



Spring, 2022-2023

SE2226 - Software Quality Assurance and Testing
Term Project Report

Project Name	Amazon Website Testing
Student Name	Pelin Sude Kırılı, Hande Hazan Keskin, Mert Kahraman
Student No	20070006023, 21070006014, 20070006020
Department	Software Engineering
Course Section No	1

Contents	Page
Introduction	2
Black-Box Testing Methods	2-6
White-Box Testing Method	7-8
Usability Testing	9
Performance Testing	9-10

Introduction:

Amazon is a website (<https://www.amazon.com.tr>) which is used for online shopping world-wide. It is found in 1994 and it is one of the biggest sites of the world. On the website, there are nearly 300 million members and more than 353 million products in multiple categories such as fashion, consumer electronics, home & furniture, grocery and lifestyle products.

In our project, We tested Amazon website in these categories:

1. Black-Box Testing
 - 1.1 Testing Report
 - 1.2 Decision Table
 - 1.3 Equivalence Table
 - 1.4 Boundary Analysis
 - 1.5 Use Case Testing
2. White-Box Technique
 - 2.1 Test for Java Selenium, ChromeDriver
3. Usability Testing
4. Performance Testing
 - 4.1 Load Testing

Testing:

1. Black-Box Testing:

1.1 Testing Report for Sign up Action:

-Checkbox is necessary

Name Surname	E-Mail Input	Password Input	2.Password	Checkbox	Output	Comment	Program error?
Testing Project	testing.se2226project@gmail.com	232014PMH	232014PMH	checked	Main Menu has opened	User signed up successfully	No
null	testing.se2226project@gmail.com	232014PMH	232014PMH	checked	Please enter your name and surname	Couldn't sign up	No
Testing Project	null	232014PMH	232014PMH	checked	Please enter e-mail	Couldn't sign up	No
Testing Project	testing.se2226project@gmail.com	null	null	Checked	Please enter password	Couldn't sign up	No
Testing Project	testing.se2226project@gmail.com	2320	2320	Checked	Password must be at least 6 characters	Couldn't sign up	No
Testing Project	testing.se2226project@gmail.com	232014PMH	2320	Checked	Passwords doesn't match	Couldn't sign up	No
Testing Project	testing.se2226project@gmail.com	232014PMH	232014PMH	Unchecked	Please check the box	Couldn't sign up	No
Testing Project	testingSe2226project@gmail.com	232014PMH	232014PMH	Checked	The email address is already taken	Couldn't sign up	No
Null	null	null	null	Unchecked	Please enter your name, please enter your email, Minimum 6 characters	All fields are empty	No

1.2 Decision Table for Log in Action:

Explanation: In this decision table we have two conditions which are: Email and Password. Therefore, we have $2^2 = 4$ different rules. T symbolizes true, F symbolizes false. The tick symbol shows us the action as known as output.

	R1	R2	R3	R4
CONDITIONS				
EMAIL	T	T	F	F
PASSWORD	T	F	T	F
ACTIONS				
HOME PAGE	✓			
SHOW ERROR MESSAGE		✓	✓	✓

1.3 Equivalence Table:

We decided to use equivalence partitioning on price filtering of Amazon's Dizüstü Bilgisayar Category. Our goal is to find valid prices according to price boundaries, to make our item search easier. Price is computer's price that placed in the given range. User click the range that he/she want to choose, and system displays the items.

Equivalence Class		Test Input	Expected Output
E1	5000<= price<= 7000	5399	1 Computer founded
E2	7000<price<=10000	8999	1 Computer founded
E3	10000<price<=15000	11999	9 Computer founded
E4	15000<price<=40000	20999	20 Computer founded
E5	40000<price	46000	4 Computer founded
U1	price<0	-3	Error message
U2	Not a valid price	“b”	Error message

Price filter category screen shot

Fiyat

- ☐ 5000 – 7000 TL
- ☐ 7000 – 10000 TL
- ☐ 10000 – 15000 TL
- ☐ 15000 – 40000 TL
- ☐ 40000 TL ve Üzeri

1.4 Boundary Analysis:

In this boundary analysis we have tested every boundary that takes place in our equivalence table. With boundary test inputs and the expected outputs of these inputs.

Equivalence Class		Test Input	Expected Output
E1	5000<= price<= 7000	4999	Couldn't Found
		5000	Couldn't Found
		5001	Couldn't Found
		6999	1 Computer founded
		7000	1 Computer founded
E2	7000<price<=10000	7001	1 Computer founded
		9999	1 Computer founded
		10000	1 Computer founded
E3	10000<price<=15000	10001	5 Computer founded
		14999	7 Computer founded
		15000	9 Computer founded
E4	15000<price<=40000	15001	16 Computer founded
		39999	18 Computer founded
		40000	8 Computer founded
E5	40000<price	40001	4 Computer founded
U1	price<0	-3	Error message
U2	Not a valid price	"b"	Error message

1.5 Use Case Testing:

Use Case Form

Use Case Name: Select and add item to the cart and delete it

Actor: Customer

Trigger: Customer wants to purchase items

Precondition: Customer must have an existing account

Normal Flow of Events:

1. Customer enters to the Amazon's sites login page.
2. Customer enters his/her e-mail.
3. Customer clicks to the "Devam et" button.
4. Customer enters his/her password.
5. Customer clicks on "Giriş Yap" button.
6. System verifies the e-mail and password and directs the customer to the home page.
7. Customer clicks on "Tümü" category.
8. Customer clicks on the displayed "Bilgisayar" button.
9. System directs the customer to the Bilgisayar page.
10. Customer chooses "Dizüstü Bilgisayarlar" button.
11. Customer chooses "Windows 11 Home" filter.
12. Customer sets a filter for price fields between 10000-32000.
13. Customer changes the filter to "Düşükten Yüksekçe".
14. System displays all the items ordered by ascending order.
15. Customer goes to second page.

16. Customer selects an item that he/she wants to purchase.
17. Customer goes over every photo of the product.
18. Customer clicks the “Sepete Ekle” button to add an item to his/her cart.
19. System displays a screen to the customer which has “Şimdi Al” and “Sepete Ekle” options.
20. Customer clicks on the option “Sepete Ekle”.
21. Customer clicks on “Sepete Git” button.
22. System displays the purchase page.
23. Customer clicks on the “Sil” button.
24. System displays a confirmation message.

Alternate Flows:

6A1: Customer enters incorrect e-mail to login.

1. System displays an error message about wrong password or e-mail.
2. Customer enters the correct e-mail.
3. Use case continues.

6A2: Customer enters incorrect password to login.

1. System displays an error message about wrong password or e-mail.
2. Customer enters the correct password.
3. Use case continues.

7A1: Customer clicks on the “Moda” category button.

1. System displays the “Moda” category page.
2. Use case continues.

12A1: Customer sets a filter for price fields between 100-1000.

1. System filters the prices for 100-1000.
2. System displays the page that informs the customer there are no product according to their filters.
3. Customer clicks on home page button.
4. Use case ends.

12A2: Customer sets a filter for Sabit Sürücü Kapasitesi between 200-499 GB.

1. System filters by the Sabit Sürücü Kapasitesi and displays the items.
2. Use case continues.

20A1: Customer clicks on the option “Şimdi Al”.

1. Customer enter his/her billing information.
2. System confirms the payment.
3. Purchase has been made.
4. Use case ends.

Test Cases:

	Scenario	Email	Password	Category	Filter	Item	Add to cart	Go to cart	Delete item	Go homepage	Expected output
1	Normal Flow/Customer selects an item and delete from the cart	V	V	Selects category	Filters item	Selects item	V	V	V	N/A	Customer successfully selects and deletes the item from the cart
2	Customer enters incorrect email	I	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	System shows error message that says email is wrong
3	Customer enters incorrect password	V	Invalid 1 time, then valid	N/A	N/A	N/A	N/A	N/A	N/A	N/A	System shows error message that says wrong password
4	Customer clicks on different category	V	V	Selects different category	N/A	N/A	N/A	N/A	N/A	N/A	Customer chooses different category to shop
5	Customer filters the price	V	V	Selects category	Different filter	Selects item	N/A	N/A	N/A	N/A	System shows that there are no products Customer can't add item to the cart returns to homepage
6	Customer selects different filter option	V	V	Selects category	Different filter	Selects item	N/A	N/A	N/A	N/A	System displays items according to the selected filter
7	Customer chooses to buy directly	V	V	Selects category	filters	Selects item	V	I	I	N/A	Customer buys the item directly

TC1, TC3, and TC5 are automated by Selenium Web driver.

Generated test cases:

Test Cases	Email	Password	Category	Filter	Item	Add to cart	Go to cart	Delete item	Go homepage	Expected output
TC1	testing.se2226project@gmail.com	232014PMH	Bilgisayarlar	Price filter (10000,32000), Operating system filter(Windows 11 home) Sorting filter (düşükten,yükseğe)	Casper Nirvana C500.1115-8V00T-G-F Intel Core i3	selected	selected	selected	N/A	Customer successfully selects and deletes the item from the cart
TC2	se2226project@gmail.com	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	System shows error message that says email is wrong
TC3	testing.se2226project@gmail.com	Wrongpassword , 232014PMH	N/A	N/A	N/A	N/A	N/A	N/A	N/A	System shows error message that says wrong password
TC4	testing.se2226project@gmail.com	232014PMH	Moda	N/A	N/A	N/A	N/A	N/A	N/A	Customer choose different category to shop
TC5	testing.se2226project@gmail.com	232014PMH	Bilgisayarlar	Price filter (100,1000), Operating system filter(Windows 11 home) Sorting filter (düşükten,yükseğe)	No item	N/A	N/A	N/A	N/A	System shows that there are no products Customer can't add item to the cart returns to homepage
TC6	testing.se2226project@gmail.com	232014PMH	Bilgisayarlar	Hard drive capacity filter(200-499GB)	N/A	N/A	N/A	N/A	N/A	System displays items according to the selected filter
TC7	testing.se2226project@gmail.com	232014PMH	Bilgisayarlar	Price filter (10000,32000), Operating system filter(Windows 11 home) And Sorting filter (düşükten,yükseğe)	Asus Zenbook 14 OLED (UX3402)	selected	Not selected	Not selected	N/A	Customer buys the item directly

2. White-Box Technique

2.1 Test for Java Selenium, ChromeDriver

We have used the Selenium library and ChromeDriver while automating out test cases. Scenarios that we have tested are given above the use case table TC1, TC3 and TC5. TC1 is the normal flow where customer logs in to the system successfully and after the necessary selections and filters, customer selects the computer, adds to the cart and then deletes from the cart. TC3 is an alternative flow where customer enters the incorrect password in the first try then in the second try enters it correctly then continues as the normal flow. TC5 is an alternative flow when customer selects different filtering options (price filter between 100 and 1000) and web application could not display any items according to the filters. We have started our automation in the login page of Amazon and we have continued according to the steps on the use case form. Here is the code of the normal flow, the rest of the code is in JavaSeleniumCode file.

```
public void normalFlow() throws InterruptedException {
    System.setProperty("webdriver.chrome.driver", "C:/Users/Mert/Selenium/ChromeDriver/chromedriver.exe");
    WebDriver driver=new ChromeDriver();
    driver.get("https://www.amazon.com.tr/ap/signin?openid.pape_max_auth_age=0&openid.return_to=https%3A%2F%2Fwww.amazon.com.tr%2F%3F%26tag%3Dtrt");

    driver.manage().window().maximize();

    Thread.sleep(2000);
    WebElement email = driver.findElement(By.xpath( xpathExpression: "//input[@type='email']"));
    email.sendKeys( ...keysToSend: "testing.se2226project@gmail.com");

    Thread.sleep(2000);
    WebElement login = driver.findElement(By.xpath( xpathExpression: "//input[@id='continue']"));
    login.click();

    Thread.sleep(2000);
    WebElement password = driver.findElement(By.xpath( xpathExpression: "//input[@id='ap_password']"));
    password.sendKeys( ...keysToSend: "232014PMH");

    Thread.sleep(2000);
    WebElement finallogin = driver.findElement(By.xpath( xpathExpression: "//input[@id='signInSubmit']"));
    finallogin.click();

    Thread.sleep(2000);
    WebElement tumu = driver.findElement(By.xpath( xpathExpression: "//i[@class='hm-icon nav-sprite']"));
    tumu.click();

    Thread.sleep(2000);
    WebElement comp = driver.findElement(By.xpath( xpathExpression: "//*[@id='hmenu-content']/ul[1]/li[10]/a"));
    comp.click();

    Thread.sleep(2000);
    WebElement laptop = driver.findElement(By.xpath( xpathExpression: "//*[@id='hmenu-content']/ul[5]/li[3]/a"));
    laptop.click();

    Thread.sleep(2000);
    WebElement systemFilter = driver.findElement(By.xpath( xpathExpression: "//*[@id='s-refinements']/div[4]/ul/li[1]/span/a/div/label/i"));
    systemFilter.click();

    Thread.sleep(2000);
}
```

```

Thread.sleep( millis: 2000);
WebElement lowPrice = driver.findElement(By.id("low-price"));
lowPrice.sendKeys( ...keysToSend: "10000");

Thread.sleep( millis: 2000);
WebElement topPrice = driver.findElement(By.id("high-price"));
topPrice.sendKeys( ...keysToSend: "32000");

Thread.sleep( millis: 2000);
WebElement git = driver.findElement(By.xpath( xpathExpression: "//input[@class='a-button-input']"));
git.click();

Thread.sleep( millis: 2000);
WebElement order = driver.findElement(By.xpath( xpathExpression: "//span[@class='a-dropdown-prompt']"));
order.click();

Thread.sleep( millis: 2000);
WebElement selectorder = driver.findElement(By.cssSelector("a[id='s-result-sort-select_1']"));
selectorder.click();

Thread.sleep( millis: 2000);
WebElement select_comp = driver.findElement(By.xpath( xpathExpression: " //span[@class='a-size-base-plus a-color-base a-text-normal']"));
select_comp.click();

Actions action = new Actions(driver);

List<WebElement> photoitems = driver.findElements(By.xpath( xpathExpression: "//*[@id='altImages']/ul/li"));

//loop is for the pictures
for (int i = 4; i < 8 ; i++) {
    action.moveToElement(photoitems.get(i)).perform();
    Thread.sleep( millis: 500);
}

Thread.sleep( millis: 2000);
WebElement cart = driver.findElement(By.xpath( xpathExpression: "//*[@id='submit.add-to-cart']"));
cart.click();

String ExpectedTitle="Sepete Eklendi";
String ActualTitle=driver.findElement(By.xpath( xpathExpression: "//*[@id='NATC_SMART_WAGON_CONF_MSG_SUCCESS']/span")).getText();

```

```

Assert.assertEquals(ExpectedTitle,ActualTitle);
System.out.println("title:" + ActualTitle);

Thread.sleep( millis: 2000);
WebElement goToCart = driver.findElement(By.xpath( xpathExpression: "//*[@id='sw-gtc']/span/a"));
goToCart.click();

Thread.sleep( millis: 2000);
WebElement delete = driver.findElement(By.xpath( xpathExpression: "//input[@value='Sil']"));
delete.click();

String ExpectedTitle2="Alışveriş Sepeti";
String ActualTitle2=driver.findElement(By.xpath( xpathExpression: "//*[@id='sc-active-cart']/div/div/div/h1")).getText();

Assert.assertEquals(ExpectedTitle2,ActualTitle2);
System.out.println("title:" + ActualTitle2);

```


(NOTE: At the end of the normal flow we have written assertion code for delete action of the code but in the web application “Amazon sepetiniz boş.” This text is shown but when we get the actual text it writes “Alışveriş sepeti” so the actual and expected text does not match so to be sure that we have deleted the item we write “Alışveriş sepeti” into the expected title too.)

```
[1684152549.782][WARNING]: This version of ChromeDriver has not been tested with Chrome version 113.
May 15, 2023 3:09:10 PM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
WARNING: Unable to find an exact match for CDP version 113, so returning the closest version found: 112
title:Sepete EkLendi
Exception in thread "main" org.junit.ComparisonFailure: Create breakpoint : expected:<A[mazon sepetiniz boş.]> but was:<A[alışveriş Sepeti]> <2 internal
    at tests.normalFlow(tests.java:104)
    at main.main(main.java:9)

Process finished with exit code 1
```

3. Usability Testing:

Website is tested for these browsers and seen that the website works correctly in these browsers:

- Google Chrome
- Mozilla Firefox
- Internet Explorer/Microsoft Edge

4. Performance Testing

Load Testing: We used Jmeter load test for Amazon Website. These screenshots are belong to 1000 number of samples.

Sample #	Start Time	Thread Name	Label	Sample...	Status	Bytes	Sent Bytes 1	Latency	Connect Time(ms)
1	17:50:12.895	Thread Group 1-5	HTTP Request	335	⚠	7551	119	335	158
2	17:50:12.910	Thread Group 1-7	HTTP Request	332	⚠	7551	119	332	145
3	17:50:12.942	Thread Group 1-11	HTTP Request	300	⚠	7551	119	300	141
4	17:50:12.895	Thread Group 1-6	HTTP Request	347	⚠	7551	119	347	158
5	17:50:12.982	Thread Group 1-13	HTTP Request	362	⚠	7551	119	362	250
6	17:50:13.242	Thread Group 1-7	HTTP Request	160	⚠	7551	119	160	0
7	17:50:12.842	Thread Group 1-1	HTTP Request	585	✅	145374	119	294	153
8	17:50:12.863	Thread Group 1-3	HTTP Request	633	✅	145350	119	348	147
9	17:50:13.092	Thread Group 1-26	HTTP Request	452	⚠	7551	119	443	213
10	17:50:13.043	Thread Group 1-21	HTTP Request	502	⚠	7551	119	501	262
11	17:50:13.072	Thread Group 1-24	HTTP Request	479	⚠	7551	119	479	238
12	17:50:13.174	Thread Group 1-34	HTTP Request	577	⚠	7551	119	577	279
13	17:50:12.922	Thread Group 1-9	HTTP Request	840	✅	145358	119	360	152
14	17:50:12.878	Thread Group 1-4	HTTP Request	884	✅	145374	119	404	175
15	17:50:12.863	Thread Group 1-2	HTTP Request	907	✅	151833	119	348	157
16	17:50:12.912	Thread Group 1-8	HTTP Request	858	✅	145374	119	354	151
17	17:50:12.933	Thread Group 1-10	HTTP Request	837	✅	151946	119	349	141
18	17:50:13.032	Thread Group 1-20	HTTP Request	783	✅	145366	119	321	181
19	17:50:13.243	Thread Group 1-41	HTTP Request	610	⚠	7551	119	610	268
20	17:50:12.992	Thread Group 1-16	HTTP Request	998	✅	145374	119	367	220
21	17:50:13.230	Thread Group 1-5	HTTP Request	782	✅	145374	119	161	0
22	17:50:12.973	Thread Group 1-14	HTTP Request	1111	✅	151811	119	418	206
23	17:50:13.293	Thread Group 1-46	HTTP Request	792	⚠	7551	119	791	377
24	17:50:13.005	Thread Group 1-17	HTTP Request	1085	✅	151946	119	397	208
25	17:50:12.983	Thread Group 1-15	HTTP Request	1130	✅	151839	119	376	196
26	17:50:13.332	Thread Group 1-50	HTTP Request	791	⚠	7551	119	791	388
27	17:50:13.342	Thread Group 1-51	HTTP Request	846	⚠	7551	119	840	421
28	17:50:13.770	Thread Group 1-10	HTTP Request	418	⚠	7551	119	418	0
29	17:50:12.954	Thread Group 1-12	HTTP Request	1246	✅	151818	119	411	258
30	17:50:13.427	Thread Group 1-1	HTTP Request	790	✅	151811	119	369	0

☒ Scroll automatically? ☐ Child samples?

No of Samples 1000Latest Sample: 646Average: 1104Deviation: 1200

Sample #	Start Time	Thread Name	Label	Sample...	Status	Bytes	Sent Bytes ↑	Latency	Connect Time(ms)
61	17:50:15.228	Thread Group 1-7	HTTP Request	420		7551	119	420	0
62	17:50:15.281	Thread Group 1-13	HTTP Request	422		7551	119	422	0
63	17:50:13.223	Thread Group 1-39	HTTP Request	2503		151834	119	573	235
64	17:50:13.202	Thread Group 1-37	HTTP Request	2532		151946	119	575	256
65	17:50:13.770	Thread Group 1-8	HTTP Request	1991		151833	119	463	0
66	17:50:13.212	Thread Group 1-38	HTTP Request	2549		151843	119	652	252
67	17:50:13.762	Thread Group 1-9	HTTP Request	2004		145366	119	443	0
68	17:50:13.232	Thread Group 1-40	HTTP Request	2610		151833	119	556	232
69	17:50:13.751	Thread Group 1-34	HTTP Request	2108		145374	119	466	0
70	17:50:13.324	Thread Group 1-49	HTTP Request	2535		151976	119	858	403
71	17:50:13.816	Thread Group 1-20	HTTP Request	2064		145374	119	520	0
72	17:50:13.163	Thread Group 1-33	HTTP Request	2722		145342	119	732	265
73	17:50:13.133	Thread Group 1-30	HTTP Request	2761		151970	119	467	189
74	17:50:13.062	Thread Group 1-23	HTTP Request	2832		151845	119	544	243
75	17:50:14.217	Thread Group 1-1	HTTP Request	1685		145374	119	784	0
76	17:50:13.545	Thread Group 1-21	HTTP Request	2374		145366	119	388	0
77	17:50:13.312	Thread Group 1-48	HTTP Request	2627		145374	119	807	375
78	17:50:13.302	Thread Group 1-47	HTTP Request	2664		145374	119	821	385
79	17:50:13.551	Thread Group 1-24	HTTP Request	2415		151809	119	368	0
80	17:50:14.012	Thread Group 1-5	HTTP Request	1954		145374	119	557	0
81	17:50:13.262	Thread Group 1-43	HTTP Request	2724		151826	119	851	367
82	17:50:15.533	Thread Group 1-26	HTTP Request	482		7551	119	481	0
83	17:50:15.545	Thread Group 1-29	HTTP Request	533		7551	119	528	0
84	17:50:13.362	Thread Group 1-53	HTTP Request	2782		151841	119	979	454
85	17:50:13.352	Thread Group 1-52	HTTP Request	2792		151841	119	989	471
86	17:50:13.274	Thread Group 1-44	HTTP Request	2908		145374	119	658	360
87	17:50:13.183	Thread Group 1-35	HTTP Request	3010		151970	119	587	262
88	17:50:13.373	Thread Group 1-54	HTTP Request	2906		145374	119	1053	587
89	17:50:15.648	Thread Group 1-7	HTTP Request	669		7551	119	669	0
90	17:50:13.392	Thread Group 1-56	HTTP Request	2930		145350	119	1158	610
<div><input checked="" type="checkbox"/> Scroll automatically? <input type="checkbox"/> Child samples?</div> <div>No of Samples: 1000Latest Sample: 646Average: 1264Deviation: 1200</div>									

Sample #	Start Time	Thread Name	Label	Sample...	Status	Bytes	Sent Bytes ↑	Latency	Connect Time(ms)
31	17:50:13.496	Thread Group 1-3	HTTP Request	772		145374	119	357	0
32	17:50:13.383	Thread Group 1-55	HTTP Request	1032		7551	119	1032	557
33	17:50:13.012	Thread Group 1-18	HTTP Request	1470		145366	119	440	231
34	17:50:13.990	Thread Group 1-16	HTTP Request	519		7551	119	511	0
35	17:50:13.022	Thread Group 1-19	HTTP Request	1522		145358	119	456	221
36	17:50:13.402	Thread Group 1-57	HTTP Request	1148		7551	119	1148	629
37	17:50:13.423	Thread Group 1-59	HTTP Request	1127		7551	119	1127	619
38	17:50:13.324	Thread Group 1-13	HTTP Request	1245		145374	119	287	0
39	17:50:14.085	Thread Group 1-46	HTTP Request	489		7551	119	489	0
40	17:50:13.453	Thread Group 1-62	HTTP Request	1121		7551	119	1121	638
41	17:50:13.463	Thread Group 1-63	HTTP Request	1111		7551	119	1111	628
42	17:50:13.053	Thread Group 1-22	HTTP Request	1659		145366	119	518	257
43	17:50:13.082	Thread Group 1-25	HTTP Request	2045		145351	119	529	228
44	17:50:13.612	Thread Group 1-78	HTTP Request	1556		7551	119	1556	841
45	17:50:13.683	Thread Group 1-85	HTTP Request	1505		7551	119	1505	815
46	17:50:14.509	Thread Group 1-16	HTTP Request	679		7551	119	679	0
47	17:50:13.672	Thread Group 1-84	HTTP Request	1516		7551	119	1504	827
48	17:50:13.402	Thread Group 1-7	HTTP Request	1826		145342	119	305	0
49	17:50:13.713	Thread Group 1-88	HTTP Request	1552		7551	119	1552	821
50	17:50:13.733	Thread Group 1-90	HTTP Request	1538		7551	119	1538	801
51	17:50:13.693	Thread Group 1-86	HTTP Request	1581		7551	119	1572	829
52	17:50:14.569	Thread Group 1-13	HTTP Request	712		7551	119	712	0
53	17:50:13.794	Thread Group 1-96	HTTP Request	1568		7551	119	1568	976
54	17:50:13.770	Thread Group 1-2	HTTP Request	1633		145374	119	485	0
55	17:50:13.762	Thread Group 1-4	HTTP Request	1653		151961	119	471	0
56	17:50:13.574	Thread Group 1-74	HTTP Request	1953		7551	119	1953	813
57	17:50:13.545	Thread Group 1-26	HTTP Request	1988		145374	119	374	0
58	17:50:13.122	Thread Group 1-29	HTTP Request	2423		145366	119	537	232
59	17:50:13.112	Thread Group 1-28	HTTP Request	2433		151810	119	547	242
60	17:50:14.268	Thread Group 1-3	HTTP Request	1297		145374	119	1246	0
<div><input checked="" type="checkbox"/> Scroll automatically? <input type="checkbox"/> Child samples?</div> <div>No of Samples: 1000Latest Sample: 646Average: 1264Deviation: 1200</div>									

Server Hostname: www.amazon.com.tr

Time taken for tests: 27.4 seconds

Completed request: 1000

Failed request: 345

Average bytes: 99221.7 bytes

Longest request: 15850

Shortest request: 113



Önemli Mesaj!

Hesabınızı daha iyi korumak için lütfen şifrenizi ve ardından aşağıdaki görselde bulunan karakterleri görüldüğü şekilde girin.

Giriş yap

testing.se2226project@gmail.com [Değiştir](#)

Şifre

[Şifremi unuttum](#)

Gördüğünüz karakterleri girin



[Yeni bir doğrulama görüntüsü deneyin](#)
[Doğrulama kodunu dinleyin](#)

Karakterleri girin

[Resmi görüntülemeye sorun mu yaşıyorsunuz?](#)

Giriş yap

☐ Oturumum açık kalsın. [Ayrıntılar](#) ▼

Additional note: if you see an screen like this enter the correct password once then run the code again this sometimes happens if the application stays open in someone's computer(password:232014PMH).