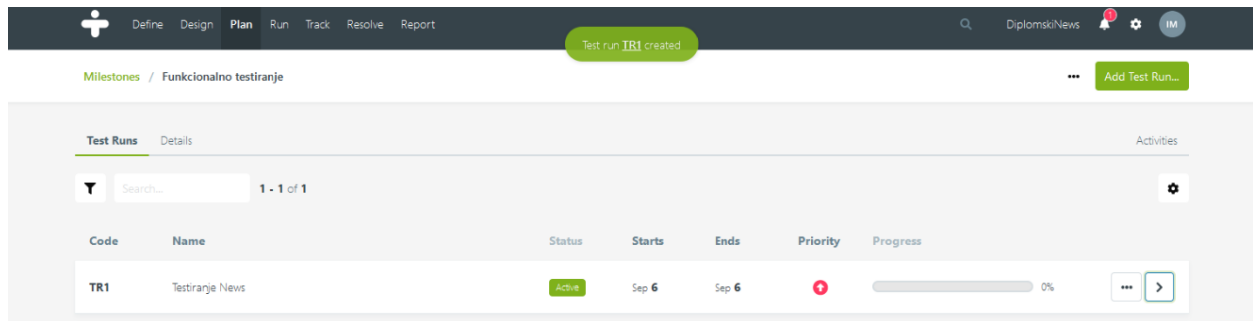
**Expected result**

1. rezultat: Nista se ne dogadja, stranica ostaje na pocetnoj stranici
2. rezultat: Nakon odabira jedne ili vise portala vijesti i klikom na Submit, vodi nas na drugu stranicu gdje su prikazane vijesti

**Sl. 6.6. Prikaz očekivanih rezultata slučaja „komponenta Submit“**

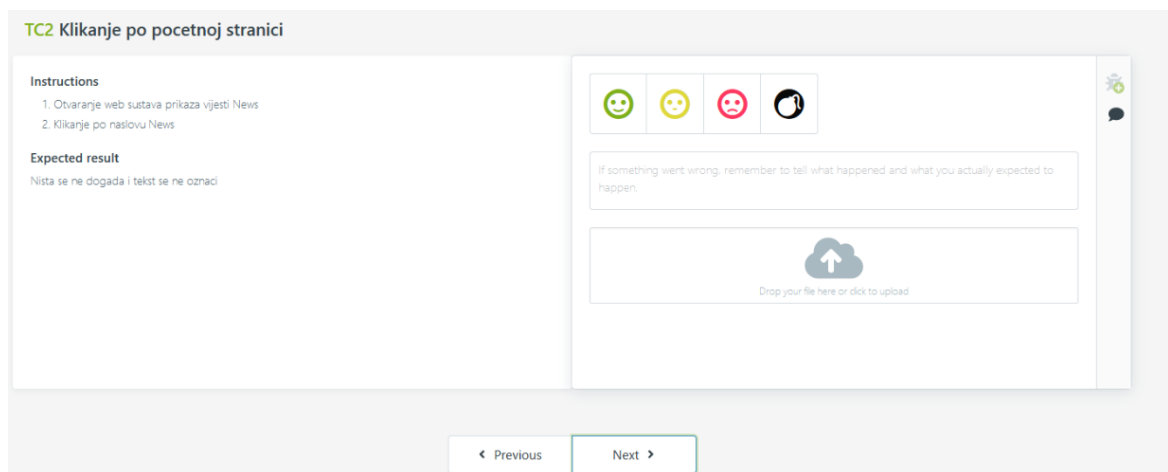
## 6.2. Provedba funkcionalnog testiranja

Nakon svih definiranja i pripremanja za provedbu web testiranja te završetka izrade web sustava prikaza vijesti od strane razvojnog tima, započinje provedba funkcionalnog testiranja kao što je prikazano na slici 6.7.



Sl. 6.7. Prikaz početka provedbe funkcionalnog testiranja

Slika 6.8 prikazuje provođenje funkcionalnog testiranja, tako što se sastoji od testnih koraka koje treba provesti na sustavu, od raspisa očekivanih rezultata kako bi se usporedilo što se želi dobiti, a što ne. Aplikacija TestMonitor nudi ocjenjivanje testnih slučajeva sa smajlicima, zeleni i sretni smajlic označava prolaz, žuti malo manje sretni predstavlja da nešto nije dobro, ali nije potrebno odmah ispravljati, dok crveni označava pogrešku koju je potrebno ispraviti, a uz njega stoji bomba koja predstavlja hitni popravak na web sustavu.

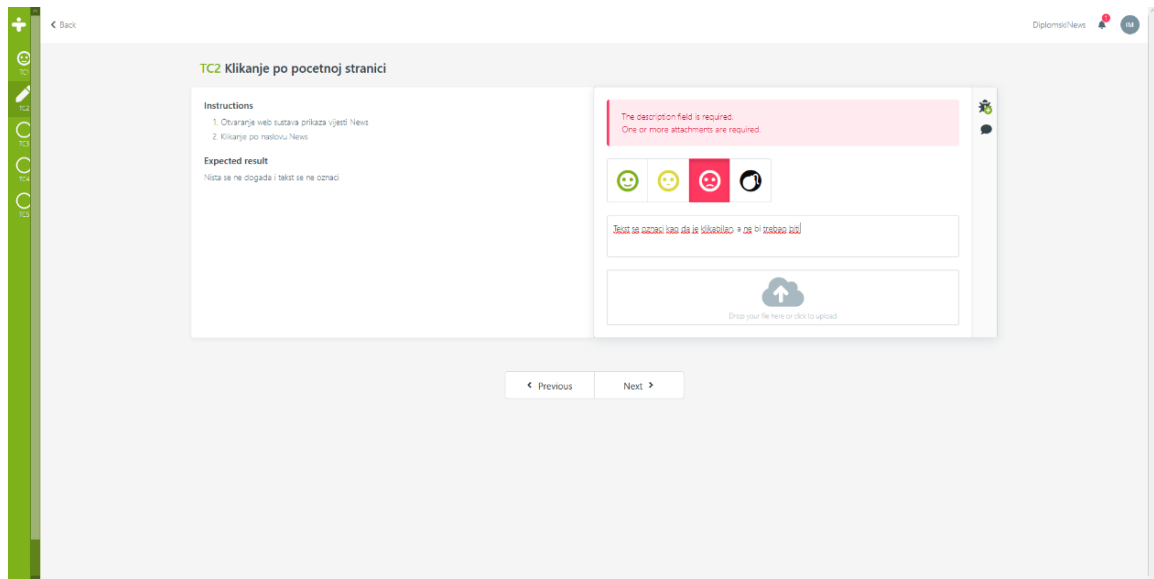


Tab. 6.8. Prikaz provedbe funkcionalnog testiranja

U slučaju da postoji pogreška na testnom slučaju potrebno je opisati što ne radi te koji su koraci rađeni i koji podaci su se koristili kako bi ponovo nastala pogreška, a uz opis koraka potrebno je priložiti i sliku na kojoj se nalazi pogreška kako bi razvojni tim što bolje razumio što se događa. Takvim načinom pristupa poboljšava se komunikacija tima, a time i projekt.



Na slici 6.9 prikazan je postupak prijave pogreške na testni slučaj „Klikanje po početnoj stranici“. Također na slici 6.10 prikazan je detaljnija prijava pogreške u sustav.



Sl. 6.9. Prijava pogreške

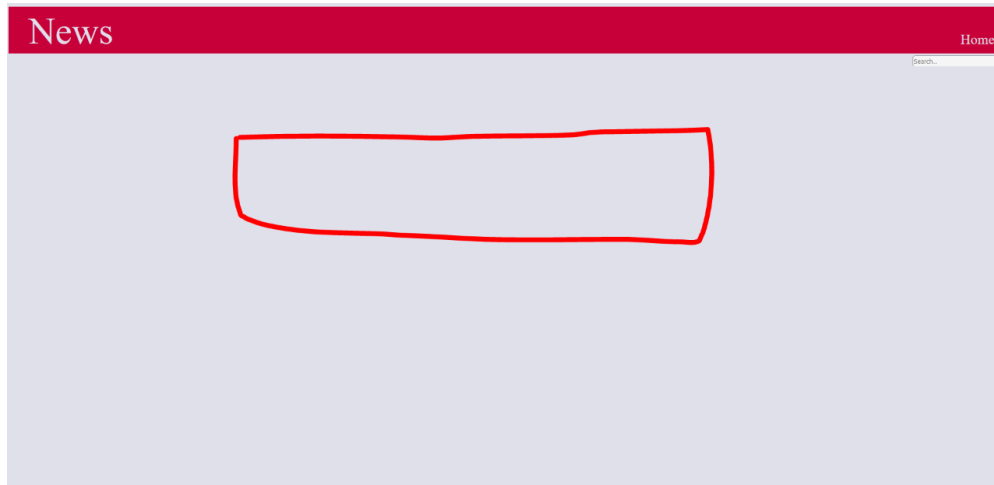
The screenshot shows a 'Add Issue' form. The fields are: 'Category \*' (Bug), 'Name \*' (Naslov News), 'Description \*' (Klikanjem na naslov News, tekst se oznaci kao da je klikabilan, a ne bi trebao biti), 'Status \*' (Open), 'Impact \*' (Low), 'Priority \*' (Medium), and 'Assignee' (Search an assignee...). At the bottom right, there are 'Cancel' and 'Save' buttons.

Sl. 6.10. Detaljni prikaz prijave pogreške

Ovakav pristup zahtjeva ispunu svih polja te detaljni opis kako bi razvojni tim znao što raditi kako bi reproducirali pogrešku te kako bi ju onda znali ispraviti. U polje kategoriju postavlja se pogreška jer je ona naš problem, dodaje se naziv po kojem će se prepoznati prijava, detaljni opis, status

trenutne prijave problema, utjecaj i prioritet pogreške te se može dodati osoba koja je radila na tom dijelu razvoja kako bi vidjela problem.

Funkcionalno testiranje web sustava prikaza vijesti detaljno je proveden i od raspisanih 40 testnih slučajeva, tri su testna slučaja pala, na sljedećim slikama 6.11, 6.12 i 6.13 vidljive su pogreške koje su se događale na web sustavu prikaza vijesti. Osim funkcionalnog testiranja u pregledniku Firefox, svi testni scenarij proveli su se i u preglednicima Chrome, Opera i Edge. Osim navedene tri pogreške ni u drugim preglednicima nije bilo pogrešaka.



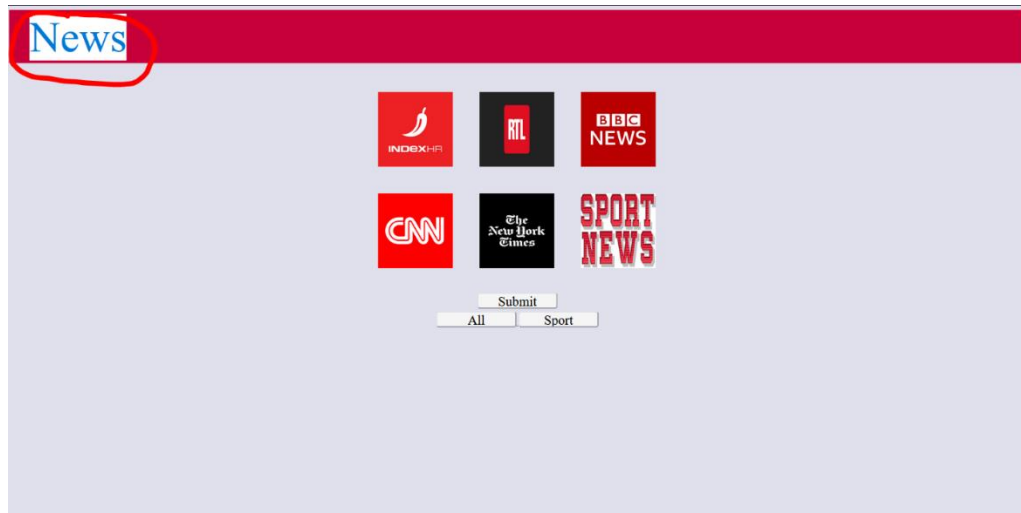
**Sl. 6.11.** *Nedostaje tekst ? – dodati 2-3 riječi objašnjenja*

Kao što se može vidjeti na slici 6.11 nedostaje tekst, ova pogreška nije velika i njen prioritet je srednji, ona nastaje ako se u komponentu pretraživanje unese riječ koja se ne nalazi u nijednim vijestima odabranih portala. Čini pogrešku jer očekivani rezultat je tekst „Nema rezultata“. U slučaju da teksta nema, korisnik ne zna što očekivati ili što napraviti u tom trenutku.



**Sl. 6.12.** *Nedostaje pretraživana riječ*

Na slici 6.12 zaokružena je komponenta pretraživanja, pogreška je nastala pretraživanjem određene riječi, zatim pregledom vijesti i čitanjem vijesti, nakon izlaska iz vijesti u komponenti se više nije nalazila riječ te je sadržaj vijesti bio sa svih portala. Očekivani rezultat je povratak na stranicu s pretraženom riječi kako korisnik ne bih ponovo morao unositi riječ i ponovo pretraživati sadržaj.



**Sl. 6.13.** Označen tekst

Označeni tekst na slici 6.13 nije klikabilan i ne vodi se na drugu putanju pa zbog toga kada se klikne na njega ne treba se ništa događati, također ni ova pogreška nije visokog prioriteta, ali ovim označavanjem očekivani rezultat je drugačiji od onog što ta komponenta predstavlja. Nakon pregled i provjere svih definiranih testnih slučajeva i prijave svih pronađenih pogrešaka „završava se“ provedbu funkcionalno testiranja. Riječ završava označena je navodnicima jer testiranje u pravilu nikada nije moguće završiti samo je potrebno znati stati.

U tablici 6.1 nalazi se sumirani prikaz prijavljenih pogreška tijekom provedbe funkcionalnog testiranja na web sustavu prikaza vijesti News.

**Tab. 6.1.** Prijavljene pogreške

Redni broj	Naziv	Prioritet	Blokirajuća greška
1.	Nedostaje tekst za nula rezultata	Srednji	Ne
2.	Nedostaje pretraživana riječ	Srednji	Ne
3.	Označen tekst	Niski	Ne

## 7. PROVOĐENJE TESTA OPTEREĆENJA WEB SUSTAVA

### 7.1. Priprema za provođenje testa opterećenja web sustava

Priprema za provođenja testa opterećenja nije jednaka pripremi provođenja funkcionalnog testiranja, ali isto zahtjeva dosta posla kao što je definiranje testnih scenarija, definiranje testnih skupina i definiranje očekivanih rezultata. Također potrebno je odrediti koji alati će koristiti, što će mjeriti testom opterećenja te kako će izvještaj testiranja izgledati, a uz to i kako će se provedba odvijati.

Za provođenje testa opterećenja također je bitno vrijeme kada će se test odvijati. Najbolje je u vrijeme dok se korisnici služe web sustavom jer tad se dobiju realni rezultati, dok vrtnjom u noćne sate dok nema korisnika ili ih je jako malo na web sustavu dobivaju se dobri rezultati, ali oni nisu realni.

Prije nego što se krene s priprema, potrebno je instalirati programe koji će se koristiti. U ovom diplomskom radu koristit će se testni alat JMeter 5.2, a uz JMeter potrebno je instalirati i Javu, a poželjno je koristiti posljednju verziju ili verzije Java+8. Nakon instalacije alata JMeter, također je potrebno instalirati alat Visual Studio Enterprise 2017. Nakon provedbe test opterećenja i dobivenih rezultata, potrebno je usporediti alate i dobivene rezultate.

#### 7.1.1. Definiranje testnih scenarija

Za web sustav prikaz vijesti News raspisani su sljedeći testni scenariji koji se mogu vidjeti u tablici 7.1. Za svaki testni scenarij raspisani su testni koraci kako bi pomogli pri snimanju scenarija za provođenje testa opterećenja.

U tablici se nalazi pet testnih scenarija, dva testna scenarija sastoje se od četiri koraka dok se tri testna scenarija sastoje od pet koraka. Ključno je naglasiti da se svi testni scenarij odvijaju na web sustavu News i da se testira opterećenje web sustava prikaza vijesti News, dok u jednom scenariju nalazi se i portal Indeks.hr koji će biti opterećen zbog ulaska u vijesti i čitanje vijesti.

Testni scenariji odabrani su po važnosti i ponašanju korisnika. Ono što bi korisnik mogao vidjeti i proći na stranici raspisano je u testnim koracima.

**Tab. 7.1.** *Odabrani testni scenariji za provođenje testa opterećenja*

Testni slučaj	Otvaranje jednog portala	Otvaranje više portala i pretraživanje	Odabir opcije prikaza svih portala (ALL)	Odabir tri vijesti pa opciju svih portala (ALL)	Odabir svih, pregled, odabir dva portala
Testni koraci					

1.	Otvaranje stranice	Otvaranje stranice	Otvaranje stranice	Otvaranje stranice	Otvaranje stranice
2.	Odabir jednog portala	Odabir tri portala	Odabir opcije svih (ALL)	Odabir tri portala pa odabir opcije svih (ALL)	Odabir svih portala ručno
3.	Pregled vijesti na portalu	Pretraživanje riječi „sport“	Pregled dobivenih vijesti	Pretraživanje vijesti „test“	Potvrda i pregled vijesti
4.	Povratak na početnu stranicu	Pregled vijesti	Povratak na početnu stranicu	Pregled vijesti	Povratak na početnu stranicu i odabir dva portala
5.	/	Povratak na početnu stranicu	/	Povratak na početnu stranicu	Pregled vijesti i ulazak u jednu vijest te povratak na početnu stranicu

### 7.1.2. Definiranje testne grupe

Nakon što su definirani testni scenariji potrebno je definirati testne grupe. Testne grupe čini broj virtualnih korisnika koji će se prijaviti na stranicu i odraditi testne korake u određeno vrijeme. Više čimbenika utječe na određivanje testnih skupina a neki od njih su: CPU računala, memorija računala, alati i definirani ograničeni broj virtualnih korisnika u korištenim alatima.

Pošto se testovi provode na lokalnom poslužitelju potrebno je paziti koliko virtualnih korisnika može podnijeti računalno pa zbog toga kako ne bi previše opteretili računalo na kojem se testovi provode, izabrane su 4 testne grupe. U sljedećoj tablici 7.2 prikazane su vrijednosti koje će se koristiti za provođenje testa u JMeter-u dok se vrijednosti u Visual Studiju nešto razlikuju.

**Tab. 7.2.** Testne grupe

<b>Virtualni korisnici</b>	1	10	80	400
<b>Period pokretanja svih niti (sec)</b>	0	0	4	10
<b>Petlja</b>	1	1	1	1
<b>Trajanje (sec)</b>	60	80	210	360

Zbog ograničenog broja virtualnih korisnika u Visual Studiju koji iznosi 250, korištene su sljedeće vrijednosti:

- Početni broj korisnika: 0 korisnika,
- Trajanje koraka: 5 sekundi,
- Povećanje broja korisnika: 10 korisnika/koraku,
- Maksimalni broj korisnika: 200.

### 7.1.3. Definiranje očekivanih rezultata

Definirani očekivani rezultati nalaze se u tablici 7.3. Rezultati predstavljaju očekivano vrijeme koje je korisniku potrebno da odradi testni scenarij. Neka vremena mjerena su štopericom dok su se koraci prolazili, a neka vremena pronađena su na internetu za očekivane rezultate ponašanja stranice. Npr. definirano je kako korisnik za otvaranje ili učitavanje stranice ne bi trebao čekati duže od pet sekundi.

**Tab. 7.3.** *Definirani očekivani rezultati*

Broj korisnika	1	10	80	400
Testni scenariji				
<b>Odabir jednog portala</b>	5 sec	8 sec	13 sec	20 sec
<b>Otvaranje više portala i pretraživanje</b>	15 sec	18 sec	25 sec	30 sec
<b>Odabir opcije prikaza svih portala (ALL)</b>	8 sec	10sec	15 sec	25 sec
<b>Odabir tri vijesti pa opciju svih portala (ALL)</b>	8 sec	10 sec	15 sec	25 sec
<b>Odabir tri vijesti pa opciju svih portala (ALL)</b>	8 sec	10 sec	15 sec	25 sec
<b>UKUPNO</b>	42 sec	56 sec	83 sec	125 sec

Rezultati se na kraju provedbe obrađuju i detaljno uspoređuju te se zaključuje ponašanje stranice u odnosu kad ju koristi 1 korisnik, 10 korisnika, 80 ili 400 korisnika. Za vrijeme provedbe testa ne bi trebala biti velika razlika između 1 i 10 korisnika, dok na 80 ili 400 se već očekuju zastoji i problemi. Nakon definiranih očekivanih rezultata, definiranih testnih skupina, testnih scenarija i

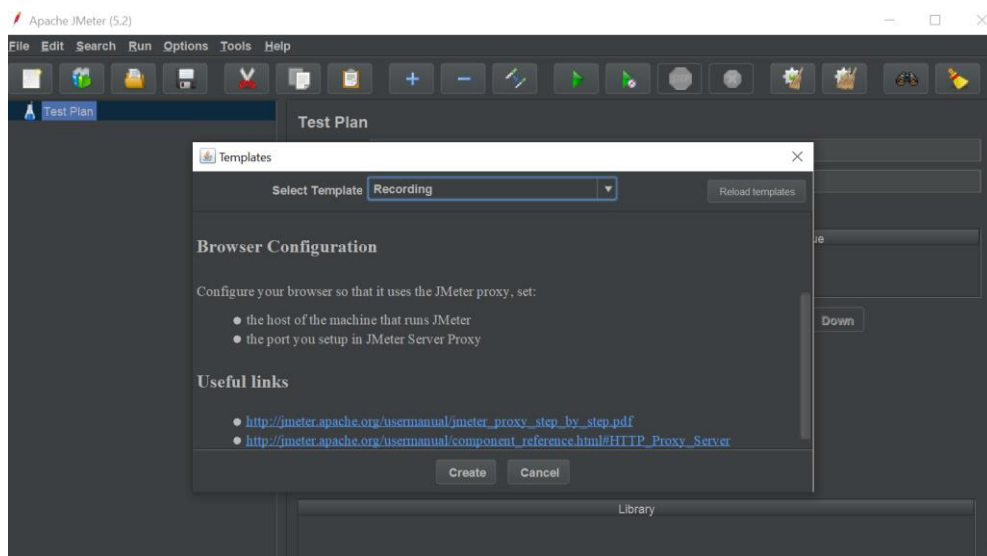
postavljenih alata sa sigurnošću može se provesti test opterećenja jer se zna što se traži i što se od web sustava očekuje.

## 7.2. Provedba testa opterećenja

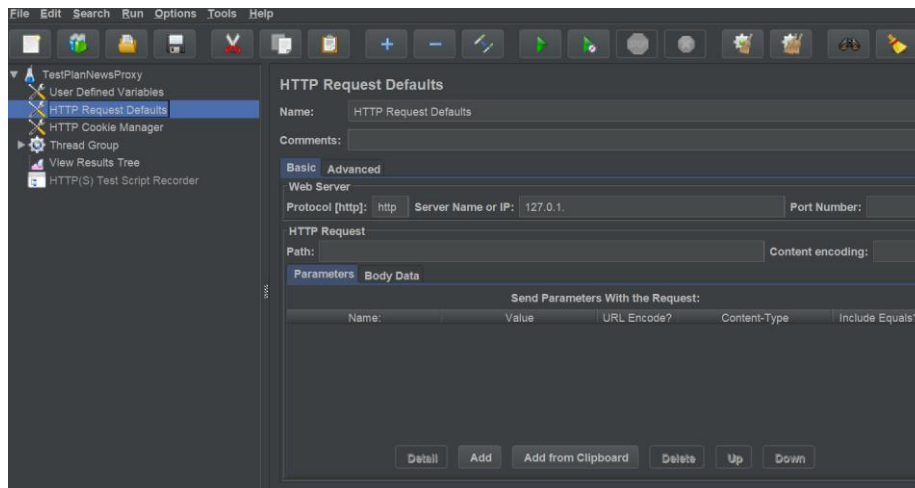
U ovom poglavlju prikazat će se provedbu testa opterećenja u alatima JMeter i Visual Studio Enterprise 2017. Pristupi testovima u ovim alatima su različiti pa tako će se i provedba razlikovati, ali rezultati bi trebali biti podjednaki. Visual Studio koristi snimalicu na Internet Exploreru što nije najbolje rješenje jer se Internet Explorer više ne koristi toliko koliko prije, a ne može biti ugašen jer je vezan za Windows 7. Na kraju provedbe moći će se zaključiti kojim alatom brže i bolje provesti test opterećenja i koji način je najsigurniji da se provjere performanse koje su potrebne.

### 7.2.1. Provedba testa opterećenja u alatu JMeter

Za početak u JMeter koristit će se predložak „Snimanje“ kako bi postojala mogućnost snimanja definiranih testnih scenarija. Na slici 7.1 može se vidjeti odabir predložaka. Nakon odabira predloška otvara se projekt koji se sastoji od naziv test plana, HTTP zahtjeva, HTTP kolačića, grupne niti i snimalice. Nakon postavljanja projekta potrebno je zapisati http zahtjev web sustava koji će se koristiti, u ovom diplomskom radu koristit će se postavke prikazane na slici 7.2.



Sl. 7.1. Odabir predloška „Snimanje“

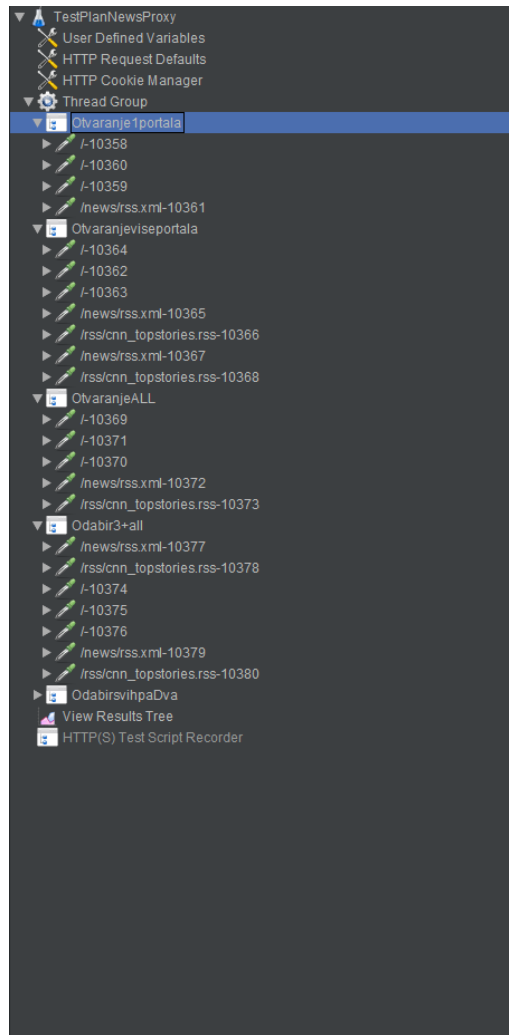


**Sl. 7.2.** *Postavke za HTTP zahtjev*

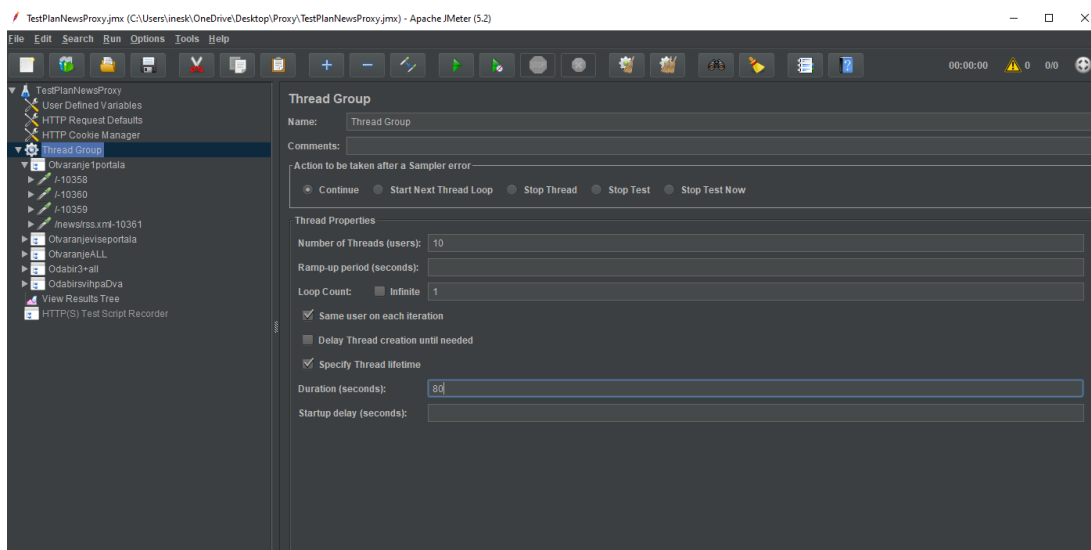
Postavljanjem projekta završava priprema u JMeter-u, još ostaje postavljanje posredničkog poslužitelja Proxy. Web sustav prikaz vijesti News testirat će se u pregledniku Chrome pa postavke za korištenje Proxy-a su jednostavne. Na svom računalu tj. računalu na kojem će se testiranje izvoditi uključuje se Proxy poslužitelj sljedećim koracima. Na Windows 10 platformi u tražilicu se upiše riječ Proxy te se odabire promjena postavki poslužitelja Proxy. Nakon otvaranja postavki uključuje se Proxy i dodaje se adresa i priključak. Postavljanjem postavki u JMeter i postavljanjem Proxy-a sve je spremno za snimanje testnih scenarija i spremanje API metoda web sustava prikaza vijesti u JMeter, prikaz snimljenih testnih scenarija i njihovi API metoda nalazi se na slici 7.3.

Kada su snimljene skripte spremne još je potrebno definirati testnu grupu, a ona se definira u elementu „Thread Group“. Primjer postavki za testnu grupu vidljivi su na slici 7.4, postavljeni podatci mogu se iščitati u poglavlju 7, tablice 7.2 pod nazivom „Testne grupe“.





Sl. 7.3. Snimljene API metode za testne scenarije



Sl. 7.4. Postavke za 10 virtualnih korisnika

Nakon postavljanja svih stavki u JMeter potrebno je napraviti .csv dokument u koji se spremaju rezultati i potrebno je napraviti folder u koji će se spremati generirani izvještaj. Test se može pokretati direktno u JMeter-u ili putem konzole. Preporučuje se putem konzole ako se testovi vrte na lokalnom poslužitelju kako bi bilo što manje komplikacija i nesporazuma, a prikaz pokretanja nalazi se na slici 7.5.

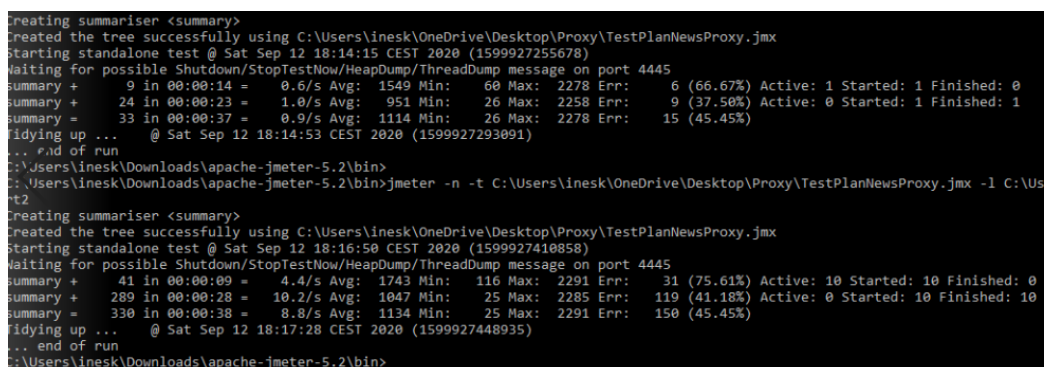
```
jmeter -n -t C:\Users\inesk\OneDrive\Desktop\Blaze\TestPlanBlaze.jmx
-l C:\Users\inesk\OneDrive\Desktop\Blaze\result80UserB.csv
-e -o C:\Users\inesk\OneDrive\Desktop\Blaze\htmlReportB\Report80B
```

**Sl. 7.5.** Naredba za pokretanje testa opterećenja u JMeter-u

Naredbena linija za pokretanje testa sastoji se od:

- JMeter - označava program u kojem se vrši radnja,
- -n - označava način rada bez GUI-a,
- -t – označava lokaciju JMeter skripte,
- Putanja skripte,
- -l – označava lokaciju .csv datoteke ili datoteke u koju se spremaju rezultati,
- Putanja do .csv datoteke,
- -e – označava izradu html izvještaja,
- -o – označava izradu mape izvještaja,
- Putanja do mape gdje će biti spremljen html izvještaj.

Primjer dobivenog izvještaja tijekom provođenja testa opterećenja vidljiv je na slici 7.6, a također se može vidjeti naredba „end of run“ koja ukazuje da je test završio.



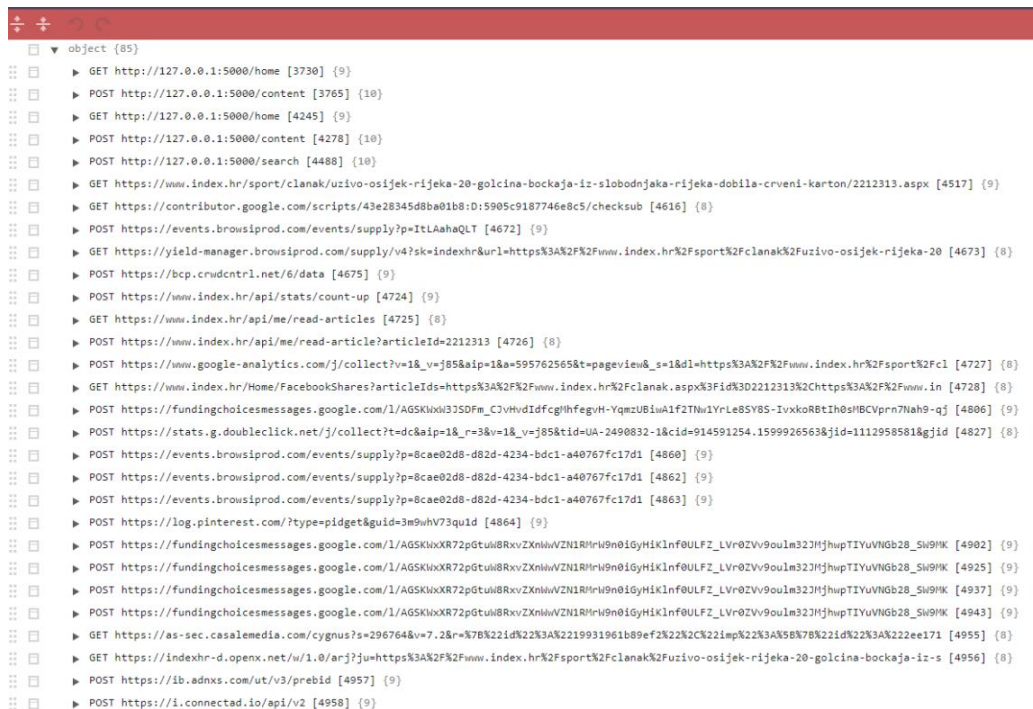
```
Creating summariser <summary>
Created the tree successfully using C:\Users\inesk\OneDrive\Desktop\Proxy\TestPlanNewsProxy.jmx
Starting standalone test @ Sat Sep 12 18:14:15 CEST 2020 (1599927255678)
Waiting for possible Shutdown/StopTestNow/HeapDump/ThreadDump message on port 4445
summary + 9 in 00:00:14 = 0.6/s Avg: 1549 Min: 60 Max: 2278 Err: 6 (66.67%) Active: 1 Started: 1 Finished: 0
summary + 24 in 00:00:23 = 1.0/s Avg: 951 Min: 26 Max: 2258 Err: 9 (37.50%) Active: 0 Started: 1 Finished: 1
summary = 33 in 00:00:37 = 0.9/s Avg: 1114 Min: 26 Max: 2278 Err: 15 (45.45%)
tidying up ... @ Sat Sep 12 18:14:53 CEST 2020 (1599927293091)
.. end of run
C:\Users\inesk\Downloads\apache-jmeter-5.2\bin>
C:\Users\inesk\Downloads\apache-jmeter-5.2\bin>jmeter -n -t C:\Users\inesk\OneDrive\Desktop\Proxy\TestPlanNewsProxy.jmx -l C:\US
rt2
Creating summariser <summary>
Created the tree successfully using C:\Users\inesk\OneDrive\Desktop\Proxy\TestPlanNewsProxy.jmx
Starting standalone test @ Sat Sep 12 18:16:50 CEST 2020 (1599927410858)
Waiting for possible Shutdown/StopTestNow/HeapDump/ThreadDump message on port 4445
summary + 41 in 00:00:09 = 4.4/s Avg: 1743 Min: 116 Max: 2291 Err: 31 (75.61%) Active: 10 Started: 10 Finished: 0
summary + 289 in 00:00:28 = 10.2/s Avg: 1047 Min: 25 Max: 2285 Err: 119 (41.18%) Active: 0 Started: 10 Finished: 10
summary = 330 in 00:00:38 = 8.8/s Avg: 1134 Min: 25 Max: 2291 Err: 150 (45.45%)
tidying up ... @ Sat Sep 12 18:17:28 CEST 2020 (1599927448935)
.. end of run
C:\Users\inesk\Downloads\apache-jmeter-5.2\bin>
```

**Sl. 7.6.** Konzola za vrijeme provođenja testa opterećenja

Izvješće tijekom provođenja testiranja sastoji se od sažetka prosječnog vremena izvođenja, minimalnog i maksimalnog vremena izvođenja te broja aktivnih i startanih korisnika kao i broja korisnika koji su završili svoje putanje. Završetkom testa radi se provjera spremljenih podataka i nastavlja se dalje mijenjam vrijednosti testne grupe.

Završetkom rada s poslužiteljem Proxy, gasi se poslužitelj i na preglednik Chrome dodaje ekstenzija BlazeMeter. Provođenje testa opterećenja je jednak kao što se već spomenulo, jedina razlika je u snimanje skripti.

Otvaranjem preglednika Chrome i otvaranjem ekstenzije BlazeMeter nudi se mogućnost naziva testa i njegovo pokretanje, nakon što se snimalica pokrene i snimi sve potrebne API metode, ekstenzija nudi spremanje u više različitih formata kao što su .jmx, .yml, selenium, itd. Osim što nudi spremanje, nudi i uređivanje API metoda pa na slici 7.7 vidljive su API metode dohvaćene putem ekstenzije.



```
object (85)
  GET http://127.0.0.1:5000/home [3730] (9)
  POST http://127.0.0.1:5000/content [3765] (10)
  GET http://127.0.0.1:5000/home [4245] (9)
  POST http://127.0.0.1:5000/content [4278] (10)
  POST http://127.0.0.1:5000/search [4488] (10)
  GET https://www.index.hr/sport/clanak/uzivo-osijek-rijeka-20-golcina-bockaja-iz-slobodnjaka-rijeka-dobila-crveni-karton/2212313.aspx [4517] (9)
  GET https://contributor.google.com/scripts/43e28345d8ba01b8:D:5905c9187746e8c5/checksub [4616] (8)
  POST https://events.browsiprod.com/events/supply?p=ItLaahQLT [4672] (9)
  GET https://yield-manager.browsiprod.com/supply/v4?sk=indexhr&url=https%3A%2F%2Fwww.index.hr%2Fsport%2Fclanak%2Fuzivo-osijek-rijeka-20 [4673] (8)
  POST https://bcp.crudcntrl.net/6/data [4675] (9)
  POST https://www.index.hr/api/stats/count-up [4724] (9)
  GET https://www.index.hr/api/me/read-articles [4725] (8)
  POST https://www.index.hr/api/me/read-article?articleId=2212313 [4726] (8)
  POST https://www.google-analytics.com/j/collect?v=1&v=j858aip=1&a=595762565&t=pageview&_s=1&d1=https%3A%2F%2Fwww.index.hr%2Fsport%2Fcl [4727] (8)
  GET https://www.index.hr/Home/FacebookShares?articleIds=https%3A%2F%2Fwww.index.hr%2Fclanak.aspx%3Fid%3D2212313%2Chttps%3A%2F%2Fwww.in [4728] (8)
  POST https://fundingchoicesmessages.google.com/L/AGSKlxXR72pGtuw8RkvZXnlwVZ1RrW9n0IgyHikInf0ULFZ_LVr0ZVv9ouIm32JHjhwTiyuVngb28_Sw9MK [4806] (9)
  POST https://stats.g.doubleclick.net/j/collect?t=dc&ip=1&r=3&v=1&v=j858tid=UA-2490832-1&cid=914591254.1599926563&jid=1112958581&gid [4827] (8)
  POST https://events.browsiprod.com/events/supply?p=8cae02d8-d82d-4234-bdc1-a40767fc17d1 [4860] (9)
  POST https://events.browsiprod.com/events/supply?p=8cae02d8-d82d-4234-bdc1-a40767fc17d1 [4862] (9)
  POST https://events.browsiprod.com/events/supply?p=8cae02d8-d82d-4234-bdc1-a40767fc17d1 [4863] (9)
  POST https://log.pinterest.com/?type=pidget&guid=3m9whV73qu1d [4864] (9)
  POST https://fundingchoicesmessages.google.com/L/AGSKlxXR72pGtuw8RkvZXnlwVZ1RrW9n0IgyHikInf0ULFZ_LVr0ZVv9ouIm32JHjhwTiyuVngb28_Sw9MK [4902] (9)
  POST https://fundingchoicesmessages.google.com/L/AGSKlxXR72pGtuw8RkvZXnlwVZ1RrW9n0IgyHikInf0ULFZ_LVr0ZVv9ouIm32JHjhwTiyuVngb28_Sw9MK [4925] (9)
  POST https://fundingchoicesmessages.google.com/L/AGSKlxXR72pGtuw8RkvZXnlwVZ1RrW9n0IgyHikInf0ULFZ_LVr0ZVv9ouIm32JHjhwTiyuVngb28_Sw9MK [4937] (9)
  POST https://fundingchoicesmessages.google.com/L/AGSKlxXR72pGtuw8RkvZXnlwVZ1RrW9n0IgyHikInf0ULFZ_LVr0ZVv9ouIm32JHjhwTiyuVngb28_Sw9MK [4943] (9)
  GET https://as-sec.casalemedia.com/cygnus?s=296764&v=7.2&r=%7B%221d%22%3A%2219931961b89ef2%22%2C%221mp%22%3A%5B%7B%221d%22%3A%222ee171 [4955] (8)
  GET https://indexhr-d.openx.net/w/1.0/ar?j= https%3A%2F%2Fwww.index.hr%2Fsport%2Fclanak%2Fuzivo-osijek-rijeka-20-golcina-bockaja-iz-s [4956] (8)
  POST https://ib.adnxs.com/ut/v3/prebid [4957] (9)
  POST https://i.connectad.io/api/v2 [4958] (9)
```

**Sl. 7.7.** API metode snimljeni ekstenzijom BlazeMeter

API metode mogu se još proširiti kao što je prikazano na slici 7.8 te se dobiju detaljnije informacije sa stranice, dohvaćeni podaci i navigacija po stranici.



**Sl. 7.8.** Proširene API metode u BlazeMeter-u

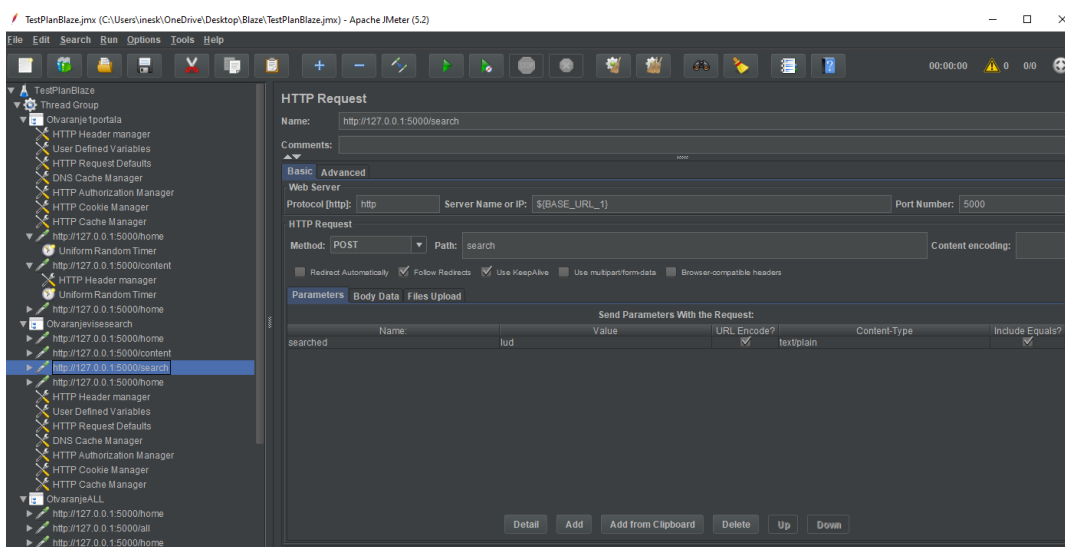
Nakon snimljene skripte, sprema se u .jmx format na računalo. Otvara se alat JMeter i učitavaju se snimljene skripte, prva se otvara dok se druge spajaju na otvorenu skriptu. Nakon slaganja i pripreme skripte, započinje isti proces kao s poslužiteljem Proxy. Vrijednosti za testne grupe se ponovo provode, na isti način se pokreće test i sprema kao što je vidljivo na slici 7.9, dok slika 7.10 prikazuje snimljene skripte s BlazeMeter-om u JMeter-u.

```

Creating summariser <summary>
Created the tree successfully using C:\Users\inesk\OneDrive\Desktop\Blaze\TestPlanBlaze.jmx
Starting standalone test @ Sat Sep 12 18:30:27 CEST 2020 (1599928227595)
Waiting for possible Shutdown/StopTestNow/HeapDump/ThreadDump message on port 4445
summary + 2 in 00:00:08 = 0.3/s Avg: 3718 Min: 125 Max: 7312 Err: 0 (0.00%) Active: 1 Started: 1 Finished: 0
summary + 4 in 00:00:51 = 0.1/s Avg: 2285 Min: 28 Max: 7631 Err: 0 (0.00%) Active: 0 Started: 1 Finished: 1
summary = 6 in 00:00:59 = 0.1/s Avg: 2763 Min: 28 Max: 7631 Err: 0 (0.00%)
Tidying up ... @ Sat Sep 12 18:31:26 CEST 2020 (1599928286520)
... end of run
C:\Users\inesk\Downloads\apache-jmeter-5.2\bin>jmeter -n -t C:\Users\inesk\OneDrive\Desktop\Blaze\TestPlanBlaze.jmx -l C:\Users\inesk\Downloads\apache-jmeter-5.2\bin\jmeter.log
Creating summariser <summary>
Created the tree successfully using C:\Users\inesk\OneDrive\Desktop\Blaze\TestPlanBlaze.jmx
Starting standalone test @ Sat Sep 12 18:32:12 CEST 2020 (1599928332420)
Waiting for possible Shutdown/StopTestNow/HeapDump/ThreadDump message on port 4445
summary + 21 in 00:00:47 = 0.4/s Avg: 19849 Min: 22 Max: 44994 Err: 0 (0.00%) Active: 10 Started: 10 Finished: 0
summary + 27 in 00:00:33 = 0.8/s Avg: 2278 Min: 21 Max: 18180 Err: 0 (0.00%) Active: 9 Started: 10 Finished: 1
summary = 48 in 00:01:20 = 0.6/s Avg: 9966 Min: 21 Max: 44994 Err: 0 (0.00%)
summary + 5 in 00:00:28 = 0.2/s Avg: 27169 Min: 3762 Max: 35989 Err: 0 (0.00%) Active: 4 Started: 10 Finished: 6
summary = 53 in 00:01:48 = 0.5/s Avg: 11589 Min: 21 Max: 44994 Err: 0 (0.00%)
summary + 3 in 00:00:02 = 1.4/s Avg: 37133 Min: 36584 Max: 38325 Err: 0 (0.00%) Active: 0 Started: 10 Finished: 10
summary = 56 in 00:01:50 = 0.5/s Avg: 12957 Min: 21 Max: 44994 Err: 0 (0.00%)
Tidying up ... @ Sat Sep 12 18:34:03 CEST 2020 (1599928443145)
... end of run

```

Sl. 7.9. Prikaz konzole testa snimljenih skripti s BlazeMeter-om



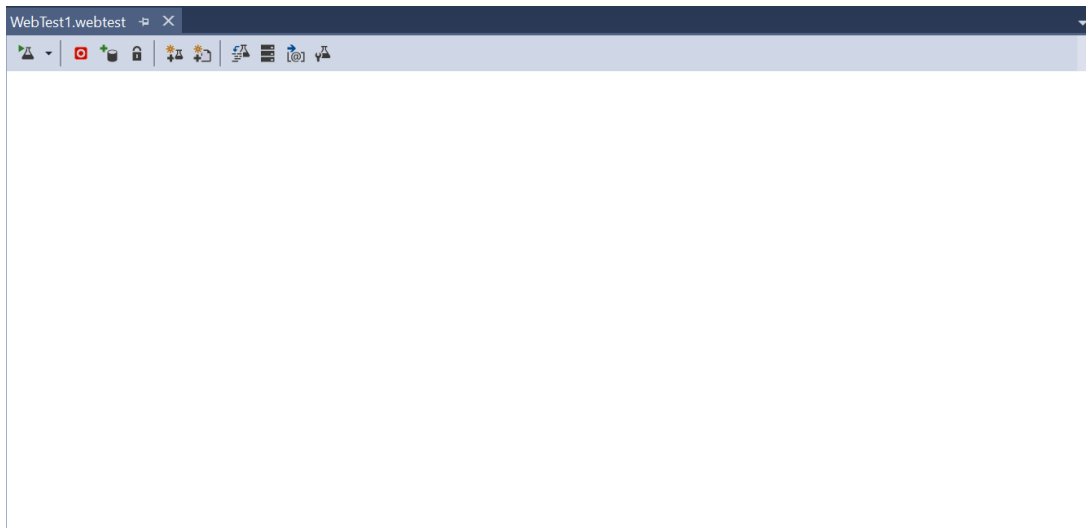
Sl. 7.10. API metode snimljene BlazeMeter-om u JMeter-u

Snimanjem skripti s poslužiteljem Proxy i ekstenzijom BlazeMeter snimljene su različite API metode. Tijekom snimanja s poslužiteljem Proxy nisu ni svi portali bili dohvaćeni, mogao se dohvatiti portal BBC News i CNN dok Index.hr, RTL, The New York Times i SPORT News iz nekog razloga nisu bili dohvaćeni. Osim problema s dohvaćanjem API metoda određenih portala, putem poslužitelja Proxy dohvaćene su API metode koje se ne nalaze na stranici niti predstavljaju pravu API metodu. Također slike nisu bile prikazane jer ih nije mogao učitati pa se ne dobije prava slika web sustava. Korištenjem BlazeMeter-a vide se sve API metode i što se događa i što se koristilo te je puno detaljniji pristup. Razlika u dva različita pristupa snimanja skripti odrazit će se i na dobivene rezultate.

### 7.2.2. Provedba testa opterećenja u alatu Visual studio

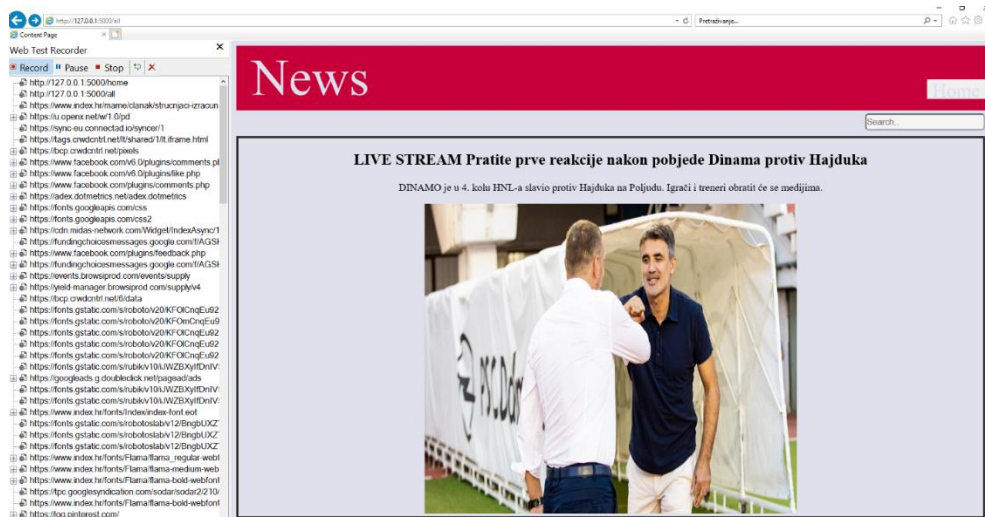
U ovom poglavlju bit će objašnjeni koraci postavljanja i provedbe testa opterećenja u alatu Visual Studio Enterprise 2017. Kao što je već navedeno u prijašnjim poglavljima, test opterećenja nije moguće provoditi u novim verzijama alata Visual Studio pa iz tog razloga se koristi verzija 2017. godine. Priprema je već objašnjena i jednaka je kao i za JMeter dok se sama provedba razlikuje. Pokretanjem Visual Studija potrebno je pokrenuti kao administrator kako bi se test opterećenja mogao provoditi na lokalno poslužitelju.

Prije svega potrebno je napraviti novi projekt, odabrati test, a zatim test performansi, potrebno je dati naziv projektu i spremiti projekt. Nakon napravljenog projekta pojavljuje se početna stranica kao na slici 7.11. gdje se može vidjeti da na lijevoj strani se nalazi snimalica (crveni kvadratić s bijelim krugom) i prije svega potrebno je snimiti skripte. U ovom slučaju snimit će se jedna skripta ali će imati više koraka, neće biti jednako kao i s JMeter-om.



**Sl. 7.11.** Početni prikaz kreiranog projekta

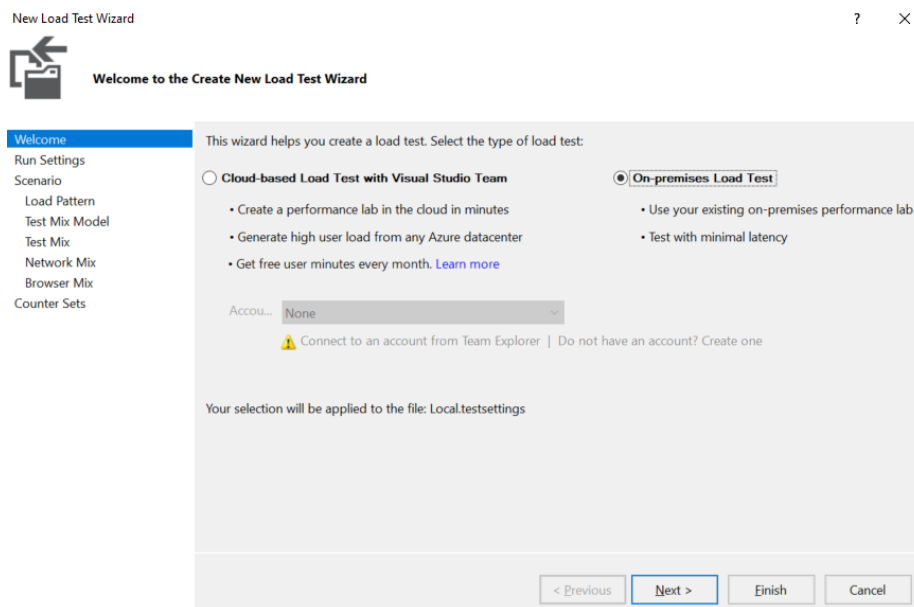
Na slici 7.12 prikazan je rad snimalice i kako snima API metode u Internet Exploreru te kako se metode spremaju.



**Sl. 7.12.** Snimanje testnih skripti u Internet Exploreru

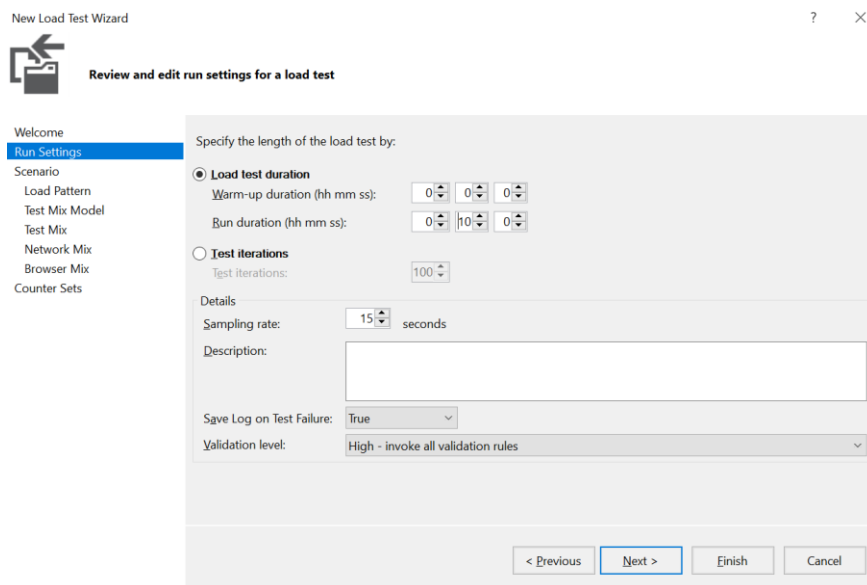
Nakon snimljenih skripti slijedi kreiranje testa opterećenja, potrebno je dodati test opterećenja tako što se u desnom prozorčiću kliknem na projekt desnim klikom zatim na „Dodaj“ pa „Test opterećenja“. Dodavanjem testa opterećenja otvara novi prozor na kojem se postavljaju vrijednosti testa opterećenja, postavke će biti objašnjene u sljedećim koracima.

Prvi korak je odabir tipa testa opterećenja, hoće li se ono provoditi na Azuru DevOps ili na lokalno računalo, odabir za test opterećenja web sustava prikaza vijesti prikazan je na slici 7.13.



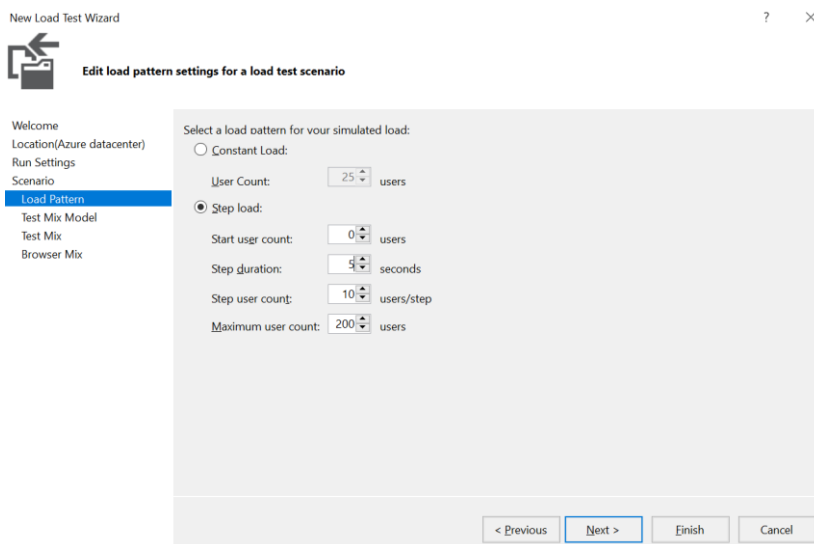
**Sl. 7.13.** Postavljanje tipa testa opterećenja

U drugom koraku potrebno je postaviti vrijeme izvođenja, u ovom slučaju izabrano je 10 minuta za cijeli test, prikaz kao na slici 7.14.



**Sl. 7.14.** Postavljanje vrijeme izvođenja testa opterećenja

Za treći korak je bitno dodati naziv scenarija koji će se koristiti, a onda odrediti testnu grupu kao što je prikazano na slici 7.15. Podaci sa slike navedeni su u prethodnom poglavlju tijekom pripreme testa opterećenja. Kako je već navedeno maksimalni broj virtualnih korisnika bit će 200 jer Visual Studio ne dopušta provedbu testa opterećenja s 250 virtualnih korisnika na lokalnom poslužitelju.

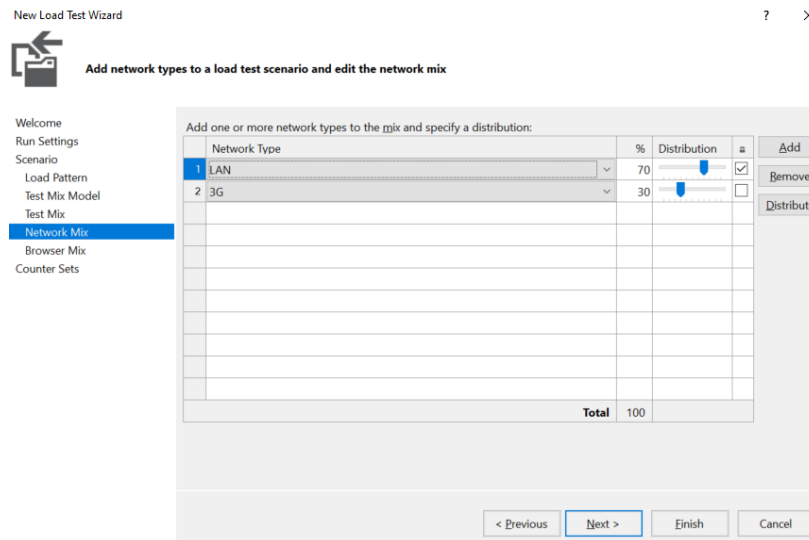


**Sl. 7.15.** Postavljanje testne grupe

Slijed četvrti korak u kojem se odabire kako će se test provoditi, ima više načina i svi su detaljno objašnjeni s desna strane, u ovom slušaju koristit će se model na konačnom broju testova. Detaljniji prikaz vidi se na slici 7.16.

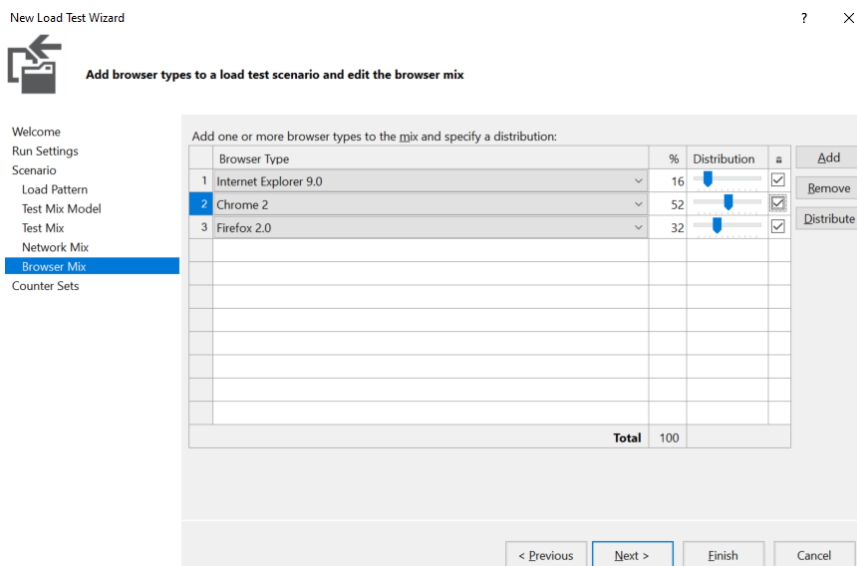






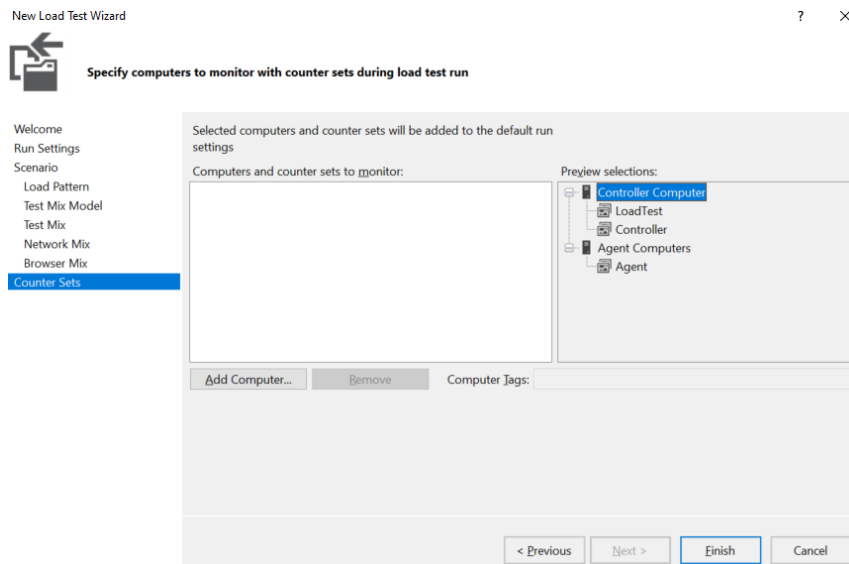
**Sl. 7.18.** Dodane brzine pri kojima će se provoditi test opterećenja

U sedmom koraku odabiru se preglednici u kojim će se test opterećenja provoditi te se odabire postotak koji će se koristiti najviše, a koji najmanje. Za testiranje web sustava prikaza vijesti News postavljen je preglednik Chrome s najvišim postotkom, zatim slijedi Firefox, a onda i Internet Explorer u kojem je testna skripta snimljena, prikaz odabira nalazi se na slici 7.19.



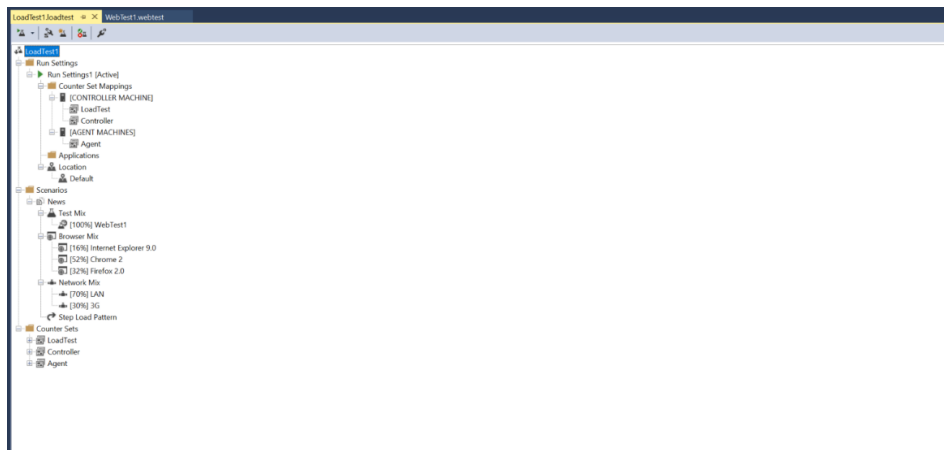
**Sl. 7.19.** Odabir preglednika

I za kraj postavljanja tj. osmi korak čini odabir na kojem računalu će se odvijati test opterećenja, odabir i izgled u ovom slučaju prikazan je na slici 7.20.



**Sl. 7.20.** Odabir računala

Nakon završetka postavljanja postavki, otvara se novi prozor u Visual Studiju koji treba izgledati kao na slici 7.21. Slijedi pregled svih postavki i ako je sve zadovoljeno pušta se test odabirom „Pokreni“ i čekaju se rezultati.



**Sl. 7.21.** Postavljen test opterećenja u alatu Visual Studio

Test opterećenja izvodi se 10 minuta, tijekom izvođenja mogu će je praćenje izrade grafova te obrada podataka.

Nakon provođenja testa opterećenja u alatu Visual Studio može se zaključiti da je provođenje u alatu JMeter jednostavnije, sučelje je više prilagođeno za API metode i postavljanje potrebnih stavki te je jasnije i razumljivije. Na rezultatima vidjet će se razlika provedbe u navedenim alatima.

## 8. REZULTATI FUNKCIONALNOG TESTIRANJA I TESTIRANJA OPTREĆENJA WEB SUSTAVA PRIKAZA VIJESTI

### 8.1. Rezultati funkcionalnog testiranja web sustava prikaza vijesti

Provođenjem funkcionalnog testiranja dobiveni su rezultati web testiranja. Rezultati prikazuju trenutni status projekta te ako rezultati nisu dobri i potrebne su velike izmjene tada se zaustavljaju sljedeće iteracije projekta i započinje se s ispravljanjem trenutnih problema i pogrešaka, ako su rezultati dobri može se nastaviti s daljnjim razvojem projekta, a usputno ispraviti pogreške ako ih ima.

Nakon pregleda i prijave pogrešaka, završava funkcionalno testiranje te se svi testni scenariji s rezultatima mogu vidjeti u tablici kao što je prikazano na slici 8.1, dok su na slici 8.2 prikazane prijavljene pogreške.

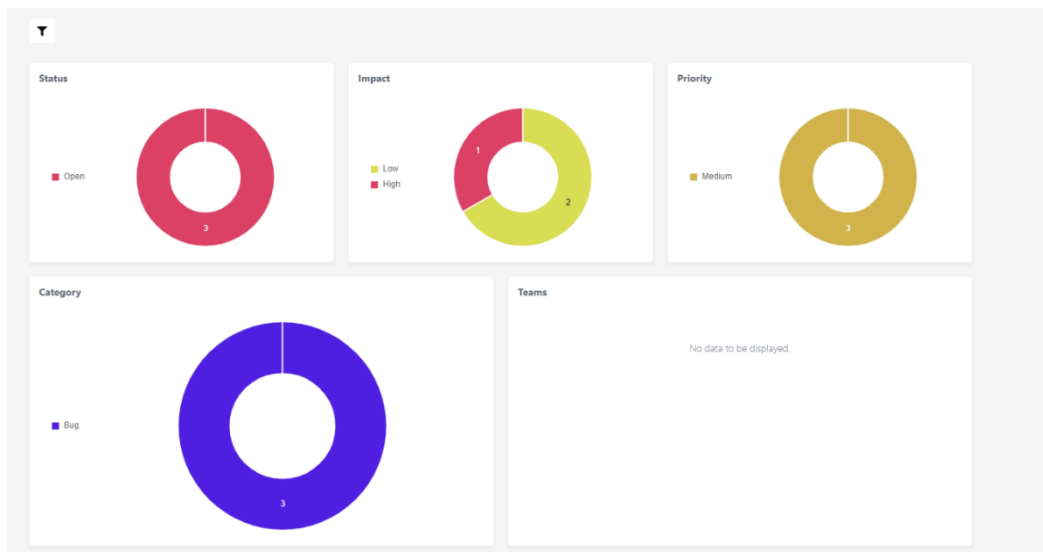
Code	Name	Progress	Status	Icon	Action
TC26	Osobit The New York Times Sport News	100%	Pass	😊	>
TC27	Komponenta Search	0%	Fail	🙄	>
TC28	Vlastruko klikanje na button ALL	100%	Pass	😊	>
TC29	Klik na button ALL	100%	Pass	😊	>
TC30	Pretraživanje u komponenti search	100%	Pass	😊	>
TC31	Vlastruko klikanje na button Sport	100%	Pass	😊	>
TC32	Klik na button Sport	100%	Pass	😊	>
TC33	Pretraživanje u komponenti search na Sportu	100%	Pass	😊	>
TC34	Izgled stranice na velikom ekranu	100%	Pass	😊	>
TC35	Izgled stranice na malom ekranu	100%	Pass	😊	>
TC36	Rad stranice pod različitim brzinama	100%	Pass	😊	>
TC37	Firefox	100%	Pass	😊	>
TC38	Chrome	100%	Pass	😊	>
TC39	Opera	100%	Pass	😊	>
TC40	Edge	100%	Pass	😊	>

Sl. 8.1. Prikaz rezultata testiranja

Code	Name	Status	Category	Impact	Action
I1	Tekst se oznaci kao da je klikabilan, a ne bi trebao biti Tekst se oznaci kao da je klikabilan, a ne bi trebao biti	Open	Bug	Low	>
I2	Naslov News Klikanjem na naslov News, tekst se oznaci kao da je klikabilan, a ne bi treba...	Open	Bug	Low	>
I3	Pretražujem riječ "boca", izbacij praznu stranicu bez poruke... Pretražujem riječ "boca", izbacij praznu stranicu bez poruke	Open	Bug	High	>

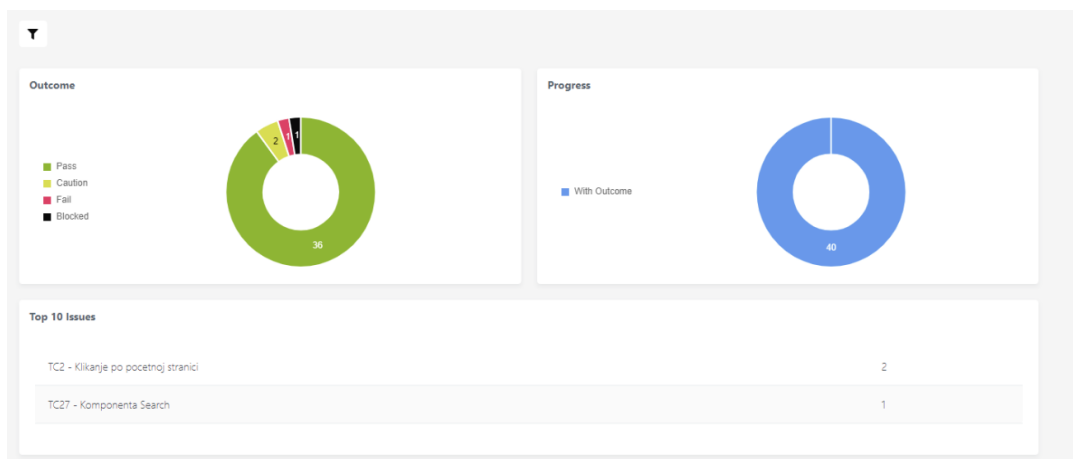
Sl. 8.2. Prikaz prijave pogrešaka

Osim prikaza rezultat u tablicama, web alat TestMonitor nudi izvoz izvještaja nakon provedenog web testiranja pa na slici 8.3 prikazani su grafovi koji prikazuju trenutnu situaciju rezultat testiranja web sustava prikaza vijesti.



**Sl. 8.3.** Prikaz izvještaja pogrešaka pomoću grafova

Također TestMonitor nudi izvještaj s grafovima koji sadrže testnih slučajeva te njihove statuse koji su vidljivi na slici 8.4.



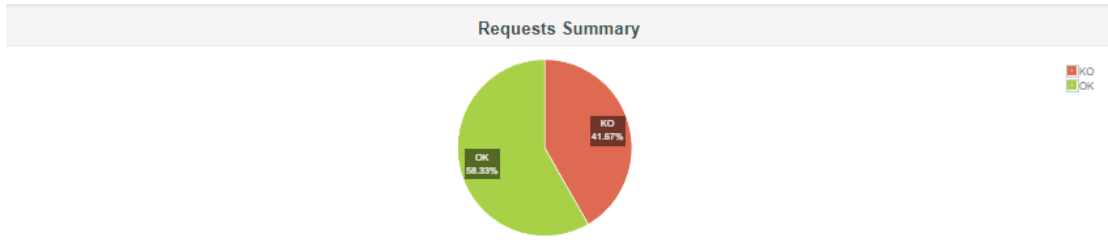
**Sl. 8.4.** Prikaz izvještaja testnih slučajeva pomoću grafova

## 8.2. Rezultati testa opterećenja web sustava prikaza vijesti

### 8.2.1. Rezultati dobiveni u programskom alatu JMeter i Proxy

Za spremanje dobivenih rezultata testiranja opterećenja web sustava prikaza vijesti News korišteni su html izvještaji. Html izvještaj je izvještaj koji sadrži tablice i grafove kojima je opisan provedeni test opterećenja, uz grafove i tablice u ovom diplomskom radu bit će priložena tablica dobivenih rezultat kako bi se mogla usporediti s tablicom očekivanih rezultata.

Na slici 8.5 nalazi se graf koji prikazuje sažetak zahtjeva tj. koliko zahtjeva je prošlo na web sustavu prikaza vijesti, a koliko nije dok je jedna osoba bila na web sustavu News.



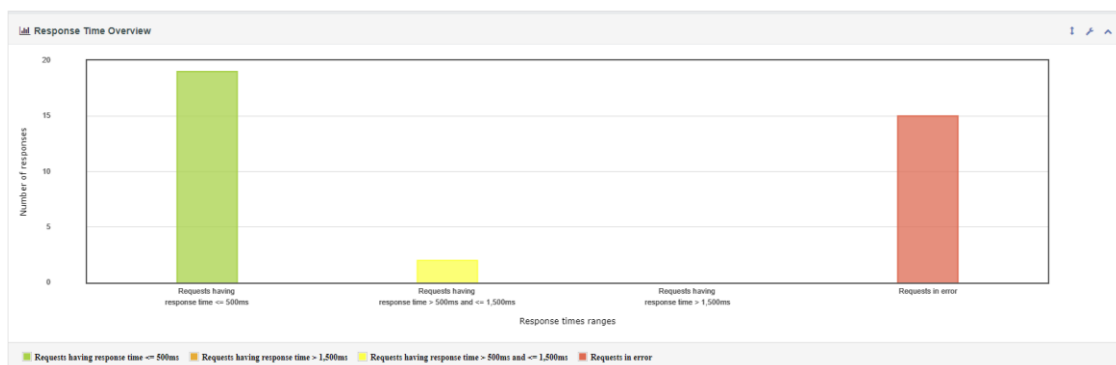
**Sl. 8.5.** Prikaz sažetka zahtjeva na web sustavu News s 1 korisnikom

Na prethodnoj slici mogu se vidjeti dvije oznake koje će se u nastavu dosta spominjati, a to su OK i KO. OK označava da su zahtjevi prošli dok KO označava zahtjeve koji su nisu prošli. Na grafu je vidljivo kako je 58.33% zahtjeva prošlo dok je 41.67% palo s 1 korisnikom opterećene testne skripte snimljene pomoću Proxy poslužitelja. Kako bi bilo jasnije shvaćeno što se događa i zašto određeni zahtjevi nisu prošli, u html izvještaju stoji tablica s jasnim prikazanom svih API metoda sa stranice. Na slici 8.6 može se vidjeti prikaz isječka tablice koja se sastoji od: naziva API metode, broja uzoraka, broja negativnih zahtjeva, prosječnog vremena izvođenja, minimalnog i maksimalnog vremena izvođenja, 90th pct što označava vrijednost za koju je 90% podatkovnih točaka manje, a 10% veće, 95th pct označava isto što i 90th pct samo za 95% podatkovnih točaka, 99th pct isto kao 90th pct i 95th pct samo za 99% podatkovnih točaka, zatim slijedi broj transakcija, broj primljenih i poslanih podataka. Ti podaci daju do znanja kako se stranica ponaša i pomoću tih vrijednosti mogu se zaključiti performanse stranice. Najpotrebniji podaci su prosječno vrijeme izvođenja „Average“, pomoću znanja tih vrijednosti zna se vrijeme izvođenja svakog testnog scenarija s obzirom na broj trenutnih korisnika.

Requests	Executions			Response Times (ms)						Throughput	Network (KB/sec)		
	Label	#Samples	KO	Error %	Average	Min	Max	90th pct	95th pct	99th pct	Transactions/s	Received	Sent
Total		36	15	41.67%	1055.33	26	2278	2257.30	2261.00	2278.00	0.98	16.31	0.43
/-10358		1	1	100.00%	2278.00	2278	2278	2278.00	2278.00	2278.00	0.44	0.98	0.00
/-10359		1	1	100.00%	2256.00	2256	2256	2256.00	2256.00	2256.00	0.44	0.99	0.00
/-10360		1	1	100.00%	2255.00	2255	2255	2255.00	2255.00	2255.00	0.44	0.99	0.00
/-10362		1	1	100.00%	2257.00	2257	2257	2257.00	2257.00	2257.00	0.44	0.99	0.00
/-10363		1	1	100.00%	2256.00	2256	2256	2256.00	2256.00	2256.00	0.44	0.99	0.00
/-10364		1	1	100.00%	2255.00	2255	2255	2255.00	2255.00	2255.00	0.44	0.99	0.00
/-10369		1	1	100.00%	2253.00	2253	2253	2253.00	2253.00	2253.00	0.44	0.99	0.00
/-10370		1	1	100.00%	2253.00	2253	2253	2253.00	2253.00	2253.00	0.44	0.99	0.00
/-10371		1	1	100.00%	2255.00	2255	2255	2255.00	2255.00	2255.00	0.44	0.99	0.00
/-10374		1	1	100.00%	2258.00	2258	2258	2258.00	2258.00	2258.00	0.44	0.99	0.00
/-10375		1	1	100.00%	2255.00	2255	2255	2255.00	2255.00	2255.00	0.44	0.99	0.00
/-10376		1	1	100.00%	2257.00	2257	2257	2257.00	2257.00	2257.00	0.44	0.99	0.00
/-10381		1	1	100.00%	2252.00	2252	2252	2252.00	2252.00	2252.00	0.44	0.99	0.00
/-10382		1	1	100.00%	2258.00	2258	2258	2258.00	2258.00	2258.00	0.44	0.99	0.00
/-10383		1	1	100.00%	2253.00	2253	2253	2253.00	2253.00	2253.00	0.44	0.99	0.00
/generate_204-10389		1	0	0.00%	78.00	78	78	78.00	78.00	78.00	12.82	1.04	4.31
/news/rss.xml-10361		1	0	0.00%	128.00	128	128	128.00	128.00	128.00	7.81	62.26	2.58
/news/rss.xml-10365		1	0	0.00%	60.00	60	60	60.00	60.00	60.00	16.67	132.83	5.50
/news/rss.xml-10367		1	0	0.00%	30.00	30	30	30.00	30.00	30.00	33.33	265.66	11.00
/news/rss.xml-10372		1	0	0.00%	84.00	84	84	84.00	84.00	84.00	11.90	94.88	3.93

### Sl. 8.6. Isječak tabličnog prikaza dobivenih rezultat s 1 korisnikom

Ono što je bitno za ovakvu vrstu provedbe testa je vrijeme odziva, na sljedećoj slici 8.7 prikazan je grafički pregled vremena odziva po zahtjevu.

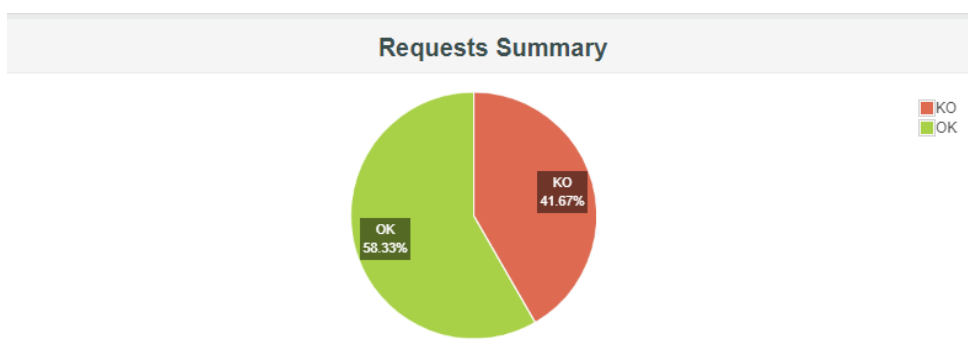


### Sl. 8.7. Vrijeme odziva s jednim korisnikom

Na x osi nalazi se raspon vremena odziva dok se na y osi nalazi broj odziva, u slučaju jednog korisnika na stranici broj, može se iščitati da je 19 zahtjeva ima vrijeme odziva manje ili jednako 500 ms, 2 zahtjeva ima vrijeme odziva više od 500 ms, a manje ili jednako od 1.500 ms, 0 zahtjeva je u rasponu više od 1.500 ms i 15 zahtjeva s pogreškom.

Može se zaključiti da vrijeme odziva na stranici dok se 1 korisnik nalazi na njoj, nije baš ni dobar, ali nije ni loš, korisnik mora duže čekati na odziv što će ga činiti nezadovoljnim i ubrzo će otići sa stranice.

Nakon dobivenih rezultata s 1 korisnikom, prelazi se na pregled rezultata s 10 korisnika. Na slici 8.8 nalazi se graf prikaza sažetka zahtjeva, isto kao i s 1 korisnikom. Oznake su iste pa ih ne treba ponovo objašnjavati.



### Sl. 8.8. Prikaz sažetka zahtjeva na web sustavu News s 10 korisnika

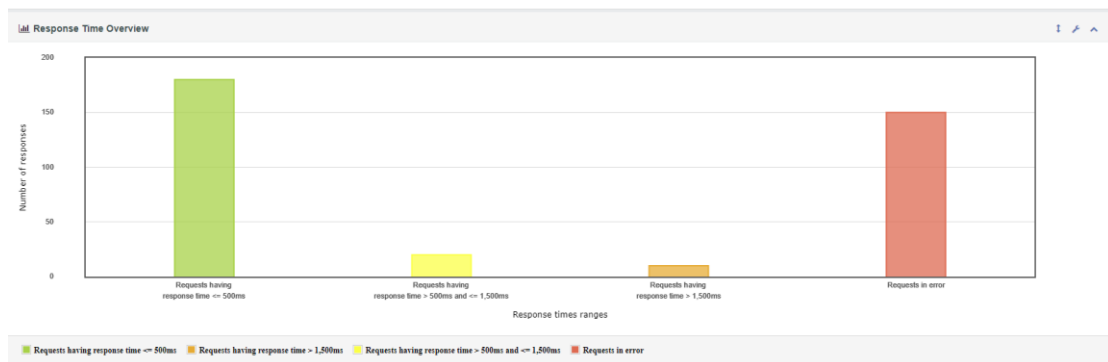
Može se vidjeti da dobiveni podaci su jednaki podacima izvršavanja testa s 1 korisnikom, razloge tome je korištenje Proxy-a i njegovih mogućnosti dohvaćanja. Na slici 8.9 nalazi se isječak tabličnog prikaza rezultata dobivenih s 10 korisnika na web sustavu prikaza vijesti.

Requests	Executions				Response Times (ms)				Throughput		Network (KB/sec)	
	# Samples	KO #	Error %	Average	Min #	Max #	50th pct #	90th pct #	Transactions #	Received #	Sent #	
Total	360	150	41.67%	1096.28	26	2291	2260.00	2262.00	2290.00	9.60	159.92	4.21
r-10358	10	10	100.00%	2209.70	2209	2291	2290.96	2291.00	2291.00	4.36	9.01	0.00
r-10359	10	10	100.00%	2256.40	2253	2263	2262.96	2263.00	2263.00	4.42	9.10	0.00
r-10360	10	10	100.00%	2255.00	2250	2262	2261.96	2262.00	2262.00	4.42	9.10	0.00
r-10362	10	10	100.00%	2254.90	2250	2260	2260.00	2260.00	2260.00	4.42	9.10	0.00
r-10363	10	10	100.00%	2258.00	2263	2264	2264.00	2264.00	2264.00	4.42	9.10	0.00
r-10364	10	10	100.00%	2249.40	2179	2262	2261.96	2262.00	2262.00	4.42	9.12	0.00
r-10369	10	10	100.00%	2244.10	2220	2265	2262.10	2265.00	2265.00	4.38	9.03	0.00
r-10370	10	10	100.00%	2258.00	2255	2262	2262.00	2262.00	2262.00	4.42	9.13	0.00
r-10371	10	10	100.00%	2244.00	2226	2261	2260.80	2261.00	2261.00	4.42	9.11	0.00
r-10374	10	10	100.00%	2224.70	2192	2259	2257.20	2259.00	2259.00	4.42	9.09	0.00
r-10375	10	10	100.00%	2254.30	2252	2258	2257.90	2258.00	2258.00	4.42	9.13	0.00
r-10376	10	10	100.00%	2256.30	2252	2261	2260.80	2261.00	2261.00	4.42	9.13	0.00
r-10381	10	10	100.00%	2256.00	2252	2259	2259.00	2259.00	2259.00	4.43	9.14	0.00
r-10382	10	10	100.00%	2256.30	2253	2261	2260.80	2261.00	2261.00	4.42	9.13	0.00
r-10383	10	10	100.00%	2235.00	2228	2254	2252.50	2254.00	2254.00	4.43	9.13	0.00
rgenerate_204-10389	10	0	0.00%	90.80	75	117	116.10	117.00	117.00	83.33	6.75	27.59
newsites.oms-10391	10	0	0.00%	132.30	116	194	187.00	194.00	194.00	50.76	414.34	16.76
newsites.oms-10395	10	0	0.00%	55.60	50	71	70.70	71.00	71.00	135.99	1121.27	45.22
newsites.oms-10397	10	0	0.00%	33.60	28	42	41.90	42.00	42.00	189.49	1387.33	55.95
newsites.oms-10372	10	0	0.00%	72.10	62	96	95.40	96.00	96.00	194.17	852.95	34.35
newsites.oms-10377	10	0	0.00%	32.60	28	37	36.60	37.00	37.00	116.20	955.17	38.38
newsites.oms-10379	10	0	0.00%	84.50	57	74	73.90	74.00	74.00	131.56	1001.05	43.43
newsites.oms-10384	10	0	0.00%	85.30	67	106	105.10	106.00	106.00	50.82	405.06	19.42
newsites.oms-10386	10	0	0.00%	32.00	25	45	44.20	45.00	45.00	74.63	615.25	24.63
nsactions_logstories-ns-10366	10	0	0.00%	192.30	180	200	207.70	208.00	208.00	40.08	1076.85	16.10
nsactions_logstories-ns-10368	10	0	0.00%	74.80	65	81	80.70	81.00	81.00	95.24	2133.18	31.90
nsactions_logstories-ns-10373	10	0	0.00%	140.30	125	149	148.00	149.00	149.00	57.14	1279.91	19.14
nsactions_logstories-ns-10378	10	0	0.00%	78.90	70	102	99.90	102.00	102.00	72.48	1823.08	24.27
nsactions_logstories-ns-10380	10	0	0.00%	151.00	138	158	158.00	158.00	158.00	61.35	1374.14	20.55
nsactions_logstories-ns-10385	10	0	0.00%	159.00	144	175	174.90	175.00	175.00	30.76	888.16	12.98
nsactions_logstories-ns-10387	10	0	0.00%	77.20	71	87	86.50	87.00	87.00	57.14	1279.91	19.14
serviceapi0823jpm70a2Nwv+1811178804016caZDhwq+e0554-B0K2N3C2698ad0276cd873c36797716e42115530375c64-10390	10	0	0.00%	112.90	103	121	120.90	121.00	121.00	80.65	1411.62	214.35
nsactions_logstories-3A2T2CVV8FO0Aivnd-fqppes-sembrucks-4ah-newsresource-org-om-10388	10	0	0.00%	2646.00	1975	2973	2972.80	2973.00	2973.00	4.81	825.34	9.50
nsactions_logstories-3A2T2CVV8FO0Aivnd-fqppes-sembrucks-4ah-newsresource-org-om-10388-0	10	0	0.00%	162.70	155	170	169.00	170.00	170.00	40.16	33.53	22.90
nsactions_logstories-3A2T2CVV8FO0Aivnd-fqppes-sembrucks-4ah-newsresource-org-om-10385-1	10	0	0.00%	797.50	722	816	815.70	816.00	816.00	12.24	8.52	6.38
nsactions_logstories-3A2T2CVV8FO0Aivnd-fqppes-sembrucks-4ah-newsresource-org-om-10388-2	10	0	0.00%	1874.50	1053	1094	1093.20	1094.00	1094.00	9.12	1558.82	7.15

Sl. 8.9. Isječak tabličnog prikaza dobivenih rezultata s 10 korisnika

U ovom diplomskom radu, tablične vrijednosti su objašnjene, ali neće biti detaljno obrađene, pomoću dobivenih tablica uzimat će se prosječno vrijeme te će se usporediti tablica očekivanih rezultata i tablica dobivenih rezultata.

Na sljedećoj slici 8.10 vidljiv je grafički prikaz vrijeme odziva s 10 korisnika. Grafički prikaz s 10 korisnika znatno se razlikuje od grafičkog prikaza s jednim korisnikom. Moguće je iščitati vrijednosti kako bi zaključili ponašanje web sustava prikaza vijesti News s 10 korisnika. Vidljivo je da 180 zahtjeva ima vrijeme odziva manje od 50 ms, njih 20 ima vrijeme odziva između 500 ms i 1.500 ms, njih 10 ima vrijeme odziva više od 1.500 ms, a 150 zahtjeva je s pogreškom.

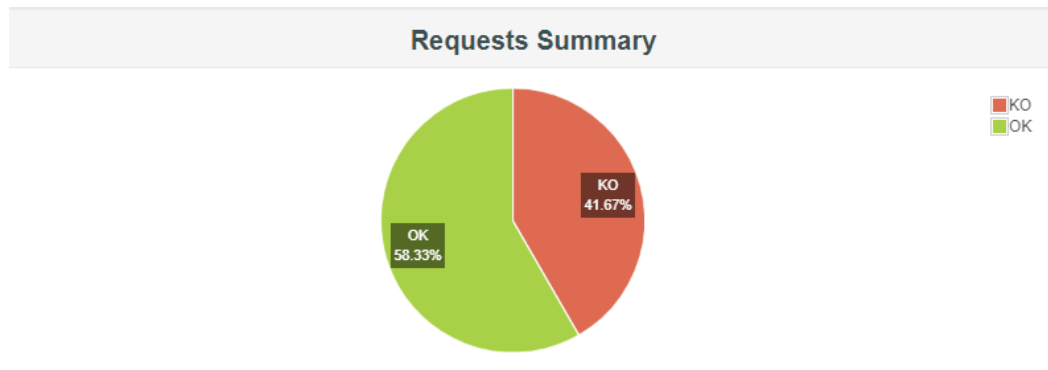


Sl. 8.10. Vrijeme odziva s 10 korisnika

Može se zaključiti da stranica nema najbolji odziv ni s 10 korisnika na njoj te bi svi morali čekati na svaki zahtjev po nekoliko sekundi i vjerojatno bi jedan dio korisnika odustao od pregleda vijesti na ovakvom web portalu.



Slika 8.11 sadrži grafički sažeti prikaz zahtjeva dok se na stranici nalazi 80 korisnika. Vidljivo je kako je grafički prikaz jednak onom s 1 korisnikom i s 10 korisnika.



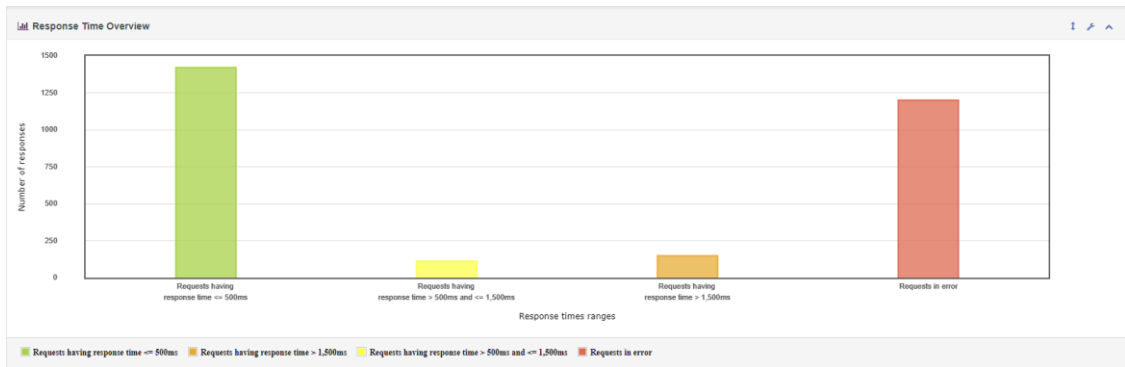
Sl. 8.11. Prikaz sažetka zahtjeva na web sustavu News s 80 korisnika

Nakon grafičkog prikaza slijedi isječak tabličnog prikaza koji daje jasnije informacije o provođenju test opterećenja na web sustavu prikaza vijesti News, na slici 8.12 vidljive su detaljnije informacije snimljenih API metoda.

Requests	Executions			Response Times (ms)					Throughput		Network (KB/sec)		
	Label	# Samples	KO #	Error %	Average	Min #	Max #	50th pct #	75th pct #	90th pct #	Transactions #	Received #	Sent #
Total		2880	1200	41.67%	1114.88	0	4815	2275.00	2313.95	3128.85	70.61	1176.95	30.85
/-10358		80	0	100.00%	765.96	0	2350	2119.90	2282.80	2350.00	20.95	42.91	0.00
/-10359		80	0	100.00%	2386.38	2224	2337	2259.80	2382.80	2337.00	34.23	89.90	0.00
/-10360		80	0	100.00%	1974.74	788	2386	2271.90	2274.00	2286.00	34.89	71.26	0.00
/-10362		80	0	100.00%	2281.00	2216	2340	2340.00	2344.95	2348.00	34.06	89.53	0.00
/-10363		80	0	100.00%	2223.60	2161	2291	2278.90	2285.90	2291.00	34.92	71.32	0.00
/-10364		80	0	100.00%	2253.63	2143	2300	2293.90	2298.95	2300.00	34.78	71.16	0.00
/-10369		80	0	100.00%	2107.74	2010	2307	2187.00	2215.90	2307.00	34.68	70.99	0.00
/-10370		80	0	100.00%	2262.78	2240	2284	2278.90	2282.90	2284.00	35.01	71.71	0.00
/-10371		80	0	100.00%	2241.81	2207	2280	2274.00	2278.00	2280.00	35.07	71.70	0.00
/-10374		80	0	100.00%	1986.90	1140	2252	2187.00	2182.95	2252.00	34.93	71.32	0.00
/-10375		80	0	100.00%	2291.10	2254	2326	2318.70	2324.90	2326.00	34.35	70.32	0.00
/-10376		80	0	100.00%	2240.71	2222	2275	2268.00	2271.95	2275.00	35.15	71.96	0.00
/-10381		80	0	100.00%	2255.03	2237	2275	2268.00	2271.00	2275.00	35.16	72.03	0.00
/-10382		80	0	100.00%	2251.15	2231	2271	2267.00	2268.95	2271.00	35.21	72.16	0.00
/-10383		80	0	100.00%	2150.27	1938	2282	2254.00	2270.75	2282.00	35.06	71.74	0.00
/generate_204-10389		80	0	0.00%	164.48	54	1007	134.00	289.70	1067.00	33.00	2.80	19.00
/news/ins.uni-10361		80	0	0.00%	137.58	66	238	163.00	201.30	238.00	312.90	2577.37	103.18
/news/ins.uni-10365		80	0	0.00%	91.29	56	112	110.90	111.95	112.00	533.33	4380.81	176.04
/news/ins.uni-10367		80	0	0.00%	70.25	38	165	105.90	108.90	165.00	230.55	1901.51	76.18
/news/ins.uni-10372		80	0	0.00%	231.40	63	1065	1055.90	1059.95	1065.00	73.60	807.02	24.28
/news/ins.uni-10377		80	0	0.00%	60.36	22	121	75.90	103.90	121.00	77.00	835.07	25.42
/news/ins.uni-10379		80	0	0.00%	186.26	77	1069	161.90	1084.45	1069.00	73.94	609.82	24.11
/news/ins.uni-10384		80	0	0.00%	185.05	67	1078	162.90	1071.85	1078.00	73.28	604.22	24.18
/news/ins.uni-10386		80	0	0.00%	68.54	26	144	97.00	131.65	144.00	75.47	822.47	24.91
/news/ins.uni-10386		80	0	0.00%	251.33	171	304	276.90	282.00	304.00	203.05	4547.91	68.01
/news/ins.uni-10386		80	0	0.00%	151.10	85	190	178.00	181.95	190.00	223.46	5005.24	74.85
/news/ins.uni-10373		80	0	0.00%	204.40	129	303	282.00	295.00	303.00	87.85	1519.83	22.73
/news/ins.uni-10378		80	0	0.00%	113.81	66	199	137.90	142.80	199.00	67.28	1507.04	22.54
/news/ins.uni-10380		80	0	0.00%	230.80	145	277	294.00	293.75	277.00	89.90	1506.80	23.28
/news/ins.uni-10385		80	0	0.00%	249.74	134	305	276.00	293.90	305.00	68.43	1532.83	22.92
/news/ins.uni-10387		80	0	0.00%	228.84	100	712	483.00	508.65	712.00	73.19	1639.41	24.52
/news/ins.uni-10388		80	0	0.00%	114.14	90	217	141.00	152.70	217.00	35.40	61.90	225.77
/news/ins.uni-10388		80	0	0.00%	2948.15	1747	4815	3357.00	3424.80	4815.00	16.25	2801.74	33.14
/news/ins.uni-10388		80	0	0.00%	166.00	150	317	178.00	188.65	317.00	89.44	57.98	39.61
/news/ins.uni-10388		80	0	0.00%	866.74	136	1165	1035.00	1089.70	1165.00	68.26	47.59	46.73
/news/ins.uni-10388		80	0	0.00%	1901.23	1286	3620	2198.00	2213.75	3620.00	21.86	3734.31	17.14

Sl. 8.12. Isječak tabličnog prikaza dobivenih rezultata s 80 korisnika

Još je potrebno prikazati vrijeme odziva koje puno govori o performansama stranice pa na slici 8.13 slijedi grafički prikaz vrijeme odziva zahtjeva s 80 ljudi.

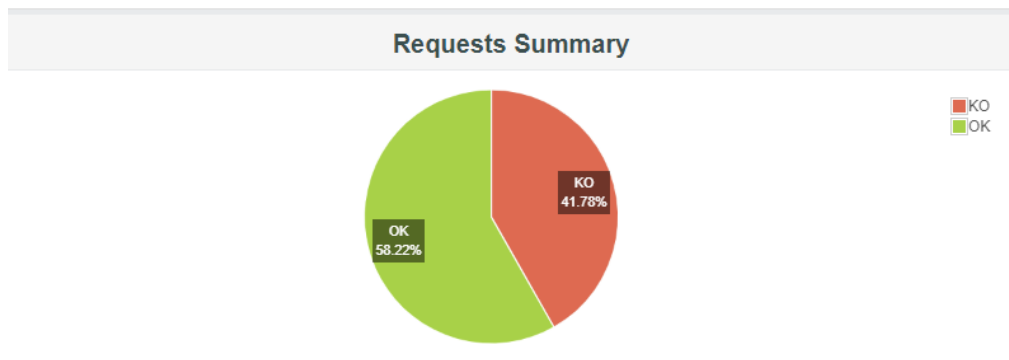


**Sl. 8.13.** *Vrijeme odziva s 80 korisnika*

Na grafu je vidljivo kako 1420 zahtjeva ima vrijeme odziva unutar 500 ms, 112 zahtjeva je unutar raspona 500 ms i 1.500 ms, a 148 zahtjeva ima vrijeme odziva iznad 1.500 ms dok je njih 1200 s pogreškom.

Također kao i do sad može se zaključiti da se stranica neće ponašati najbolje tj. njen odziv nije dovoljno dobar da izdrži 80 korisnika te da zadovolji njihove uvjete. Previše je zahtjeva s pogreškom da bi stranica za sve funkcionirala normalno.

Za kraj rada s posredničkim poslužiteljem Proxy, ostaje prikaz rezultata s 400 korisnika. Na slici 8.14 nalazi se grafički prikaz zahtjeva koji su prošli, a koji nisu i može se vidjeti da je postotak jednak onome do sada.



**Sl. 8.14.** *Prikaz sažetka zahtjeva na web sustavu News s 400 korisnika*

Zatim slijedi isječak tabličnog prikaza, iz kojeg se mogu pročitati potrebna vremena za izvođenje snimljenih testnih skripti te se dobivena vremena mogu zapisati u tablicu i kako je već navedeno, usporediti s očekivanim rezultatima, na slici 8.15 nalazi se isječak tabličnog prikaza rezultata testa opterećenja s 400 korisnika.



Broj korisnika	1	10	80	400
Testni scenariji				
Odabir jednog portala	0.9 sec	1.7 sec	3.2 sec	8.7 sec
Otvaranje više portala i pretraživanje	1.5sec	2.2 sec	6.7 sec	11.6 sec
Odabir opcije prikaza svih portala (ALL)	1.5 sec	2.3 sec	6.6 sec	9.1 sec
Odabir tri vijesti pa opciju svih portala (ALL)	0.9 sec	1.7 sec	2.2 sec	10.2 sec
Odabir tri vijesti pa opciju svih portala (ALL)	1.5 sec	2.3 sec	5.6 sec	10.4 sec
<b>UKUPNO</b>	6.3 sec	10.2 sec	24.3 sec	46 sec

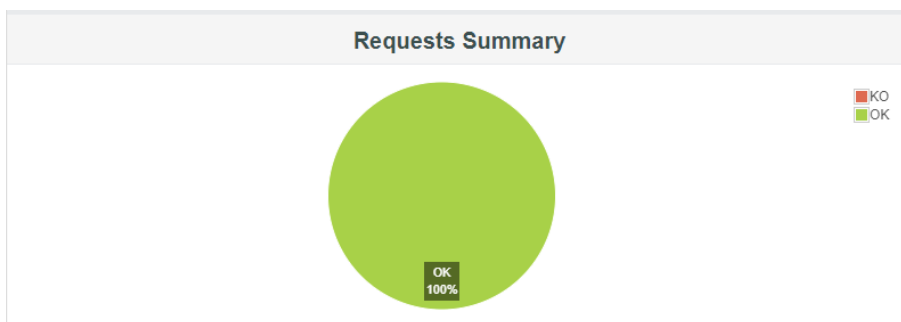
Usporedbenom tablice očekivani rezultata i dobivenih, jako je velika razlika. Kako posrednički poslužitelj Proxy nije snimio sve API metode sa stranice, ne dobivaju se ni realni podaci. Za izvedbe testnih skripti treba puno kraće vremena što nije moguće da 1 korisnik obavi u to vrijeme. Također se može zaključiti da vrijeme odziva nije najbolje ni u jednom slučaju, ni kad je 1 korisnik na web sustavu, a ni kad su 400 korisnika na web sustavu.

Osim usporedbe s očekivanim rezultatima, mogu se usporediti rezultati dobiveni u tablici pa se tako može vidjeti da je jednom korisniku potrebno 0.9 sec kako bi otvorio web sustav, odabrao portal i prelistao dobivene vijesti, što nije realnost da 1 korisnik u tom vremenu takvu radnju može napraviti. Također vrijeme potrebno da istu radnju izvrši 400 korisnika je 8.7 sec što također nije realno da toliki broj korisnika u tako malo vremenu izvršiti istu radnju.

Provedbom istog testa u JMeter-u i snimalicom BlazeMeter-a, zaključit će se je li problem u samom web sustavu prikaza vijesti News ili se problem stvara i prilikom snimanja.

### **8.2.2. Rezultati dobiveni u programskom alatu JMeter i BlazeMeter**

Isti postupak prikaza rezultata kao u prethodnom potpoglavlju provodit će se i u ovom potpoglavlju. Prvi rezultati objašnjeni su za jednog korisnika kao što se vidi na slici 8.17. Vidljiva je i razlika dobivenih rezultata s poslužiteljem Proxy i ekstenzijom BlazeMeter s istim postavkama za testne grupe i istim postupkom snimanja, samo drugim alatom.



**Sl. 8.17.** Prikaz sažetka zahtjeva na web sustavu News s 1 korisnikom koristeći BlazeMeter

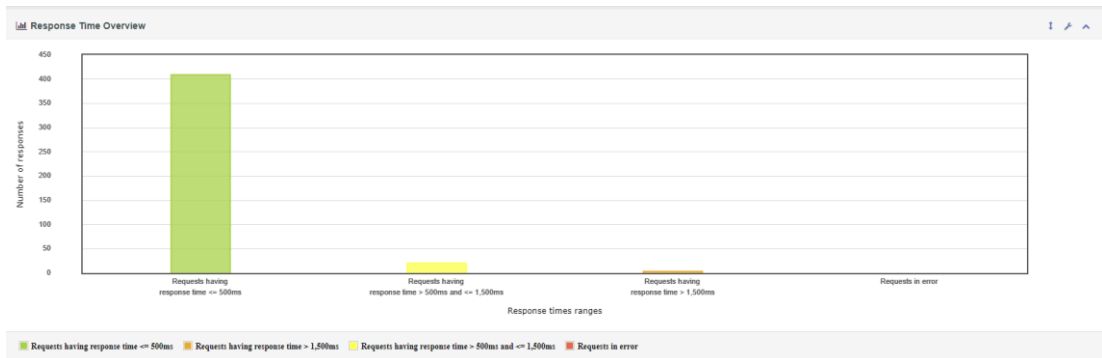
Na grafu se može vidjeti kako je u ovom slučaju tj. slučaju s 1 korisnikom prolaznost zahtjeva 100%, dok je s Proxy poslužiteljem bila 58.44%.

Sljedeći prikaz na slici 8.18. je isječak tabličnog prikaza te je iz njega potrebno prikupiti vremena za određene testne scenarije kako bi opet mogle vrijednosti usporediti s očekivanim vrijednostima.

Requests	Label	#Samples	Executions			Average	Min	Max	Response Times (ms)				Throughput		Network (KB/s)	
			KO	%	Error %				20th pct	50th pct	90th pct	95th pct	Transactions	Received	Sent	
Total		432	0	0.00%	218.95	3	7631	335.00	600.40	1451.38	7.41		3432.66	8.50		
http://127.0.0.1:5000/content		2	0	0.00%	7471.50	7312	7631	7631.00	7631.00	7631.00	0.04		2229.44	5.19		
http://127.0.0.1:5000/content-0		2	0	0.00%	606.00	347	865	865.00	865.00	865.00	0.05		6.50	0.03		
http://127.0.0.1:5000/content-1		2	0	0.00%	3.50	3	4	4.00	4.00	4.00	0.05		0.01	0.03		
http://127.0.0.1:5000/content-2		2	0	0.00%	553.50	279	825	825.00	825.00	825.00	0.05		9.34	0.03		
http://127.0.0.1:5000/content-3		2	0	0.00%	551.00	234	860	860.00	860.00	860.00	0.05		12.51	0.03		
http://127.0.0.1:5000/content-4		2	0	0.00%	537.50	288	789	789.00	789.00	789.00	0.05		6.30	0.03		
http://127.0.0.1:5000/content-5		2	0	0.00%	602.00	261	943	943.00	943.00	943.00	0.05		16.59	0.03		
http://127.0.0.1:5000/content-6		2	0	0.00%	531.50	231	832	832.00	832.00	832.00	0.05		16.43	0.03		
http://127.0.0.1:5000/content-7		2	0	0.00%	597.50	287	908	908.00	908.00	908.00	0.05		13.73	0.03		
http://127.0.0.1:5000/content-8		2	0	0.00%	310.50	236	385	385.00	385.00	385.00	0.05		11.88	0.03		
http://127.0.0.1:5000/content-9		2	0	0.00%	279.50	233	326	326.00	326.00	326.00	0.05		6.29	0.03		
http://127.0.0.1:5000/content-10		2	0	0.00%	171.50	3	340	340.00	340.00	340.00	0.05		5.92	0.03		
http://127.0.0.1:5000/content-11		2	0	0.00%	343.50	298	389	389.00	389.00	389.00	0.05		15.69	0.03		
http://127.0.0.1:5000/content-12		2	0	0.00%	316.00	293	339	339.00	339.00	339.00	0.05		21.94	0.03		
http://127.0.0.1:5000/content-13		2	0	0.00%	369.00	230	380	380.00	380.00	380.00	0.05		27.74	0.03		
http://127.0.0.1:5000/content-14		2	0	0.00%	175.00	54	296	296.00	296.00	296.00	0.05		6.45	0.03		
http://127.0.0.1:5000/content-15		2	0	0.00%	229.50	92	367	367.00	367.00	367.00	0.05		15.83	0.03		
http://127.0.0.1:5000/content-16		2	0	0.00%	144.50	62	227	227.00	227.00	227.00	0.05		9.07	0.03		
http://127.0.0.1:5000/content-17		2	0	0.00%	219.00	51	387	387.00	387.00	387.00	0.05		35.04	0.03		
http://127.0.0.1:5000/content-18		2	0	0.00%	109.50	47	172	172.00	172.00	172.00	0.05		3.51	0.03		
http://127.0.0.1:5000/content-19		2	0	0.00%	109.00	48	170	170.00	170.00	170.00	0.05		3.05	0.03		
http://127.0.0.1:5000/content-20		2	0	0.00%	132.50	51	214	214.00	214.00	214.00	0.05		7.80	0.03		
http://127.0.0.1:5000/content-21		2	0	0.00%	135.50	65	206	206.00	206.00	206.00	0.05		5.37	0.03		

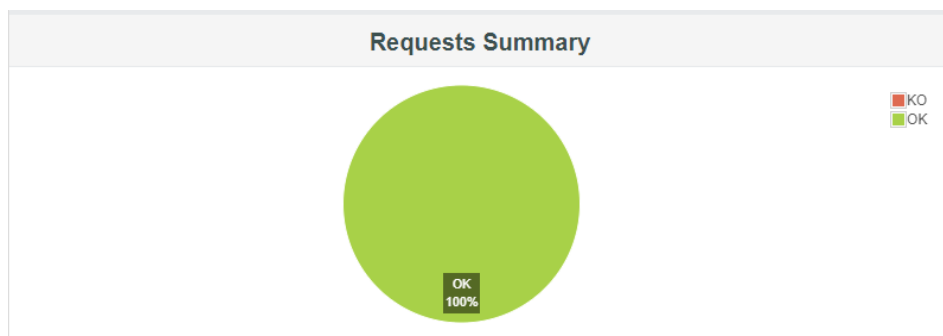
**Sl. 8.18.** Isječak Tabličnog prikaza dobivenih rezultata s 1 korisnikom koristeći BlazeMeter

Pošto je jako puno snimljenih API metoda, ne stane cijeli prikaz tablice pa je na slici 8.18 prikazan samo početak, pojmove koje sadrži tablica objašnjeni su u prethodnom potpoglavlju. Tablične vrijednosti su bitne kako bi došli do vremena za izvođenje testnih skripti. Na sljedećoj slici tj. slici 8.19 nalazi se grafički prikaz vrijeme odziva stranice s 1 korisnikom i pomoću tog grafa može se zaključiti kakve su performanse web sustava prikaza vijesti.



**Sl. 8.19.** *Vrijeme odziva s 1 korisnikom koristeći BlazeMeter*

Sa slike 8.19 dobivene su informacije da 409 zahtjeva ima vrijeme odziva ispod 500 ms, njih 20 ima vrijeme odziva u rasponu od 500 ms do 1.500 ms, a tri zahtjeva ima vrijeme odziva iznad 1.500 sekundi i nema niti jednog zahtjeva s pogreškom. Može se zaključiti da je stranica brza, korisnik neće čekati odgovor poslužitelja i neće biti ljut te će se poželjeti vratiti na web sustav prikaz vijesti jer ima veliko vrijeme odziva. Nakon testa opterećenja s 1 korisnikom, slijedi test opterećenja s 10 korisnika. Na slici 8.20 nalazi se grafički prikaz zahtjeva s 10 korisnika na web sustavu te se vidi jednakost s grafičkom prikazu zahtjeva s 1 korisnikom što čini pozitivne rezultate stranice.



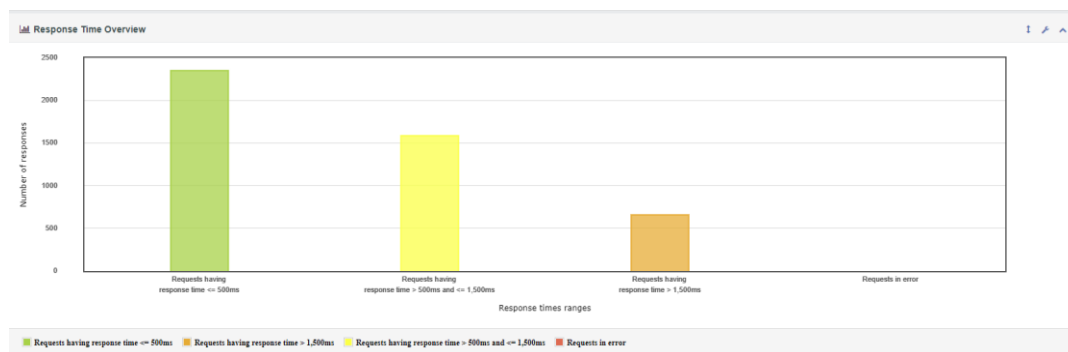
**Sl. 8.20.** *Prikaz sažetka zahtjeva na web sustavu News s 10 korisnika koristeći BlazeMeter*

Nakon grafa slijedi isječak tabličnog prikaza koji se može vidjeti na slici 8.21, a također su prikazane samo početne vrijednosti jer nije stala cijela tablica.

Requests	Executions			Response Times (ms)						Throughput	Network (KB/sec)		
	Label	#Samples	KO	Error %	Average	Min	Max	90th pct	95th pct	99th pct	Transactions/s	Received	Sent
Total		4595	0	0.00%	940.68	2	44994	1852.90	2616.60	5707.20	41.70	18993.21	47.56
http://127.0.0.1:5000/all	1	0	0	0.00%	38325.00	38325	38325	38325.00	38325.00	38325.00	0.03	1453.02	3.96
http://127.0.0.1:5000/all-0	1	0	0	0.00%	6380.00	6380	6380	6380.00	6380.00	6380.00	0.16	35.18	0.07
http://127.0.0.1:5000/all-1	1	0	0	0.00%	914.00	914	914	914.00	914.00	914.00	1.09	0.23	0.62
http://127.0.0.1:5000/all-2	1	0	0	0.00%	913.00	913	913	913.00	913.00	913.00	1.10	445.31	0.55
http://127.0.0.1:5000/all-3	1	0	0	0.00%	169.00	169	169	169.00	169.00	169.00	5.92	479.87	2.98
http://127.0.0.1:5000/all-4	1	0	0	0.00%	1005.00	1005	1005	1005.00	1005.00	1005.00	1.00	423.73	0.50
http://127.0.0.1:5000/all-5	1	0	0	0.00%	629.00	629	629	629.00	629.00	629.00	1.59	412.27	0.80
http://127.0.0.1:5000/all-6	1	0	0	0.00%	360.00	360	360	360.00	360.00	360.00	2.78	419.23	1.40
http://127.0.0.1:5000/all-7	1	0	0	0.00%	1016.00	1016	1016	1016.00	1016.00	1016.00	0.98	584.86	0.50
http://127.0.0.1:5000/all-8	1	0	0	0.00%	1045.00	1045	1045	1045.00	1045.00	1045.00	0.96	514.03	0.49
http://127.0.0.1:5000/all-9	1	0	0	0.00%	390.00	390	390	390.00	390.00	390.00	2.56	836.78	1.29
http://127.0.0.1:5000/all-10	1	0	0	0.00%	389.00	389	389	389.00	389.00	389.00	2.57	406.64	1.29
http://127.0.0.1:5000/all-11	1	0	0	0.00%	380.00	380	380	380.00	380.00	380.00	2.63	562.60	1.32
http://127.0.0.1:5000/all-12	1	0	0	0.00%	527.00	527	527	527.00	527.00	527.00	1.90	270.31	0.95
http://127.0.0.1:5000/all-13	1	0	0	0.00%	1148.00	1148	1148	1148.00	1148.00	1148.00	0.87	559.08	0.44
http://127.0.0.1:5000/all-14	1	0	0	0.00%	1677.00	1677	1677	1677.00	1677.00	1677.00	0.60	571.04	0.30
http://127.0.0.1:5000/all-15	1	0	0	0.00%	341.00	341	341	341.00	341.00	341.00	2.93	460.22	1.47
http://127.0.0.1:5000/all-16	1	0	0	0.00%	1051.00	1051	1051	1051.00	1051.00	1051.00	0.95	463.41	0.48
http://127.0.0.1:5000/all-17	1	0	0	0.00%	327.00	327	327	327.00	327.00	327.00	3.06	655.17	1.54
http://127.0.0.1:5000/all-18	1	0	0	0.00%	2348.00	2348	2348	2348.00	2348.00	2348.00	0.43	528.58	0.21
http://127.0.0.1:5000/all-19	1	0	0	0.00%	162.00	162	162	162.00	162.00	162.00	6.17	509.01	3.10

**Sl. 8.21.** *Isječak tabličnog prikaza dobivenih rezultata s 10 korisnika koristeći BlazeMeter*

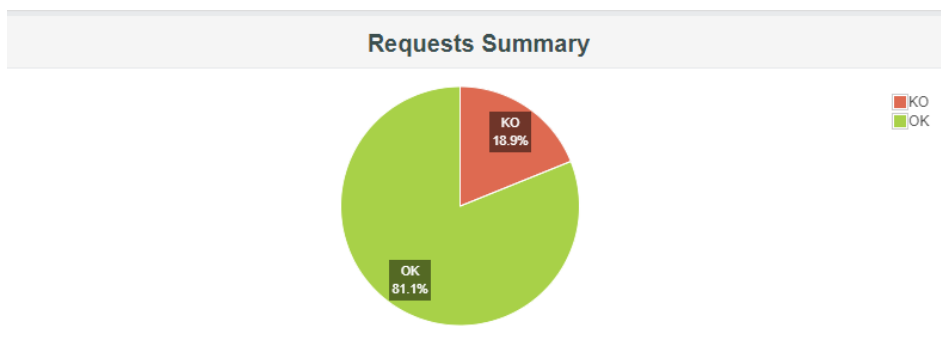
Nakon tabličnog prikaza potrebno je još provjeriti vrijeme odziva dok je 10 korisnika na web sustavu prikaza vijesti, a to vrijeme se može iščitati sa slike 8.22 na kojoj se nalazi grafički prikaz vrijeme odziva.



**Sl. 8.22.** *Vrijeme odziva s 10 korisnika koristeći BlazeMeter*

Za 2351 zahtjev vrijeme odziva je ispod 500 ms, a za 1586 zahtjeva vrijeme odziva je između 500 ms i 1.500 ms i samo 658 zahtjeva ima vrijeme odziva iznad 1.500 ms što znači da je stranica brza i kada ima 10 korisnika na njoj te da će korisnici biti zadovoljni njenom brzinom i radom jer neće štekati niti će korisnik morati čekati.

Nakon provjere s 10 korisnika, slijedi provjera s 80 korisnika, kao što se može vidjeti na slici 8.23 počinju se javljati problemi tj. pojavljuju se zahtjevi s pogreškom.



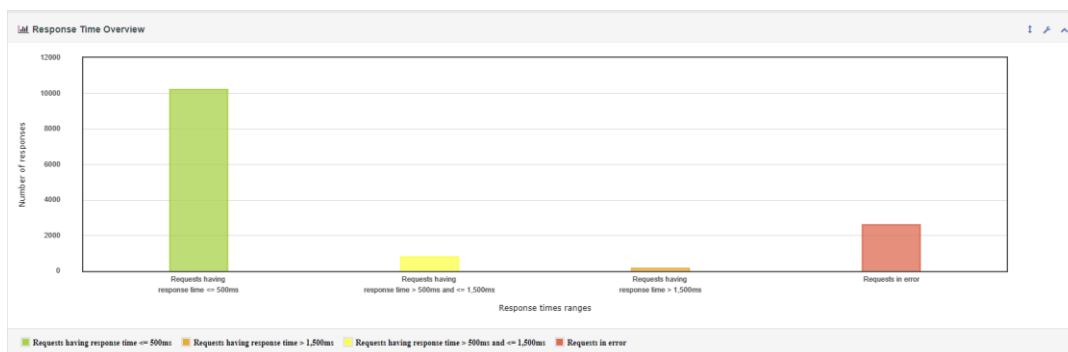
**Sl. 8.23.** Prikaz sažetka zahtjeva na web sustavu News s 80 korisnika koristeći BlazeMeter

Iz grafičkog prikaza vidljivo je kako 81.1% zahtjeva je dobro, dok je 18.9% s pogreškom. Može se zaključiti da je web sustav prikaz vijest News opterećen već s 80 korisnika.

Na slici 8.24 nalazi se isječak tabličnog prikaza tj. samo njegov početak zbog veličine podataka, a na slici 8.25 nalazi se grafički prikaz vrijeme odziva web sustava dok se na njemu nalazi 80 korisnika.

Executions			Response Times (ms)							Throughput	Network (KB/sec)	
#Samples	KO	Error %	Average	Min	Max	Median	90th pct	95th pct	99th pct	Transactions/s	Received	Sent
13760	2600	18.90%	480.14	0	9326	189.00	2019.00	2029.00	2058.00	52.55	2429.02	61.15
80	0	0.00%	324.43	227	2549	265.00	408.30	553.95	2549.00	1.36	2.31	0.92
5	0	0.00%	139.00	101	237	120.00	237.00	237.00	237.00	0.10	1.00	0.05
80	3	3.75%	596.01	0	2056	520.50	1073.50	1419.05	2056.00	1.35	122.10	0.61
80	3	3.75%	96.49	0	715	54.00	194.30	325.20	715.00	1.36	12.78	0.79
80	3	3.75%	108.20	0	669	91.00	198.20	413.15	669.00	1.36	7.04	0.78
80	3	3.75%	186.96	0	1069	151.00	279.30	505.90	1069.00	1.36	1.02	0.65
80	3	3.75%	191.70	0	2149	40.50	628.70	906.00	2149.00	1.36	18.23	0.84
80	3	3.75%	187.05	0	1451	40.50	605.70	1093.45	1451.00	1.36	20.36	0.84
80	3	3.75%	195.84	0	1472	37.00	771.30	933.55	1472.00	1.36	10.65	0.83
80	3	3.75%	172.23	0	1196	35.00	715.30	856.40	1196.00	1.36	14.44	0.84
80	3	3.75%	163.54	0	2030	35.00	626.50	798.15	2030.00	1.36	23.37	0.84
80	3	3.75%	168.21	0	2239	36.50	631.40	812.95	2239.00	1.36	14.89	0.84
80	3	3.75%	293.71	0	4032	40.50	749.00	2151.95	4032.00	1.36	9.68	0.84
80	2	2.50%	200.49	0	2428	40.50	431.80	1351.55	2428.00	1.36	13.05	0.85

**Sl. 8.24.** Isječak tabličnog prikaza dobivenih rezultata s 80 korisnika koristeći BlazeMeter

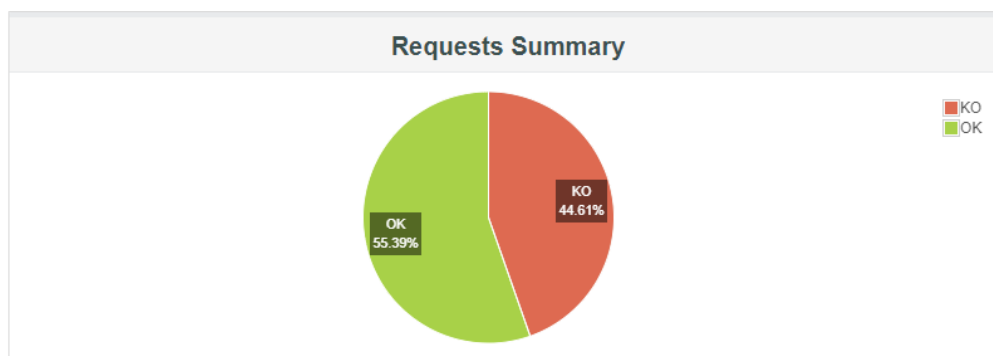




**Sl. 8.25.** *Vrijeme odziva s 80 korisnika koristeći BlazeMeter*

Iščitani podaci iz grafa su: 10219 zahtjeva ima vrijeme odziva ispod 500 ms, njih 790 ima vrijeme odziva između 500 ms i 1.500 ms, a 151 zahtjev ima vrijeme odziva iznad 1.500 ms dok 2600 zahtjeva je s pogreškom. Iz ovih podataka može se zaključiti da će se manjini događati pogreške i neće moći dohvatiti sve potrebne podatke ako se na web sustavu u isto vrijeme nalazi oko 80 korisnika. Web sustav nije jako opterećen, ali su vidljive naznake početka opterećenja.

Nakon provjere opterećenja sustava s 80 korisnika, ostaje još provjera opterećenosti s 400 korisnika pa na slici 8.26 može se vidjeti grafički prikaz opterećenosti sustava.



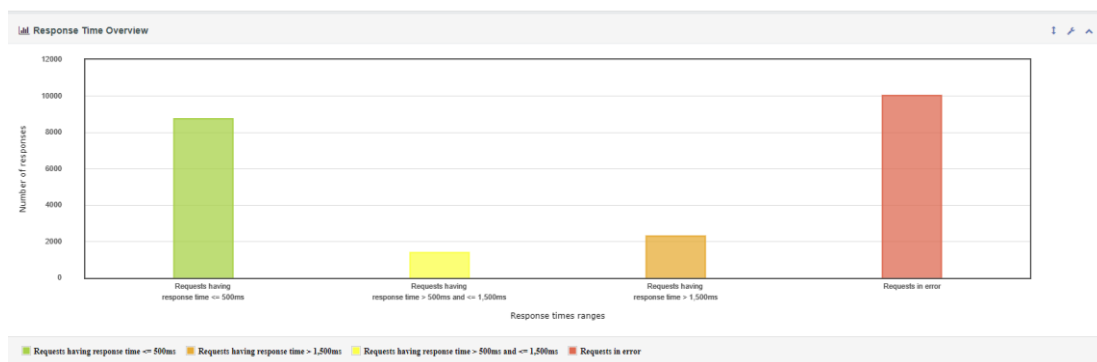
**Sl. 8.26.** *Prikaz sažetka zahtjeva na web sustavu News s 400 korisnika koristeći BlazeMeter*

Iz grafičkog prikaza vidljivo je da 55.39% zahtjeva je dobro obrađeno, dok 44.61% zahtjeva nije dobro obrađeno, detaljnije vrijednosti mogu se iščitati iz isječka tablice sa slike 8.27.

Executions			Response Times (ms)							Throughput	Network (KB/sec)	
#Samples	KO	Error %	Average	Min	Max	Median	90th pct	95th pct	99th pct	Transactions/s	Received	Sent
22475	10027	44.61%	4560.26	0	221363	406.50	8142.20	22748.65	101602.99	51.63	2562.08	44.10
800	800	100.00%	2041.26	2002	3129	2040.00	2054.00	2058.00	2060.00	5.12	13.52	0.00
1438	1438	100.00%	2473.50	2005	52773	2041.00	2666.00	3131.00	12751.91	4.48	11.83	0.00
3898	3898	100.00%	2255.90	2003	52776	2042.00	2059.00	2648.05	6423.02	12.99	34.32	0.00
1092	1092	100.00%	2510.28	2006	51582	2041.00	2691.20	3211.85	9586.73	4.19	11.07	0.00
40	0	0.00%	917.48	189	17689	230.50	736.50	4217.10	17689.00	0.52	0.22	0.44
26	0	0.00%	2740.85	221	36288	344.00	10769.80	27872.60	36288.00	0.29	0.27	0.18
26	0	0.00%	1662.12	178	12605	295.00	9223.50	12035.20	12605.00	0.39	0.20	0.11
26	0	0.00%	1078.65	43	23683	58.00	1387.40	15977.05	23683.00	0.29	0.12	0.10
78	0	0.00%	180.53	39	885	157.00	496.60	637.90	885.00	2.37	1.03	9.66
46	1	2.17%	1322.63	231	43072	301.50	682.40	1744.35	43072.00	0.42	0.43	0.93
50	30	60.00%	817.36	264	12243	312.50	1055.70	5239.25	12243.00	0.83	0.66	0.46
42	1	2.38%	829.86	156	12485	216.00	1503.40	7345.25	12485.00	0.79	0.16	1.31
58	58	100.00%	2820.50	152	53959	209.50	11878.10	21014.45	53959.00	0.30	0.81	0.17
971	0	0.00%	538.06	176	69510	197.00	641.80	1520.80	7054.92	5.17	0.44	6.00
203	0	0.00%	632.08	176	21600	193.00	734.00	1464.60	11299.24	1.56	0.13	1.68
52	0	0.00%	1416.29	726	18863	770.50	1297.50	5911.70	18863.00	0.78	0.07	0.66
43	0	0.00%	253.07	88	2141	125.00	285.00	1786.20	2141.00	1.01	1.12	0.99
43	0	0.00%	305.35	85	6971	116.00	288.40	619.40	6971.00	0.87	0.96	0.87
43	0	0.00%	438.28	89	10622	132.00	376.40	1300.40	10622.00	0.72	0.81	0.73
42	0	0.00%	166.05	80	861	120.00	282.00	459.05	861.00	1.03	1.14	1.05
43	0	0.00%	514.56	219	4805	295.00	898.20	2517.80	4805.00	1.04	1.66	0.72
49	49	100.00%	793.22	159	17829	200.00	1459.00	3088.00	17829.00	0.79	2.10	0.68
184	184	100.00%	130.32	45	2842	66.00	140.00	409.50	2213.85	3.88	10.15	3.39

Sl. 8.27. Isječak Tabličnog prikaza dobivenih rezultata s 400 korisnika koristeći BlazeMeter

Nakon tabličnih vrijednosti, potrebno je provjeriti vrijeme odziva web sustava tijekom 400 korisnika, prikaz na slici 8.28.



Sl. 8.28. Vrijeme odziva s 400 korisnika koristeći BlazeMeter

Za 8750 zahtjeva vrijeme odziva je ispod 500 ms, a za 1395 zahtjeva vrijeme odziva je između 500 ms i 1500 ms, dok je za 2303 zahtjeva vrijeme odziva iznad 1500 ms i 10027 zahtjeva je s pogreškom. S ovim podacima može se zaključiti da se web sustav neće dobro ponašati ako se na

njoj nalazi 400 korisnika. Vrijeme odziva je veliko i dugo će se čekati na obradu zahtjeva ili će se odmah dobiti pogreška. Može se zaključiti da web sustav ne bi podnio 400 korisnika odjednom te da je sustav dobro opterećen. U sljedećoj tablici 8.2 prikazani su dobiveni rezultati za testne skripte tijekom provođenja testa opterećenja.

**Tab. 8.2.** *Dobiveni rezultati s ekstenzijom BlazeMeter*

Broj korisnika	1	10	80	400
Testni scenariji				
<b>Odabir jednog portala</b>	7.4 sec	10.2 sec	8.9 sec	14.6 sec
<b>Otvaranje više portala i pretraživanje</b>	9.1 sec	18.1 sec	37.2 sec	31.2 sec
<b>Odabir opcije prikaza svih portala (ALL)</b>	7.3 sec	5.3 sec	8.8 sec	22.5 sec
<b>Odabir tri vijesti pa opciju svih portala (ALL)</b>	7.5 sec	8.4 sec	9.2 sec	21.7 sec
<b>Odabir tri vijesti pa opciju svih portala (ALL)</b>	6 sec	5.3 sec	8.7 sec	90.1 sec
<b>UKUPNO</b>	31.3 sec	47.3 sec	72.8 sec	180.1 sec

Usporedbom dobivenih rezultata s očekivanim rezultatima, vidi se da su rezultati vrlo blizu očekivanim, ima manjih odstupanja od očekivane vrijednosti, a na odstupanje utječe poslužitelj, brzina rada korisnika, obrada zahtjeva, itd. Također iz ovih rezultata može se zaključiti da se web sustav prikaz vijesti News dobro ponaša pod opterećenjem ako je manje od 400 korisnika u isto vrijeme na njemu.

Ako se uspoređuju samo rezultati dobiveni u tablici, vidi se kako je vrijeme izvođenja s 400 korisnika puno veće od vremena izvođenja s 1 korisnikom pa se može zaključiti da je sustav s 400 korisnika preopterećen i neće dobro funkcionirati u takvom okruženju. U slučaju kad je na web sustavu 10 korisnika, rezultati su skoro pa podjednaki s onima kada je na web sustavu samo 1 korisnik, a može se vidjeti kako problemi nastaju kada je na stranici 80 korisnika u isto vrijeme, sustav je tad opterećen, ali nije preopterećen.

### 8.3. Rezultati dobiveni u programskom alatu Visual Studio

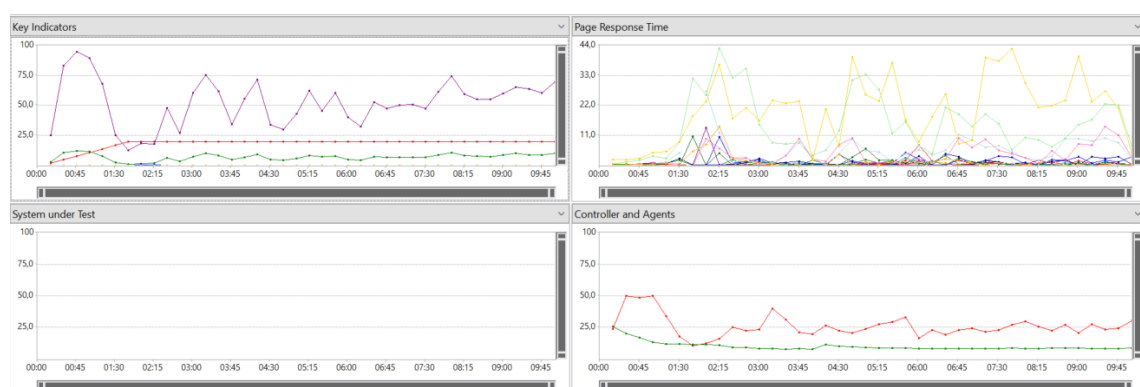
Visual Studio ne nudi izvještaje za izradu kao mogućnost sa JMeter-om , u Visual Studiju izrađuju se grafovima odmah pri provedbi testa opterećenja te se završetkom kreira tablica kao na slici 8.29.

Load Test Summary	
<b>Test Run Information</b>	
Load test name	LoadTest1
Description	
Start time	12.9.2020. 22:32:01
End time	12.9.2020. 22:42:01
Warm-up duration	00:00:00
Duration	00:10:00
Controller	Local run
Number of agents	1
Run settings used	Run Settings1
<b>Overall Results</b>	
Max User Load	200
Tests/Sec	2,61
Tests Failed	1.566
Avg. Test Time (sec)	42,4
Transactions/Sec	0
Avg. Transaction Time (sec)	0
Pages/Sec	72,0
Avg. Page Time (sec)	2,00
Requests/Sec	242
Requests Failed	31.633
Requests Cached Percentage	43,8
Avg. Response Time (sec)	1,44
Avg. Content Length (bytes)	9.567

Sl. 8.29. Tablični prikaz sažetka testa opterećenja u Visual Studiju

Iz tablice se može iščitati vrijeme izvođenja, trajanje izvođenja, broj maksimalnih korisnika na stranici, broja izvođenih testova po sekundi, testova koji su pali, prosječnog vremena, broja zahtjeva po sekundi, broj zahtjeva s pogreškom, vrijeme odziva, itd.

Na slici 8.30 nalazi se grafički prikaz rezultata provedenih u Visual Studiju te će se pomoću njih može zaključiti kako se stranica ponaša pod opterećenjem.



Sl. 8.30. Prikaz grafova testa opterećenja u Visual Studiju

Na x osi grafova nalaze se vrijeme izvođenja cijelog testa opterećenja u Visual Studiju, dok na y osima su različite vrijednosti, ovisne o grafu na kojem se gledaju. Graf s ključnim pokazateljima

sadrži broj pogrešaka po sekundi koji su označeni ljubičastom bojom i vidljivo je kako vremena variraju, najviši domet dostiže odmah na početku testa, a zatim naglo opada, crvena boja označava opterećenje korisnika, a zelena stranice po sekundi. Na grafu vrijeme odziva, vidljivo je jako puno podataka koje je teško iščitati. Podatci na grafu vrijeme odziva označavaju API Metode sa stranice, API metoda označena zelenom bojom koja ide do dometa od 44 sekunde je dohvaćanje (GET) početne stranice. Ako je potrebno početnoj stranici da se učitava 44 sekundi tada korisnik odustaje od pregleda takve stranice i više se na nju ne vraća. Žutom bojom označena je API metoda POST kojom se u tražilicu upisuje riječ te se pretražuje, kako vrijeme testa ide i pojavljuje se što više korisnika tako raste i vrijeme odziva za API metodu POST.

Treći graf sistema i testa neće biti objašnjen jer na njemu nema podataka, a graf kontrolera i agenta prikazuje postotak procesora koji označen crvenom bojom i dostupnost memorije tj. potrebnih resursa za provođenje testa opterećenja.

Obradom rezultata može se zaključiti da će web sustav prikazati vijesti raditi u ovim uvjetima jer od 242 poslana zahtjeva, 31 zahtjev je s pogreškom, sustav je dobro opterećen i dobiveni su dobri rezultati.

Nakon provedbe Testa opterećenja u svim navedenim alatima, može se zaključiti da diplomski rad „Programsko rješenje web sustava za prilagodljivi višekriterijski prikaz vijesti iz različitih izvora“ tj. web sustav News dobro radi tj. da su njegove funkcionalnosti dobre i da će izdržati više korisnika u isto vrijeme tj. maksimalni broj korisnika koji može podnijeti je 400, s tim da će nekima stvarati problemi ako je na stranici 80 korisnika u isto vrijeme tj. stranica će biti nešto sporija.

Poboljšanje web sustava News može se učiniti ispravljanjem prijavljenih pogrešaka, dodavanjem novih funkcionalnosti tj. iteracija razvoja, postavljanjem web sustava na poslužitelj koji može izdržati preko 500 korisnika u isto vrijeme.

#### **8.4. Usporedba korištenih alata**

U ovom potpoglavlju prikazat će se usporedba korištenih alata tj. zaključuju se kojim alatom se najbolje i najlakše došlo do približno točnih rezultata i koji alat je bio najbolji za korištenje. U tablici 8.3. nalaze se podaci usporedbe alata po koracima provedbe Testa opterećenja.

Po podacima iz tablice se može zaključiti da je korištenjem alata JMeter i ekstenzije BlazeMeter najlakše provesti Test opterećenja te se dobiju najtočniji rezultati tj. rezultati približni očekivanim.

**Tab. 8.3.** *Usporedba korištenih alata za provedbu Testa opterećenja*

<b>Koraci</b>	<b>JMeter i Proxy poslužitelj</b>	<b>JMeter i ekstenzija BlazeMeter</b>	<b>Visual Studio</b>
<b>Priprema</b>	Testni scenarija, broj korisnika, očekivani rezultati	Testni scenarija, broj korisnika, očekivani rezultati	Testni scenarija, broj korisnika, očekivani rezultati
<b>Snimalica</b>	Putem Proxy poslužitelja	Putem Chrome-ove ekstenzije BlazeMeter	Putem Internet Explorera koji se danas više ne koristi
<b>API metode</b>	Ne dohvaća sve API metode ili dohvati nasumične API metode	Dohvaća sve API metode sa stranice i daje mogućnost detaljnog pregleda	Dohvaća sve API metode sa stranice, ali ne da mogućnost detaljnog pregleda
<b>Postavljanje parametra</b>	Postavljanje broja korisnika i vremena izvođenja	Postavljanje broja korisnika i vremena izvođenja	Postavljanje testa, brzina, preglednika, broj korisnika, vremena izvođenja, itd.
<b>Rezultati</b>	Nisu dobri niti podjednaki očekivanim rezultatima te, ali ih je lagano analizirati	Rezultati su približno točni tj. podjednaki su očekivanim rezultatima te ih je lagano analizirati	Rezultate je teško iščitati i analizirati kako bi se usporedili s očekivanim
<b>Izvještaj</b>	Nudi html izvještaj	Nudi html izvještaj	Grafički ili tablični prikaz koji se ne može izvesti van aplikacije

## 9. ZAKLJUČAK

U diplomskom radu provedeno je testiranje web aplikacije izrađene u sklopu diplomskog rada „Programsko rješenje web sustava za prilagodljivi višekriterijski prikaz vijesti iz različitih izvora“. Provedene su dvije vrste testova na dva različita načina. Prvi način je ručno testiranje pomoću kojeg je provedeno funkcionalno testiranje web sustava prikaza vijesti News, a drugi način je automatizirano testiranje pomoću kojeg je provedeno nefunkcionalno testiranje, odnosno test opterećenja. Prije svega bilo je potrebno istražiti što je testiranje i kako će se testiranje provoditi te u kojim alatima, kao i upoznavanje s web sustavom News, a zatim je bilo potrebno napraviti pripremu, provedbu i analizu web testiranja. Za provedbu funkcionalnog testiranja korišten je alat TestMonitor za raspis testnog plana, dok su za provedbu nefunkcionalnog testiranja, odnosno testa opterećenja korišteni alati JMeter i Proxy poslužitelj, JMeter i ekstenzija BlazeMeter, te Visual Studio.

Nakon detaljne pripreme koja čini temelj web testiranja, uslijedila je provedba testiranja kojom su dobiveni rezultati koji se razlikuju zbog provedbe u različitim testnim alatima. Rezultati funkcionalnog testiranja su izvrsni, nije pronađena pogreška koja bi mogla sprječavati rad web sustava News, već su potrebne manje izmjene za poboljšanje rada. Dobiveni rezultati provedbe testa opterećenja ukazuju na performanse web sustava News, odnosno koliko korisnika u isto vrijeme sustav može uspješno posluživati. Analizom dobivenih rezultata testa opterećenja može se zaključiti da sustav trenutno maksimalno može podnijeti 400 korisnika, s tim da će stranica tada sporo raditi te će neki korisnici dobiti pogreške. Međutim, ako se sustav postavi na poslužitelj s većim kapacitetom, tada će biti veća mogućnost posluživanja većeg broja korisnika u isto vrijeme bez pogreške u radu. Detaljnim provođenjem funkcionalnog i nefunkcionalnog testiranja može se zaključiti da je web testiranje na sustavima kao što je web sustav prikaza vijesti News neophodan te da web sustav prikaza vijesti News trenutno može podnijeti do 400 korisnika i da su njegove funkcionalnosti ispravne.

## LITERATURA

- [1] F. Maras, Programsko rješenje web sustava za prilagodljivi višekriterijski prikaz vijesti iz različitih izvora, diplomski rad, Fakultet elektrotehnike, računarstva i informacijskih tehnologija, 2020.
- [2] Web Application Testing: 8 Step Guide to Website Testing, <https://www.guru99.com/web-application-testing.html> [pristupano 23. lipnja 2020.]
- [3] Functional testing, <https://test.io/functional-testing/> [pristupano 23. lipnja 2020.]
- [4] Microsoft, Quickstart: Create a load test project, <https://docs.microsoft.com/en-us/visualstudio/test/quickstart-create-a-load-test-project?view=vs-2019>, [pristupano 24. lipnja 2020.]
- [5] A Complete Guide to Performance Testing Types: Steps, Best Practices, Metrics and More, <https://dzone.com/articles/a-complete-guide-to-performance-testing-types-test> [pristupano 24. lipnja 2020.]
- [6] Z.M. Jiang, A. E. Hassan, Automatic Identification of Load Testing Problems, znanstveni rad, Software Analysis and Intelligence Lab (SAIL) Queen's University, studeni 2015.
- [7] A. Smith, Preparing for Load Testing Best Practices, Software Testing Magazin, travanj 2014.
- [8] Z. M. Jiang, Ahmed E. Hassan, A Survey on Load Testing of Large-Scale Software Systems, znanstveni rad, studeni 2015.
- [9] J. M. Islamia, R. K. M. Amjad, Performance testing (load) of web applications based on test case management, znanstveni rad, Department of Computer Engineering, travanj 2016.
- [10] M. Tudor, D. Martinović, Testiranje programske podrške, Pomorstvo, Vol. 19, 2005.
- [11] Metode i tehnike testiranja programske podrške, prezentacija, Zablude testiranja, FERIT, predavanja u ak. god. 2018/19.
- [12] Što je ručno ispitivanje?, <https://hr.itpedia.nl/2017/10/11/wat-is-handmatig-testen/>, [pristupano 23. lipnja 2020.]
- [13] Metode i tehnike testiranja programske podrške, prezentacija, Regresijsko testiranje FERIT, predavanja u ak. god. 2018/19.
- [14] J. Maltar, Automatizacija testiranja, <http://mrkve.etfos.hr/pred/ozm/si/sem12.pdf>, Osijek 2017. godine, str. 2/13



- [15] M. Fewster, D. Graham, Software Test Automation: Effective use of test execution tools, ACM Press Books, London, 1994.
- [16] I. Jukić, Automatsko testiranje programa, <https://zir.nsk.hr/islandora/object/foi:3888/preview>, Varaždin 2018. godine, str.39/100
- [17] Functional testing, <http://www.mstsolutions.com/blog/content/genericfunctional-testing-software-testingie=UTF-8> , blog [pristupano 24. lipnja 2020.]
- [18] A. Spillner, T. Linz i H.Schafer, Software testing foundations, Rocky Nook Inc., Santa Barbara, 2014.
- [19] Metode testiranja i tipovi testova, prezentacija 2019., str. 39-75.
- [20] Functional Testing Types – 25 Best Practices, Tips & More!, <https://www.qasymphony.com/blog/functional-testing-types/> [pristupano 28. lipnja 2020.]
- [21] Software Testing Fundamentals, Functional Testing, <http://softwaretestingfundamentals.com/functional-testing/>, [pristupano 28. lipnja 2020.]
- [22] BlazeMeter, <https://en.wikipedia.org/wiki/BlazeMeter>, [pristupano 28. lipnja 2020.]
- [23] JMeter, <https://jmeter.apache.org/index.html> , [pristupano 28. lipnja 2020.]
- [24] What is JMeter? Introduction & Uses, <https://www.guru99.com/introduction-to-jmeter.html> [pristupano 28. lipnja 2020.]
- [25] Proxy, <https://hr.wikipedia.org/wiki/Proxy> [pristupano 28. lipnja 2020.]
- [26] HTTP Proxy Server in JMeter: Record Example Script, <https://www.guru99.com/introduction-to-jmeter.html> [pristupano 28. lipnja 2020.]
- [27] BlazeMeter, <https://en.wikipedia.org/wiki/BlazeMeter>. [pristupano 29. lipnja 2020.]
- [28] Chrome web trgovina, BlazeMeter | The Continuous Testing Platform, <https://chrome.google.com/webstore/detail/blazemeter-thecontinuous/mbopgmdnpcbohpnfglhbfongabi> [pristupano 29. lipnja 2020.]
- [29] Microsoft Recommends JMeter and BlazeMeter as Preferred Load Testing Tools, <https://www.blazemeter.com/blog/microsoft-recommends-jmeter-and-blazemeter-as-preferred-load-testing-tools> [pristupano 29. lipnja 2020.]
- [30] Load Test, [https://www.google.com/search?q=visual+studio+load+test&rlz=1C1CHBF\\_enHR831HR831&sxsrf=ALeKk03tj4kG8qdd3Ry3HIXhpGA6dibQHQ:1594148590313&source=lnms&tbm=isch&sa=X&ved=2ahUKEwj9icej6rvqAhW1wsQBHVKuBHYQ\\_AU](https://www.google.com/search?q=visual+studio+load+test&rlz=1C1CHBF_enHR831HR831&sxsrf=ALeKk03tj4kG8qdd3Ry3HIXhpGA6dibQHQ:1594148590313&source=lnms&tbm=isch&sa=X&ved=2ahUKEwj9icej6rvqAhW1wsQBHVKuBHYQ_AU)

- <oAXoECAwQAw&biw=1280&bih=529#imgrc=AKEqISpc873yYM> [pristupano 29. lipnja 2020.]
- [31] Test Management Software. Simplified, <https://www.testmonitor.com/> [pristupano 13. rujna 2020.]
- [32] Web-Shop 0.1, aplikacija, FSB, [pristupano 13. rujna 2020.]
- [33] Testni slučaj i testni scenarij - 2020 – SOFTVER,  
<https://hr.betweenmates.com/difference-between-test-case-and-test-scenario-7369>,  
[pristupano 13. rujna 2020.]

## SAŽETAK

U ovom diplomskom radu objašnjena je metodologija, razine i vrste testiranja. U praktičnom dijelu rada prikazana priprema, provedba i analiza funkcionalnog testiranja i testiranja opterećenja na web aplikaciji za višekriterijski prikaz vijesti iz različitih izvora. Pripremom, provedbom i analizom rezultata funkcionalnog testiranja može se zaključiti kako su sve funkcionalnosti navedenog web sustava ispravne, te da su potrebne manje izmjene kako korisnika ne bi dovele do zabune. Pripremom provedbom i analizom rezultata testa opterećenja može se zaključiti da navedeni web sustav trenutno istovremeno podnosi do nekoliko stotina korisnika. Osim metodologije i provedbe testiranja, objašnjeni su korišteni alati kako bi se testovi mogli provesti i njihove mogućnosti te se uspoređuje njihov rad i analizira u kojem alatu je najbolje provesti funkcionalno i nefunkcionalno testiranje.

**Ključne riječi:** funkcionalno testiranje, nefunkcionalno testiranje, performanse sustava, programska podrška, testiranje opterećenja.

## **ABSTRACT**

This paper explains the methodology, levels and types of testing. In the practical part of the paper, the preparation, implementation and analysis of functional testing and load testing on the thesis "Software solution of the web system for adaptive multi-criteria display of news from different sources" is performed. By preparing, implementing and analyzing the results of functional testing, it can be concluded that all functionalities of the said web system have been corrected that minor changes are needed so that users should not make confusions, prepared evidence and analysis of load test results can be concluded users at the same time, if there are more than 400 users on the web system, then the system will be overloaded and will crash. In addition to the methodology and implementation of testing, the tools used to enable the tests to be conducted are explained and their capabilities compare their work and draw a conclusion in which tool is best to perform functional and non-functional testing.

**Keywords:** functional testing, non-functional testing, load testing, software, system performance

## **ŽIVOTOPIS**

Ines Maras (ex. Koić) rođena je 05.01.1996. u Osijeku. U Osijeku završava osnovnu školu „Grigor Vitez“, te 2010. upisuje Prvu gimnaziju u Osijeku. 2014. godine upisuje Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek, smjer računarstvo. Godine 2018. upisuje i uspješno završava „Span Academy“.

---

Ines Maras

## **PRILOZI**

Prilog 1: Diplomski rad u .pdf formatu.

Prilog 2: Diplomski rad u .docx formatu

Prilog 3: Poveznica spremljen u tekstualni dokument za pristup testnom planu funkcionalnog testiranja

Prilog 4: Testni plan rađen s poslužiteljem Proxy u .jmx formatu

Prilog 5: Testni plan rađen s ekstenzijom BlazeMeter u .jmx formatu

Prilog 6: Visual Studio projekt provedbe testa opterećenja .sln formatu

Prilog 7: Izvješća o obavljenom testiranju naziva „htmlReport.html“

Prilog 8: Izvješća o obavljenom testiranju naziva „htmlReportB.html“