Software Quality Assurance & Testing Project

Department: Software Engineering

Lecturer: Doç. Dr. Korhan Karabulut

Prepared By:

Miray Yıldırım

İsmail Tekin

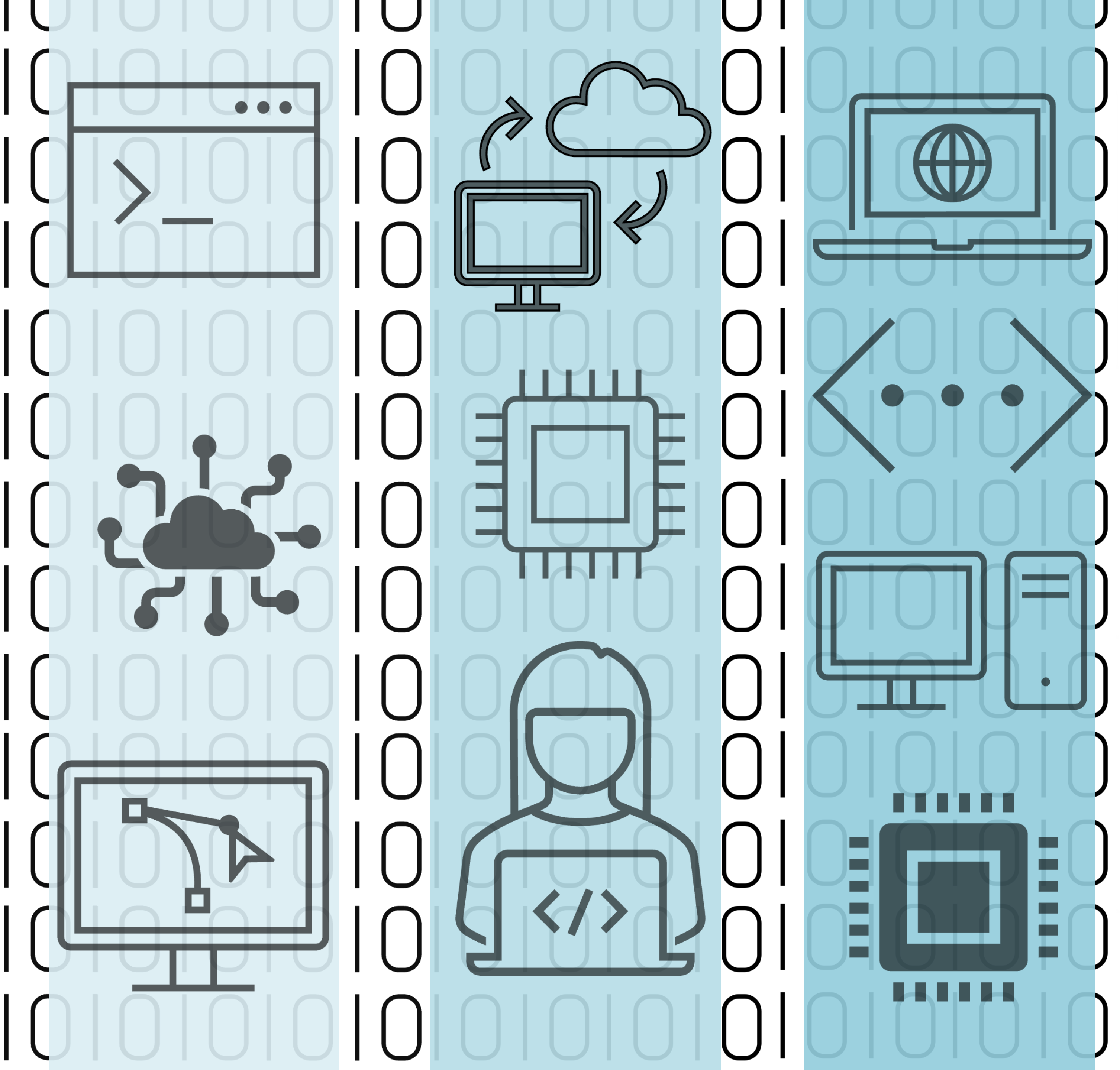
Dilara Kuyar

İlayda Gürbüzerol

https://technicalguftgu.in/wp-content/uploads/2024/07/Tests-in-Softwareentwicklung.webp

OVERVIEW

1. Introduction
2. Testing Strategy
3. Tools and Technologies
4. Test Execution
5. Test Results
6. Conclusion



https://i.teknolojioku.com/storage/files/images/2021/02/23/kitapyurdu-Q8o7.jpg

Kitapyurdu

Kitapyurdu is a major online bookstore that offers a wide range of books across various genres, including fiction, non-fiction, academic, and children's literature. In addition to books, it also sells stationery and educational materials. The platform provides user reviews, regular discounts, and a loyalty points system, making it a popular choice for readers looking for convenient and affordable book shopping online.

Testing Strategy

- ❖ Blackbox Testing
  - ➢ Equivalence Partitioning
  - ➢ Boundary Value Analysis
  - ➢ Use Case Testing
  - ➢ Decision Table Testing
  - ➢ Selenium Webdriver

- ❖ Whitebox Testing
  - ➢ Load Testing

https://www.webdevstory.com/wp-content/uploads/2024/01/software-development-word-cloud-1-750x375.webp

Tools And Technologies

Python was chosen as the main programming language for test automation due to its simplicity, readability, and vast ecosystem. It allows rapid development of test scripts and easy integration with automation tools like Selenium.

https://upload.wikimedia.org/wikipedia/commons/thumb/d/d5/Selenium_Logo.png/980pxSelenium_Logo.png

JUnit is a widely used testing framework in the Java ecosystem, ideal for unit testing and test-driven development. It supports annotations, assertions, and test runners, helping structure and execute tests efficiently.

https://upload.wikimedia.org/wikipedia/commons/5/59/JUnit_5_Banner.png

Selenium WebDriver is a powerful tool for automating web browsers. It enables testers to mimic user interactions with web pages and validate UI behavior, supporting multiple languages and cross-browser testing.

https://pngate.com/wp-content/uploads/2025/05/python-logo-blue-yellow-programming-language-icon-1.png

Equivalence Partitioning 1

Test Case ID: ECP-KY-02
Tested Area: Registration Page – Email Field
Equivalence Classes:
Valid: Proper email format (e.g., user@example.com)
Invalid: Missing “@” symbol, missing domain, empty input
Test Description:
The user enters “ilaydaexample.com” in the email field and
attempts to submit the registration form.
Input Value: “ilaydaexample.com”
Expected Result:
The system displays an error message such as “Please enter a valid
email address.” Form submission is blocked.

Equivalence Class	Description	Input Example	Expected Result
Valid	Proper email format	user@example.com	Form accepts the input and proceeds.
Invalid (No "@")	Missing "@" symbol	ilaydaexample.com	Error: "Please enter a valid email address." Submission blocked.
Invalid (No domain)	Missing domain (e.g., ".com")	test@	Error: "Please enter a valid email address." Submission blocked.
Invalid (Empty)	Empty input	""	Error: "This field is required." Submission blocked.

Equivalence Partitioning 2

Test Case ID: ECP-KY-05
Tested Feature: Product Listing – Category Filter Dropdown
Equivalence Classes:
Valid: Selecting an existing category (e.g., “Children’s Books”)
Invalid: Not selecting any category / selecting a disabled or blank option
Test Description:
The user selects the category “Children’s Books” from the filter dropdown menu.
Input Value: "Children’s Books"
Expected Result:
Only products from the “Children’s Books” category are listed.

Equivalence Class	Selection Type	Input Example	Expected Result
Valid	Existing category	"Children’s Books"	Only products from the selected category are displayed.
Invalid (No Selection)	Blank/Default option	(None selected)	Shows all products or prompts to select a category.
Invalid (Disabled)	Disabled/Non-selectable	(Grayed-out option)	Prevents selection; no filtering applied.

Boundary Value Analysis 1

	Equivalence Class	Test Input	Expected Output
E1	0 <= e <= 1	0	No result
		1	No result
E2	1 < e <= 100		
		2	Related Results
		99	Related Results
		100	Related Results
U1	e < 0	-5	Impossible
U2	e > 100	165	Error (uses only first 100 chars)

Boundary Value Analysis 2

	Equivalence Class	Test Input	Expected Output
E1	0 < e <= 958,40	1	19 result
		2	20 result
		958	492 result
		958,40	493 result
E2	958,40 < e <= 10^21-1	986	493 result
		10^21-2	493 result
		10^21-1	493 result
U1	e < 0	-5	Error (turns into positive)
U2	e >= 10^21	10^23	Error (changes number notation)

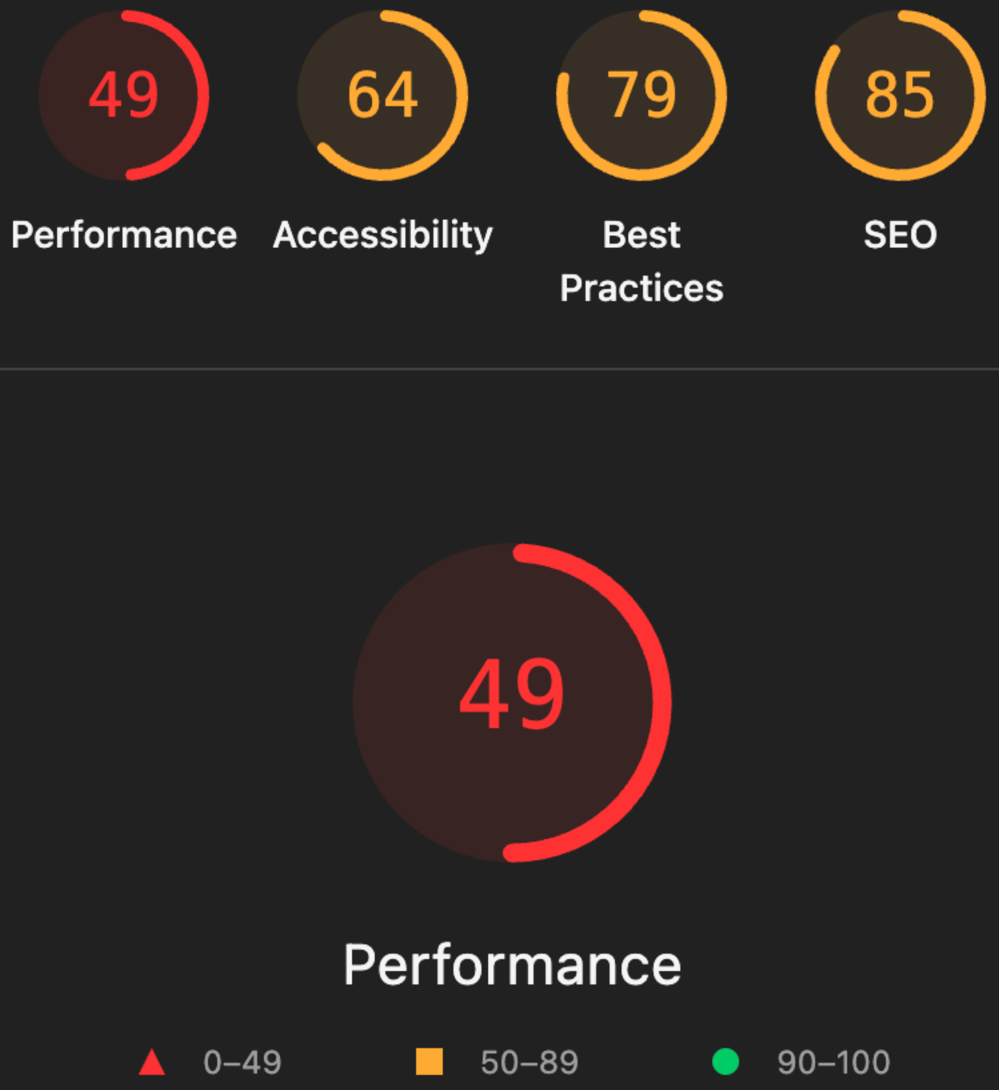
Use Case Testing

Use Case Scenario Name:	User Places an Order
Description:	The user logs into their account, adds a book to the cart, and places an order by entering address and payment information.
Actors:	User, System
Pre-Conditions:	• The user has an active internet connection. • The Kitapyurdu.com homepage is accessible. • The user has a valid Kitapyurdu account. • The user has logged in.
Main Scenario Steps:	1) The user searches for a book and selects a product from the search results. 2) The user clicks the “Add to Cart” button. 3) The system adds the product to the cart. 4) The system displays a message, such as “You have added the item to your cart.” in the top-left corner of the page. 5) The user opens the cart and clicks the “Checkout” button. 6) The user enters valid delivery address information. 7) The user selects Shipping Selection. 8) The user enters the continue button. 9) The user enters valid payment information. 10) The user clicks the “Confirm Order” button.
Post-Conditions:	The order is successfully created, and the user sees the order summary.

Decision Table Testing

Test Case ID	Username	Password	Expected Result	Description
DT01	Valid	Valid	Login Successful	Correct credentials provided
DT02	Valid	Invalid	”No match for E-Mail Address and/or Password.” message is displayed	Valid user but wrong password
DT03	Invalid	Valid	“No match for E-Mail Address and/or Password.” message is displayed	Username not registered in the system
DT04	Invalid	Invalid	“No match for E-Mail Address and/or Password.” message is displayed	Username not registered in the system

Google LightHouse Testing

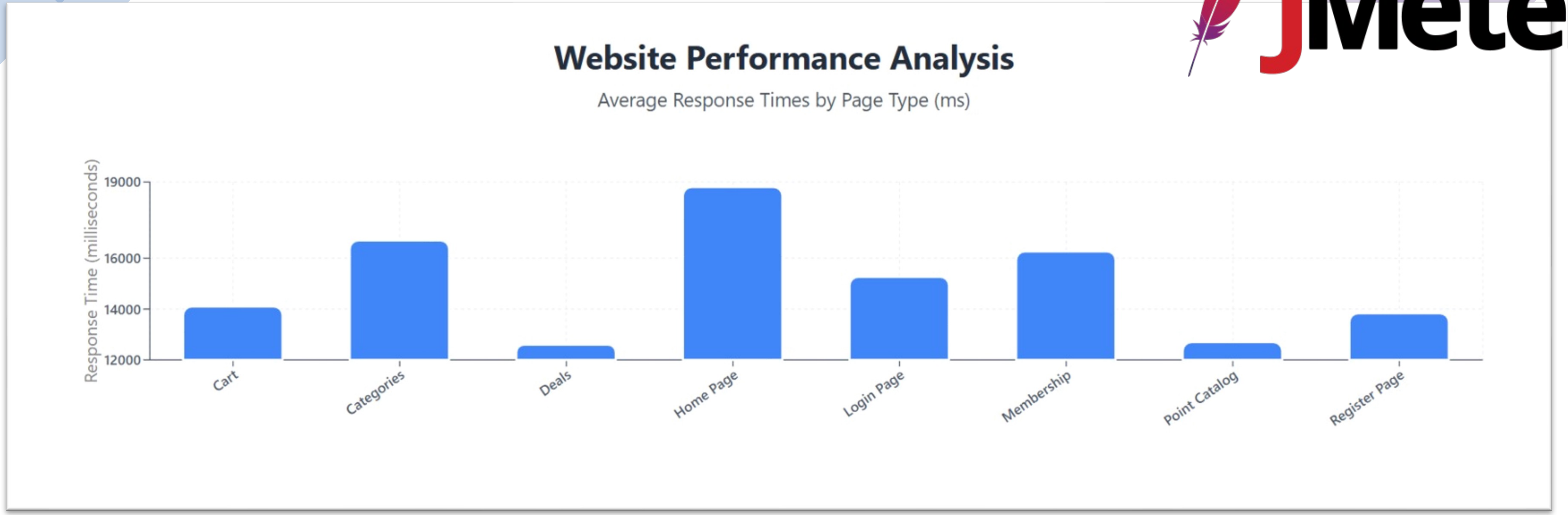


Analysis

Performance:
Kitapyurdu’s Lighthouse performance score is 49, which is considered poor. The site shows slow loading times, with First Contentful Paint at 3.8 seconds and Largest Contentful Paint at 6.8 seconds. These delays are caused by unoptimized large images, long JavaScript execution times, inefficient caching, and heavy main-thread work. Additionally, the Total Blocking Time of 490 ms negatively affects interactivity and user experience.

Accessibility:
The accessibility score is 64, indicating several issues that may hinder users with disabilities. Missing alt attributes on images, lack of labels on form elements, and links without discernible names reduce the usability for assistive technologies such as screen readers. Insufficient contrast between background and foreground colors also affects readability and overall accessibility.

Best Practices:
With a score of 79, the site meets most best practices but still has room for improvement. The use of 32 third-party cookies raises privacy and security concerns. Security headers like CSP, HSTS, and COOP are not fully implemented, and some JavaScript errors are detected. These issues could make the site more vulnerable to attacks such as clickjacking and reduce its overall reliability.



Performance Test Results

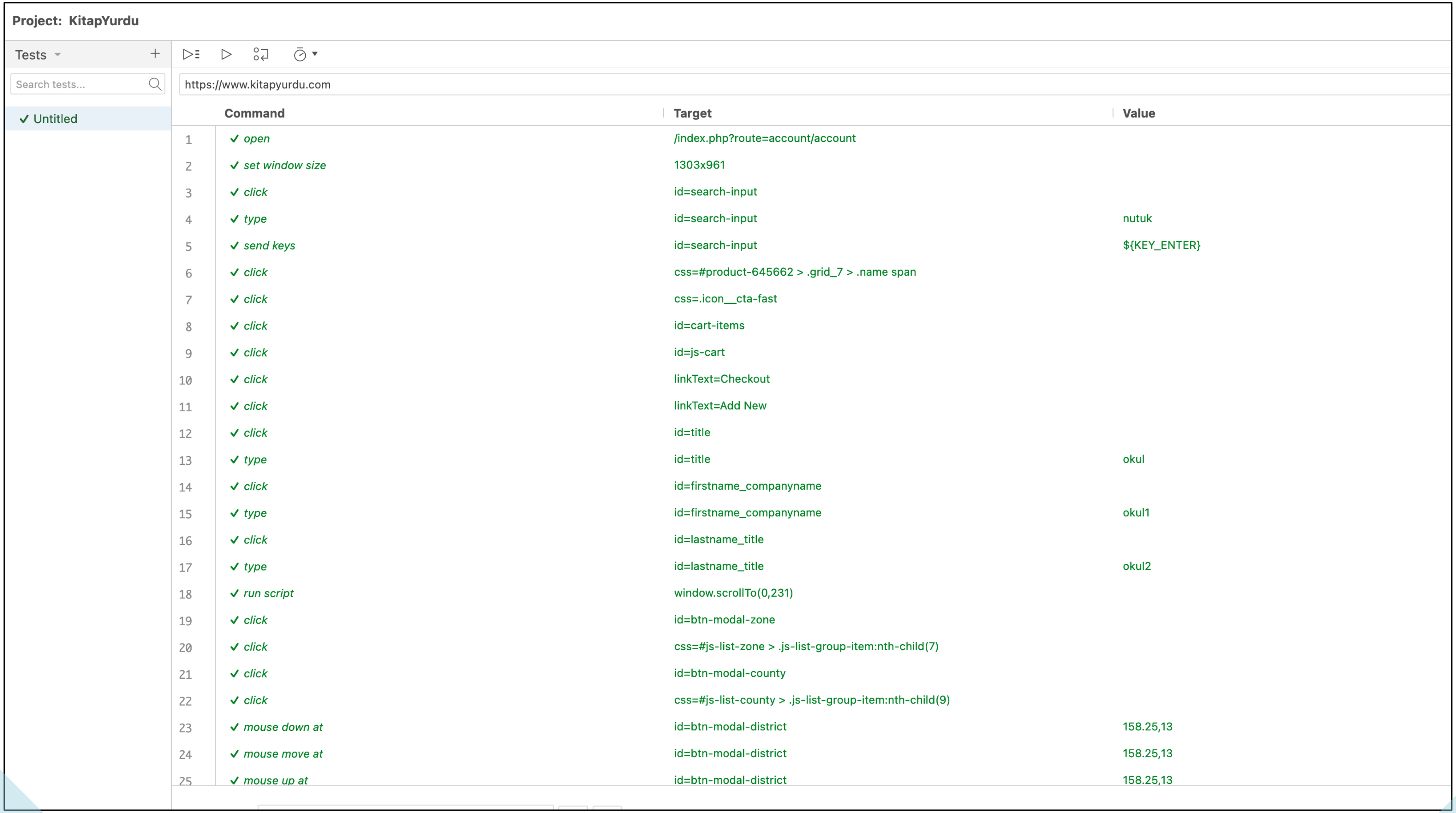
Chart: Shows the average response times (ms) for Kitapyurdu pages.

Finding: At 19 seconds, the **Home Page** is the slowest performing page. All page response times are at a level that could negatively impact the user experience.

Tool: The tests were conducted by generating a virtual user load with **Apache JMeter**.

https://upload.wikimedia.org/wikipedia/commons/2/22/Apache_JMeter.png?20160517202659

Test Results



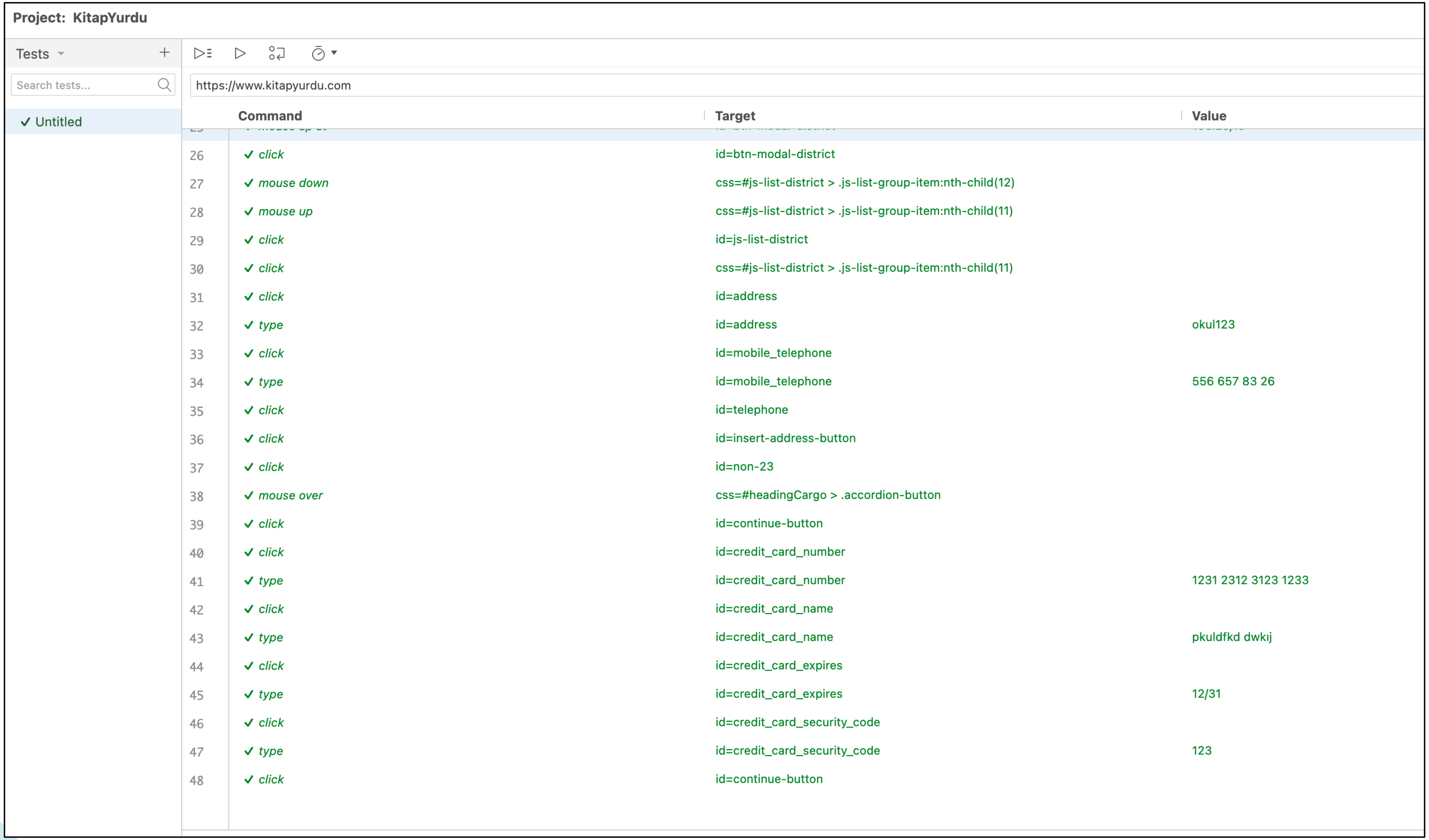
Project: KitapYurdu

✓ Untitled

https://www.kitapyurdu.com

	Command	Target	Value
1	✓ *open*	/index.php?route=account/account	
2	✓ *set window size*	1303x961	
3	✓ *click*	id=search-input	
4	✓ *type*	id=search-input	nutuk
5	✓ *send keys*	id=search-input	${KEY_ENTER}
6	✓ *click*	css=#product-645662 > .grid_7 > .name span	
7	✓ *click*	css=.icon__cta-fast	
8	✓ *click*	id=cart-items	
9	✓ *click*	id=js-cart	
10	✓ *click*	linkText=Checkout	
11	✓ *click*	linkText=Add New	
12	✓ *click*	id=title	
13	✓ *type*	id=title	okul
14	✓ *click*	id=firstname_companyname	
15	✓ *type*	id=firstname_companyname	okul1
16	✓ *click*	id=lastname_title	
17	✓ *type*	id=lastname_title	okul2
18	✓ *run script*	window.scrollTo(0,231)	
19	✓ *click*	id=btn-modal-zone	
20	✓ *click*	css=#js-list-zone > .js-list-group-item:nth-child(7)	
21	✓ *click*	id=btn-modal-county	
22	✓ *click*	css=#js-list-county > .js-list-group-item:nth-child(9)	
23	✓ *mouse down at*	id=btn-modal-district	158.25,13
24	✓ *mouse move at*	id=btn-modal-district	158.25,13
25	✓ *mouse up at*	id=btn-modal-district	158.25,13

Test Results



Project: KitapYurdu

https://www.kitapyurdu.com

	Command	Target	Value
26	✓ *click*	id=btn-modal-district	
27	✓ *mouse down*	css=#js-list-district > .js-list-group-item:nth-child(12)	
28	✓ *mouse up*	css=#js-list-district > .js-list-group-item:nth-child(11)	
29	✓ *click*	id=js-list-district	
30	✓ *click*	css=#js-list-district > .js-list-group-item:nth-child(11)	
31	✓ *click*	id=address	
32	✓ *type*	id=address	okul123
33	✓ *click*	id=mobile_telephone	
34	✓ *type*	id=mobile_telephone	556 657 83 26
35	✓ *click*	id=telephone	
36	✓ *click*	id=insert-address-button	
37	✓ *click*	id=non-23	
38	✓ *mouse over*	css=#headingCargo > .accordion-button	
39	✓ *click*	id=continue-button	
40	✓ *click*	id=credit_card_number	
41	✓ *type*	id=credit_card_number	1231 2312 3123 1233
42	✓ *click*	id=credit_card_name	
43	✓ *type*	id=credit_card_name	pkuldfkd dwkıj
44	✓ *click*	id=credit_card_expires	
45	✓ *type*	id=credit_card_expires	12/31
46	✓ *click*	id=credit_card_security_code	
47	✓ *type*	id=credit_card_security_code	123
48	✓ *click*	id=continue-button	

Conclusion: In this project, we used Python, Selenium WebDriver, and JUnit to test various functionalities of www.kitapyurdu.com. Using Boundary Value Analysis, we tested the search bar input length and the display of horror genre results. With Equivalence Class Partitioning, we covered scenarios like user registration and category selection. Use Case-based testing was applied to features such as purchasing and adding an address, while the login process was validated through Decision Table Testing. Some of these tests were automated, while others were done manually. Additionally, we evaluated the site's performance using Lighthouse. Overall, this approach allowed us to identify key areas for improvement and ensure more reliable and user-friendly interactions on the platform.

THANK YOU FOR LISTENING