17.06.2025

YAŞAR
ÜNİVERSİTESİ

# SE 2226 SOFTWARE QUALITY ASSURANCE AND TESTING

**Prepared By:**
Miray Yıldırım
İsmail Tekin
Dilara Kuyar
İlayda Gürbüzerol
**Department:** Software Engineering
**Lecturer:**
Doç. Dr. Korhan Karabulut

## Executive Summary

This report comprehensively details the software quality assurance and testing activities conducted for the Kitapyurdu.com online bookstore platform as part of the SE 2226 project. The primary objective was to ensure the functionality, reliability, and overall quality of the website, which is a major online platform offering a diverse range of books and educational materials. The testing encompassed various critical functionalities, including user registration, product listing, search bar, price filtering, user order placement, and login processes. Both automated and manual testing methodologies were employed, leveraging key technologies such as **Python, Selenium WebDriver, JMeter, JUnit, and Google Lighthouse**.

Key findings indicate successful validation of core functional flows, ensuring that users can register, log in, browse products, and place orders as expected. However, the Google Lighthouse analysis revealed significant areas for improvement, particularly in website performance (scoring a "poor" 49) and accessibility (scoring 64), highlighting issues such as slow loading times, unoptimized images, and insufficient elements for users with disabilities. Best practices and security also showed room for enhancement, notably concerning thirdparty cookies and security headers. A detailed analysis of average response times further corroborated performance concerns, showing periods of increasing latency during testing.

This report outlines the deviations from planned testing, factors that impeded progress, and provides a comprehensive set of test measures. It concludes with valuable lessons learned from the project, offering key recommendations for future development and testing efforts to enhance Kitapyurdu.com's user experience and overall quality. While testing met its completion criteria in terms of execution and issue identification, the report clearly articulates the areas where the software itself did not meet optimal quality standards, serving as a critical diagnostic tool for continuous improvement.

## Introduction of 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

### 1. Black-Box Testing

Black-box testing, also known as functional testing, focuses on the functionality of the software without regard to the internal structure or workings of the application. In the context of the testing performed for the website www.kitapyurdu.com, the black-box testing strategy involves validating the system's behavior based on the provided requirements and specifications.

1. **Equivalence Partitioning:** This testing technique involves dividing input values into equivalence classes and testing representative values from each class. For Kitapyurdu.com, this was applied to:
    - **User Registration Module (Email Input Field Validation):** Valid and invalid email formats (e.g., correct test@example.com, missing @, missing domain, empty fields) were tested to ensure the system correctly accepts valid inputs and rejects invalid ones with appropriate error messages. ○ **Product Listing and Category Filtering Functionality:** Representative valid category selections (e.g., "Children's Books," "Science Fiction") were tested to ensure accurate filtering, along with scenarios for invalid or non-existent category selections.
2. **Boundary Value Analysis:** This technique focuses on testing values at the boundaries of input ranges, as errors often occur at these points. For Kitapyurdu.com, this was applied to:
    - **Search Bar Functionality:** Tests included minimal input (e.g., a single character or an empty string) and inputs at or around the typical maximum character limit for search queries, to evaluate the search bar's robustness and accuracy.
    - **Price Filtering/Numeric Input Fields:** Boundary values for price ranges (e.g., minimum valid price, values just above/below a certain threshold, extremely large numbers) were tested to ensure correct processing and error handling for numerical inputs.
3. **Use Case Testing:** Use case testing verifies the system's ability to handle typical user scenarios, ensuring end-to-end functionality. For www.kitapyurdu.com, a primary use case tested was:
    - **User Order Placement Workflow:** This involved a complete scenario from a user logging into an existing account, searching for a book, adding it to the shopping cart, proceeding through checkout (address, payment), and finally confirming the order. This verified the entire core e-commerce process.
4. **Decision Tables:** Decision tables help ensure that all combinations of conditions and actions are systematically tested, particularly for complex logic. In the context of www.kitapyurdu.com, decision tables were used to test: ○ **User Login Process:** Various scenarios were tested based on user inputs for credentials (e.g., valid username/valid password, valid username/invalid password, invalid username/any password), confirming the system's precise response for each combination, such as successful login or specific error messages.

## Tools and Technologies

In the testing process for www.kitapyurdu.com, the following tools and technologies were employed to ensure comprehensive testing coverage and effective automation:

1. **Programming Languages & Testing Frameworks: Python and Java JUnit:**
    - **Python** was chosen as a primary programming language for writing automated tests due to its simplicity, readability, and extensive libraries for web automation. Python provides robust support for interacting with web elements and handling various test scenarios efficiently. ○ **JUnit** (a framework primarily for Java) was also considered or its principles applied, indicating a structured approach to unit or integration testing where Java components might have been involved in the broader testing ecosystem. This allowed for organized test case development and execution, ensuring code-level validation where applicable.
2. **Testing Framework: Selenium WebDriver:**
    - Selenium WebDriver is a widely used testing framework for automating web browsers. It allowed testers to simulate user interactions with web elements on www.kitapyurdu.com, navigate through web pages, and verify expected behaviors such as user login, product searches, and order placement. Selenium WebDriver offers support for multiple programming languages, including Python, making it suitable for cross-platform and web-based testing needs.
3. **Performance & Quality Audit Tools: JMeter and Google Lighthouse:**
    - **JMeter** was utilized for performance evaluation, specifically to simulate user load and measure page loading times and server response under various conditions for www.kitapyurdu.com. This provided crucial insights into the website's scalability and stability under stress.
    - **Google Lighthouse** was employed as an automated open-source tool to audit web page quality. It provided comprehensive reports and scores on www.kitapyurdu.com's performance, accessibility, adherence to best practices, and search engine optimization (SEO), highlighting areas for improvement beyond functional correctness.

By leveraging Python, Java JUnit, and Selenium WebDriver for test automation, alongside JMeter for performance assessment and Google Lighthouse for holistic quality audits, testers were able to create robust, maintainable test suites that validate the functionality, user experience, and technical health of www.kitapyurdu.com across various aspects. This comprehensive testing approach contributes significantly to the overall quality and reliability of the website, ensuring a seamless experience for its users.

# Test Plan

## 1. Introduction:

The test plan outlines the strategy and approach for testing www.kitapyurdu.com. It defines the scope, objectives, test scenarios, testing techniques, resources, and schedules for the testing phase of this prominent online bookstore platform.

## 2. Objectives:

The primary objectives of the testing phase for Kitapyurdu.com are as follows:

- Ensure the functionality of Kitapyurdu meets the specified requirements.
- Validate the user experience and interface design across various interactions.
- Ensure compatibility across different browsers and devices to provide a consistent user experience.
- Verify the performance and responsiveness of the application under various loads.
- Identify and document any defects or areas for improvement in the platform.

## 3. Scope:

The test plan covers the following aspects of testing for Kitapyurdu.com:

- **Functional Testing:** Validate the functional requirements of the Kitapyurdu platform, including user registration, login, product search, filtering, order placement, and other core e-commerce functionalities.
- **User Interface (UI) Testing:** Assess the usability, navigability, and accessibility of the interface design, ensuring an intuitive and consistent user experience.
- **Compatibility Testing:** Ensure that the website functions correctly and consistently across various browsers and device types .
- **Performance Testing:** Evaluate the responsiveness, speed, and scalability of the application under different load conditions to identify bottlenecks and ensure optimal performance.
- **Automation Testing:** Automate repetitive and critical test cases to streamline testing processes, improve efficiency, and support faster regression cycles.

## 4. Testing Techniques:

The following testing techniques will be employed for Kitapyurdu.com:

- **Black-box Testing:** This overarching approach assesses the functionality of the Kitapyurdu application from an external perspective, without delving into its internal code structure.
- **Equivalence Partitioning:** Input ranges will be divided into equivalent classes to design test cases effectively. This was applied, for example, in testing user registration email input fields and product category filtering.

- **Boundary Value Analysis:** Test cases will focus on boundary conditions of input variables (e.g., minimum/maximum values, just inside/outside valid ranges) to uncover defects. This was used for testing the search bar input length and numeric input fields like price filters.
- **State Transition Testing:** This technique will validate transitions between different states of the application, ensuring the system behaves as expected when conditions change (e.g., login state, shopping cart state changes).
- **Use Case Testing:** Scenarios based on typical user interactions and workflows will be evaluated end-to-end to ensure the system handles complete processes correctly. A key example is the comprehensive order placement workflow.
- **Decision Tables:** Decision tables will be created to define different combinations of inputs and expected outcomes for complex logic, such as various scenarios during the user login process.
- **Compatibility Testing:** The website will be tested across various browsers and devices to ensure consistent functionality and appearance for a broad user base.
- **Automation Testing:** Automated test execution will be performed utilizing tools such as **Selenium WebDriver** (for functional UI automation) and **Python** as the scripting language. **JMeter** will be used for performance evaluation, and **Google Lighthouse** for auditing web vital scores (performance, accessibility, best practices, SEO).

# Test Execution

## 1-Use Cases Test Scenarios, Equivalence Classes, Decision Table and Boundary Value Analysis

In the www.kitapyurdu.com project, we designed test scenarios to evaluate both the functionality and usability of the website. To ensure comprehensive testing, we applied **Equivalence Class Partitioning**, **Boundary Value Analysis**, **Use Case Testing**, and **Decision Table Testing** techniques, often integrating them within the same corresponding test case tables.

### Equivalence Partitioning

In this section, three real test cases are designed using the Equivalence Class Partitioning (ECP) technique. Inputs are divided into valid and invalid equivalence classes, and representative values from each class are selected to minimize test effort while maximizing coverage. The selected functionalities reflect real user interactions on Kitapyurdu.com such as applying price filters, fillingthe registration form, and adding a product to the cart.

**Test Case ID:** ECP-KY-01
**Tested Area:** Search Page – Price Filter **Equivalence Classes:**

- Valid: 0 ≤ price ≤ 1000 • Invalid: price <

0, price > 1000

**Test Description:**
The user applies a price filter between 100 and 500 TL on the search results page.

**Input Value:** Min: 100, Max: 500 **Expected Result:**
Only books priced between 100 and 500 TL are listed. Products outside this range are not shown.

| Equivalence Class | Range | Input Example | Expected Result |
|---|---|---|---|
| Valid | 0 ≤ price ≤ 1000 | Min: 100, Max: 500 | Only books priced between 100-500 TL are displayed. |
| Invalid (Lower Bound) | price < 0 | Min: -10 | Error message or no results (system should reject negative values). |
| Invalid (Upper Bound) | price > 1000 | Max: 1500 | Error message or no results (system should reject values above the max threshold). |

**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. |

**Test Case ID:** ECP-KY-03
**Tested Area:** Product Detail Page – Quantity Input **Equivalence Classes:**

• Valid: 1 ≤ quantity ≤ 99

• Invalid: quantity = 0, quantity < 0, quantity > 99

**Test Description:**
The user manually enters “0” as the quantity on a book’s product detail page and clicks the “Add to Cart” button.

**Input Value:** 0
**Expected Result:**
The system shows an error message like “Please enter a valid quantity.” The product is not added to the cart.

| Equivalence Class | Range | Input Example | Expected Result |
|---|---|---|---|
| Valid | 1 ≤ Qty ≤ 99 | 5 | Product added to cart successfully. |
| Invalid (Zero) | Qty = 0 | 0 | Error: "Please enter a valid quantity." No cart update. |
| Invalid (Negative) | Qty < 0 | -2 | Error: "Please enter a valid quantity." No cart update. |
| Invalid (Exceeds Max) | Qty > 99 | 100 | Error: "Maximum quantity is 99." No cart update. |

**Test Case ID:** ECP-KY-04
**Tested Feature:** Homepage/Search Bar – User Input
**Equivalence Classes:**

• **Valid:** 1–255 characters (searchable input)

• **Invalid:** 0 characters (empty), >255 characters (too long) **Test Description:**

1. The user submits a search request with a valid input: “historybooks”.

2. The user submits an empty input.

**Input Values:**

• Case A (Valid): "history books" (14 characters)

• Case B (Invalid): "" (0 characters) **Expected Result:**

• Case A: Search results page is displayed with relevant products.

• Case B: The system shows an error like “Please enter a keyword.”

| Equivalence Class | Length | Input Example | Expected Result |
|---|---|---|---|
| Valid | 1–255 chars) | "history books" (14 | Displays relevant search results. chars |
| Invalid (Empty) | 0 chars | "" | Error: "Please enter a keyword." No search performed. |
| Invalid (Too Long) | >255 chars | 256-character string | Error: "Input too long." or truncation (if applicable). |

**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/Nonselectable | (Grayed-out option) | Prevents selection; no filtering applied. |

**Boundary Value Analysis**

Test Case ID: BVA-KY-01
Tested Feature: Search Bar Input Length Functionality  Equivalence
Classes:
Valid: 0≤e≤100 (representing valid input lengths for search queries)
Invalid: e<0 or e>100 (representing invalid input lengths)  Test
Description:
This test case verifies the behavior of the search bar based on various input lengths, covering valid and invalid equivalence classes as defined by the system's specifications. It includes testing boundary values and typical values within each class to ensure correct system responses such as "No result", "Related results", "Impossible", and "Error (uses only first 100 chars)".
Input Value: Varies (e.g., "0", "1", "2", "99", "100", "-5", "165")
Expected Result: Varies (e.g., "No result", "Related Results", "Impossible", "Error (uses only first 100 chars)")

| 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) |

Test Case ID: BVA-KY-02
Tested Feature: Max Price Filter for Horror Genre  Equivalence
Classes:
Valid: 0<e≤10  21
 −1 (representing valid price ranges for filtering, split into two sub-classes based on expected results)
Invalid: e<0 or e≥10^21
  (representing invalid price inputs)  Test
Description:
This test case evaluates the maximum price filter functionality for the "Horror" genre by inputting various price values. It aims to confirm that the system correctly processes valid price ranges, yielding the expected number of results, and handles invalid inputs by displaying appropriate error messages such as "Error (turns into positive)" or "Error (changes number notation)".

Input Value: Varies (e.g., "1", "2", "958", "958,40", "986", "10^21 −2", "10^21−1", "-5", "10^23")
Expected Result: Varies (e.g., "19 result", "20 result", "492 result", "493 result", "Error (turns into positive)", "Error (changes number notation)")

| 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) |

Test Case ID: BVA-KY-03
Tested Feature: User Registration Form Validation Equivalence Classes:
Valid:
Name Length: 2≤ Name Length ≤30
Name UTF-8: Yes
Lastname Length: 2≤ Lastname Length ≤30
Lastname UTF-8: Yes
Email Length: 10≤ Email Length ≤50
Email Valid: Yes
Email Already Registered: No
Password Length: 8≤ Password Length ≤20
Password UTF-8: Yes
Password Conformation Accurate: Yes
KVKK Agreed: Yes Not
Human Verified: Yes
Invalid:
Name Length: 2> Name Length or Name Length >30
Name UTF-8: No
Surname Length: 2> Surname Length or Surname Length >30
Surname UTF-8: No
Email Length: 10> e-mail Length or e-mail Length >50
Email Valid: No
Email Already Registered: Yes
Password Length: 8> Password Length or Password Length >20
Password UTF-8: No
Password Conformation Accurate: No
KVKK Agreed: No Not
Human Verified: No
Test Description:

This comprehensive test case validates the user registration form by evaluating all defined equivalence classes for various input fields including name, last name, email, and password, along with consent and human verification. It covers valid inputs that should lead to successful progression or completion of registration, and invalid inputs that should trigger specific error messages such as "First Name must be between 2 and 30 characters!", "E-Mail Address does not appear to be valid!", or "Ask For Different e-mail". The test aims to ensure the robustness of the form's data validation and user feedback mechanisms.
Input Value: Varies based on each specific field and its respective equivalence class (e.g., Name Length = 2, Name Length = 35, e-mail Length = 0, Password Length = 24, etc.)
Expected Result: Varies based on each specific field and its respective equivalence class (e.g., "Proceeds Next Step", "Compeletes Registration", "Ask For Valid Input", "First Name must be between 2 and 30 characters!", "E-Mail Address does not appear to be valid!", "Ask For Different e-mail", etc.)

| Equivalence Class | | Test Input | Expected Output |
|---|---|---|---|
| E1 | 2 <= Name Length <= 30 | | |
| | | 2 | Proceeds Next Step |
| | | 3 | Proceeds Next Step |
| | | 29 | Proceeds Next Step |
| | | 30 | Proceeds Next Step |
| E2 | Is Name UTF-8 | | |
| | | Yes | Proceeds Next Step |
| | | No | Ask For Valid Input |
| E3 | 2 <=Lastname Length <= 30 | | |
| | | 2 | Proceeds Next Step |
| | | 3 | Proceeds Next Step |
| | | 29 | Proceeds Next Step |
| | | 30 | Proceeds Next Step |
| E4 | Is Lastname UTF-8 | | |
| | | Yes | Proceeds Next Step |
| | | No | Ask For Valid Input |
| E5 | 10 <= Email Length <= 50 | | |
| | | 10 | Proceeds Next Step |
| | | 11 | Proceeds Next Step |
| | | 49 | Proceeds Next Step |
| | | 50 | Proceeds Next Step |
| E6 | Is e-mail Vaild | | |
| | | Yes | Proceeds Next Step |
| | | No | Ask For Valid Input |
| E7 | Is e-mail Already Registered | | |
| | | Yes | Ask For Different e-mail |
| | | No | Proceeds Next Step |
| E8 | 8 <= Password Length <= 20 | | |
| | | 8 | Proceeds Next Step |
| | | 9 | Proceeds Next Step |

| | | | |
|---|---|---|---|
| | | 19 | Proceeds Next Step |
| | | 20 | Proceeds Next Step |
| E9 | Is Password UTF-8 | | |
| | | Yes | Proceeds Next Step |
| | | No | Ask For Valid Input |
| E10 | Is Password Conformation Accurate | | |
| | | Yes | Proceeds Next Step |
| | | No | Ask For password correction |
| E11 | Is KVKK Agreed | | |
| | | Yes | Proceeds Next Step |
| | | No | Ask For Agreement |
| E12 | Is Not Human Verified | | |
| | | Yes | Compeletes Registration |
| | | No | Ask For Try Again Later |
| U1 | 2 > Name Length | Name Length = 1 | *First Name must be between 2 and 30 characters!* |
| U2 | Name Length > 30 | Name Length = 35 | *First Name must be between 2 and 30 characters!* |
| U3 | 2 > Surname Length | Surname Length = 1 | *Last Name must be between 2 and 30 characters!* |
| U4 | Surname Length > 30 | Surname Length = 34 | *Last Name must be between 2 and 30 characters!* |
| U5 | 10 > e-mail Length | e-mail Length = 0 | *E-Mail Address does not appear to be valid!* |
| U6 | e-mail Length > 50 | e-mail Length = 58 | *E-mail length may be at most 50 characters!* |
| U7 | 8 > Password Length | Password Length = 3 | *Password must be between 8 and 20 characters!* |
| U8 | Password Length > 20 | Password Length = 24 | *Password must be between 8 and 20 characters!* |

## Use Case Scenarios:

Use Case ID: UCD-KY-01
Tested Feature: Book Search Functionality Actors:
User
Pre-Conditions:
* The user has internet access.
* The Kitapyurdu.com homepage is available for use.
Test Description:

This use case describes the process of a user searching for a specific book on Kitapyurdu.com using the website's search engine. The scenario covers the steps from clicking the search bar, entering the book title, and initiating the search, to the system displaying the search results and the user reviewing them.
Main Scenario Steps:
1.The user clicks on the search bar.
2.The user enters the book title in the search bar.
3.The user clicks the “Search” button.
4.The system displays the search results.
5.The user reviews the search results page.
Post-Conditions: The user can see the search results containing the book they searched for.
Expected Result: The system should successfully display relevant search results based on the entered book title.

| | |
|---|---|
| **Use Case Scenario Name:** | User searches for a book |
| **Description:** | The user searches for a specific book using the search engine on Kitapyurdu.com . |
| **Actors:** | User |
| **Pre-Conditions:** | • The user has internet access.<br>• The Kitapyurdu.com homepage is available for use. |
| **Main Scenario Steps:** | 1) The user clicks on the search bar.<br>2) The user enters the book title in the search bar.<br>3) The user clicks the “Search” button.<br>4) The system displays the search results.<br>5) The user reviews the search results page. |
| **Post-Conditions:** | The user can see the search results containing the book they searched for. |

Use Case ID: UCD-KY-02
Tested Feature: Add to Cart Functionality
Actors: User, System  Pre-Conditions:
* The user has internet access.
* The Kitapyurdu.com homepage is available to use.
Test Description:
This use case details how a user adds a book to their shopping cart from a product details page on Kitapyurdu.com. It includes navigating to the product page via search, clicking the "Add to Cart" button, and observing the system's confirmation message and cart icon update.
Main Scenario Steps:
1.The user accesses the Kitapyurdu.com home page.
2.The user clicks on the search bar and enters the book name.
3.The user presses the search button.
4.The system displays the search results.
5.The user clicks on a product from the search results to view the product detail page.
6.The user clicks the “Add to Cart” button.
7.The system adds the product to the cart.
8.The system displays a message, such as “You have added the item to your cart.” in the topleft corner of the page.
9.The system updates the cart icon.
Post-Conditions: The book is successfully added to the cart and the cart is updated.
Expected Result: The book should be added to the shopping cart, a confirmation message should be displayed, and the cart icon should reflect the added item.

| | |
|---|---|
| **Use Case Scenario Name:** | User Adds a Book to the Cart |
| **Description:** | The user adds a book to the shopping cart from the product details page. |
| **Actors:** | User, System |
| **Pre-Conditions:** | • The user has internet access.<br>• The Kitapyurdu.com homepage is available to use. |
| **Main Scenario Steps:** | 1) The user accesses the Kitapyurdu.com home page.<br>2) The user clicks on the search bar and enters the book name.<br>3) The user presses the search button.<br>4) The system displays the search results.<br>5) The user clicks on a product from the search results to view the product detail page.<br>6) The user clicks the "Add to Cart" button.<br>7) The system adds the product to the cart.<br>8) The system displays a message, such as "You have added the item to your cart." in the top-left corner of the page.<br>9) The system updates the cart icon. |
| **Post-Conditions:** | The book is successfully added to the cart and the cart is updated |

Use Case ID: UCD-KY-03
Tested Feature: Order Placement Process
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.
Test Description:
This use case describes the end-to-end process of a logged-in user placing an order on Kitapyurdu.com. It encompasses searching for a book, adding it to the cart, proceeding to checkout, entering delivery address and payment information, and finally confirming the order.
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 topleft 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.  Expected Result: The order should be successfully processed, and an order summary should be displayed to the user.

| | |
|---|---|
| **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.<br>• The Kitapyurdu.com homepage is accessible.<br>• The user has a valid Kitapyurdu account.<br>• The user has logged in. |
| **Main Scenario Steps:** | 1) The user searches for a book and selects a product from the search results.<br>2) The user clicks the "Add to Cart" button.<br>3) The system adds the product to the cart.<br>4) The system displays a message, such as "You have added the item to your cart." in the top-left corner of the page.<br>5) The user opens the cart and clicks the "Checkout" button.<br>6) The user enters valid delivery address information.<br>7) The user selects Shipping Selection.<br>8) The user enters the continue button.<br>9) The user enters valid payment information.<br>10) The user clicks the "Confirm Order" button. |
| **Post-Conditions:** | The order is successfully created, and the user sees the order summary. |

Use Case ID: UCD-KY-04
Tested Feature: Address Information Update Actors:
User
Pre-Conditions:
* The user has an active internet connection.
* The user has a valid Kitapyurdu account.
* The Kitapyurdu.com account setting page is accessible.
Test Description:
This use case outlines the steps a user takes to update their address information within their Kitapyurdu.com account. It covers the login process, navigating to the "My Account" and "My Addresses" sections, entering new address details, saving changes, and the system's validation of the updated information.
Main Scenario Steps:
1.The user clicks the "Login" button and enters valid email and password.
2.The system verifies credentials and logs the user in.
3.The user navigates to the "My Account" page.
4.The user selects the field "My Addresses" to update.
5.The user selects the field "New Address" to update.
6.The user enters the new information and saves the changes.
7.The system validates the updated information. Post-Conditions: The account information is successfully updated and the user is notified. Expected Result: The address information should be successfully updated in the user's account, and a confirmation message should be displayed.

| | |
|---|---|
| **Use Case Scenario Name:** | User Updates Address Information |
| **Description:** | The user updates their address information. |
| **Actors:** | User |
| **Pre-Conditions:** | • The user has an active internet connection.<br>• The user has a valid Kitapyurdu account.<br>• The Kitapyurdu.com account setting page is accessible. |
| **Main Scenario Steps:** | 1) The user clicks the "Login" button and enters valid email and password.<br>2) The system verifies credentials and logs the user in.<br>3) The user navigates to the "My Account" page.<br>4) The user selects the field "My Addresses" to update.<br>5) The user selects the field "New Address" to update.<br>6) The user enters the new information and saves the changes.<br>7) The system validates the updated information. |
| **Post-Conditions:** | The account information is successfully updated and the user is notified. |

## Decision Table Testing:

Test Case ID: DTT-KY-01
Tested Feature: User Login Form
Description: This test case validates the behavior of the login form when users attempt to log in with various combinations of valid and invalid usernames and passwords. It covers scenarios where both credentials are correct, or one or both are incorrect, to ensure the system responds with the appropriate success or error messages.

Conditions & Actions:
DT01: Valid Username, Valid Password → Login successful (Correct credentials provided)
DT02: Valid Username, Invalid Password → "No match for EMail Address and/or Password." message is displayed (Valid user, but wrong password)
DT03: Invalid Username, Valid Password → "No match for EMail Address and/or Password." message is displayed (Username not registered in the system)
DT04: Invalid Username, Invalid Password → "No match for EMail Address and/or Password." message is displayed (Username not registered in the system)  Expected Result: The system should accurately reflect the login attempt's outcome, either granting access or displaying the specified error message for incorrect credentials.

| 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 |

Test Case ID: DTT
-KY-02
Tested Feature: Price Filtering Mechanism
Description: This test case examines the functionality of the price filter, verifying its behavior when valid and logically consistent price ranges are applied, as well as when illogical or inconsistent ranges are entered. It ensures the system correctly lists products or displays an appropriate message indicating no products are found.
Conditions & Actions:
DT05: Valid minimum price, Valid maximum price → Matching products are listed (Both values are within acceptable range and in correct order)
DT06: Higher than maximum price, Lower than minimum price → "There are no products to list." message is displayed (Logical inconsistency in range) Expected Result: The system should either display a list of products matching the valid price range or a "There are no products to list." message for invalid or inconsistent ranges.

| Test Case ID | Min Price | Max Price | Expected Result | Description |
|---|---|---|---|---|
| DT05 | Valid minimum | Valid maximum | Matching products are listed | Both values are within acceptable range and in correct order |
| DT06 | Higher than maximum | Lower than minimum | "There are no products to list. " message is displayed | Logical inconsistency in range |

Test Case ID: DTT

-KY-03

Tested Feature: Search Functionality

Description: This test case aims to verify how the application handles different types of search inputs, specifically focusing on valid search terms that should yield relevant results and invalid characters that should trigger a "no match" message.

Conditions & Actions:

DT07: Valid search term → Relevant results are displayed (User enters a known author or book title)

DT08: Invalid characters → "There is no product that matches the search criteria." message is displayed (User enters a string with only special or nonsensical characters) Expected Result: The system should return relevant search results for valid queries and a "no product match" message for queries with invalid characters.

| Test Case ID | Search Query | Expected Result | Description |
|---|---|---|---|
| DT07 | Valid search term | Relevant results are displayed | User enters a known author or book title |
| DT08 | Invalid characters | "There is no product that matches the search criteria. " message is displayed | User enters a string with only special or nonsensical characters |

Test Case ID: DTT

-KY-03

Tested Feature: Search Functionality

Description: This test case aims to verify how the application handles different types of search inputs, specifically focusing on valid search terms that should yield relevant results and invalid characters that should trigger a "no match" message.

Conditions & Actions:

DT07: Valid search term → Relevant results are displayed (User enters a known author or book title)

DT08: Invalid characters → "There is no product that matches the search criteria." message is displayed (User enters a string with only special or nonsensical characters) Expected Result: The system should return relevant search results for valid queries and a "no product match" message for queries with invalid characters.

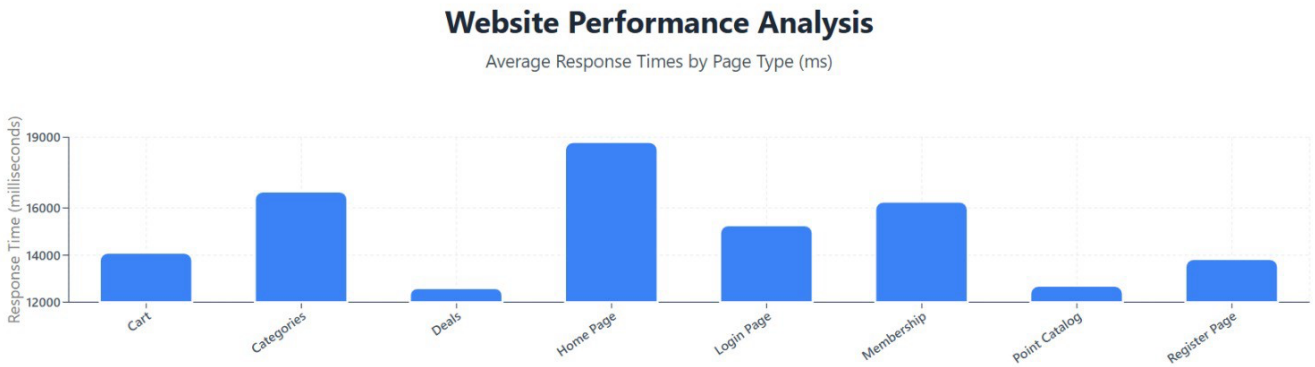
| Test Case ID | First Name | Last Name | E-Mail | Password | KVKK Agreement | Expected Result | Description |
|---|---|---|---|---|---|---|---|
| DT09 | Valid | Valid | Valid | Valid | Checked | Registration successful | All inputs are valid |
| DT10 | Too short | Valid | Valid | Valid | Checked | “First Name must be between 2 and 30 characters!” message is displayed | Invalid first name |
| DT11 | Valid | Too short | Valid | Valid | Checked | “Last Name must be between 2 and 30 characters!” message is displayed | Invalid last name |
| DT12 | Valid | Valid | Invalid | Valid | Checked | “E-Mail Address does not appear to be valid!” message is displayed | Invalid email |
| DT13 | Valid | Valid | Valid | Invalid | Checked | “Password must be between 8 and 20 characters!” message is displayed | Invalid password |
| DT14 | Valid | Valid | Valid | Valid | Not checked | “You must agree to the KVKK!” message is displayed | KVKK Agreement checkbox not checked |

Test Case ID: DTT

# 2.Performance Evaluation of Page Loading  Times with JMeter

## Key Metrics Overview

1. # Samples: The number of requests sent for each label (4 for all labels).
2. Average: The average response time for the requests.
3. Median: The median response time for the requests.
4. 90% Line, 95% Line, 99% Line: The response time below which 90%, 95%, and 99% of the samples fall, respectively.
5. Min: The minimum response time.
6. Max: The maximum response time.
7. Error %: The percentage of requests that resulted in errors.
8. Throughput: The number of requests per second.
9. Received KB/sec: The amount of data received from the server per second.
10. Sent KB/sec: The amount of data sent to the server per second.




## Detailed Analysis

**1.Error Rates**

● No Errors: None of the pages have any errors, indicating that all requests were successful.

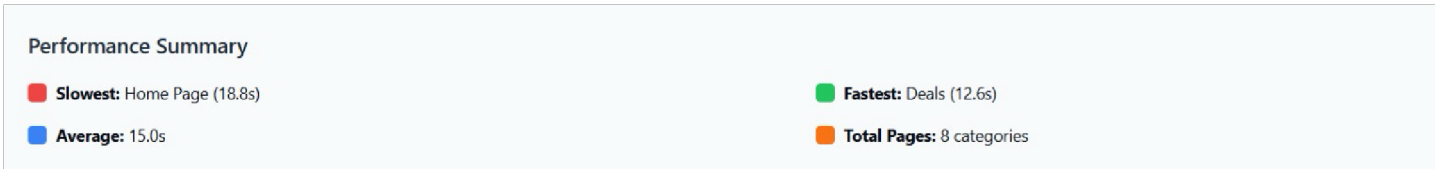
**2.Sent Data Transfer Rates (Kbytes/sec)**

●Highest: 14.85 (Total)
●Lowest: 0.27 (Login page)

**3.Minimum Response Time (ms)**

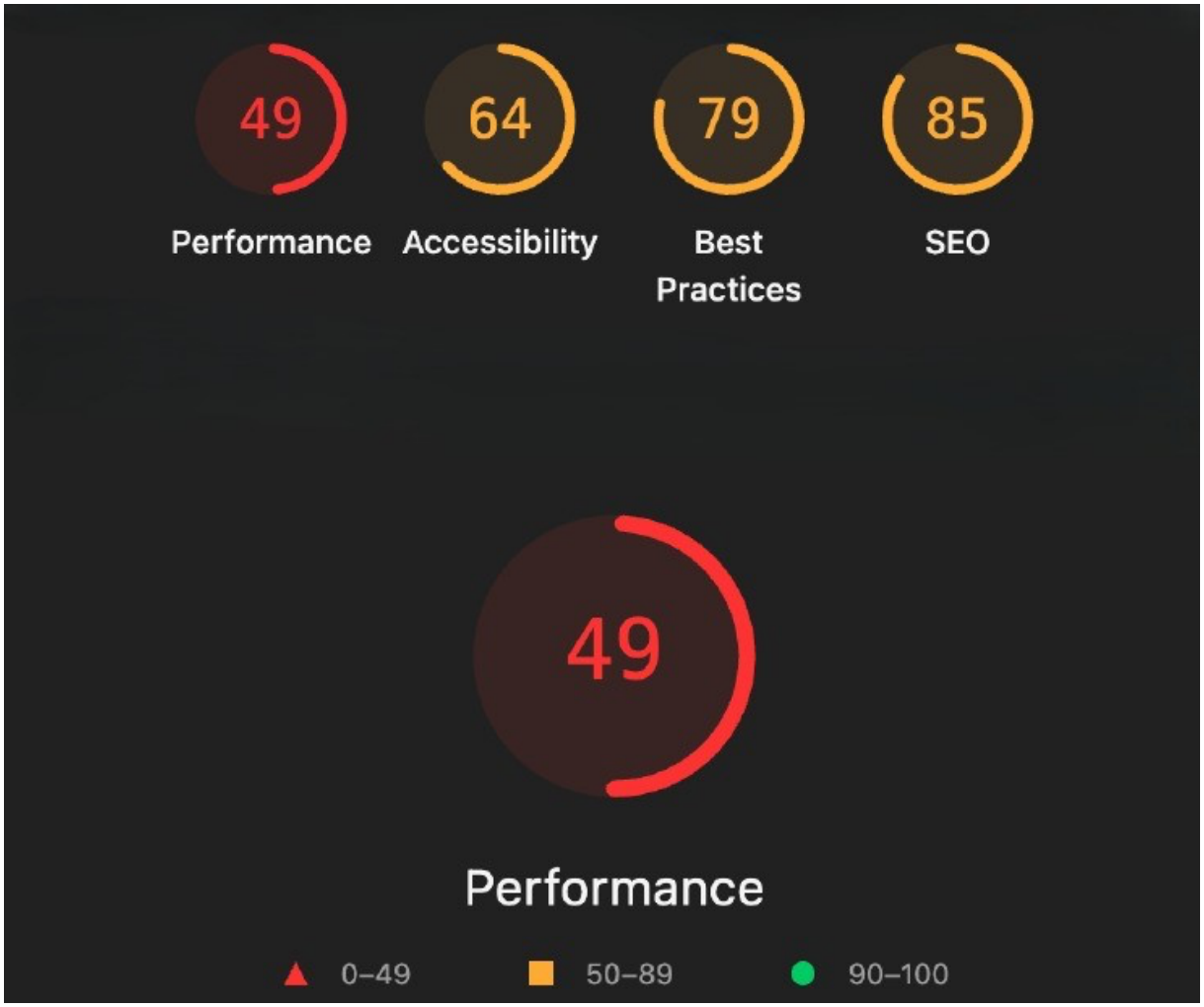
●Highest: 2895.00 (Cart)
●Lowest: 2.00 (Deals)

**4.Maximum Response Time (ms)** ●Highest:
50104.00 (Login Page)
●Lowest: 4876.00 (Point Catalog)




**Summary:**

## 3.Performance Analysis with Google Lighthouse



## Performance Rating (49)

**Key Metrics:**

- **First Contentful Paint (FCP):** 3.8 seconds (Indicates content appears slowly)
- **Largest Contentful Paint (LCP):** 6.8 seconds (Main content takes a very long time to load)
- **Total Blocking Time (TBT):** 490 ms (Significant time where the main thread is blocked, affecting interactivity)
- **Speed Index:** 4.8 seconds (How quickly content is visually displayed during page load) - **Cumulative Layout Shift (CLS):** 0.05 (Indicates minimal unexpected layout shifts, which is good)
- **Time to Interactive:** 8.8 seconds (Time until the page is fully interactive)

**Diagnostics:**

- Serve images in next-gen formats (significant saving opportunity)
- Defer offscreen images (lazy loading)
- Efficiently encode images
- Reduce initial server response time
- Eliminate render-blocking resources (e.g., CSS, JS)
- Remove unused JavaScript
- Avoid enormous network payloads
- Reduce JavaScript execution time (6.0 seconds detected)
- Avoid long main-thread tasks
- Ensure text remains visible during webfont load
- Serve static assets with an efficient cache policy

## Accessibility (64)

**Names and Labels:**

- Form elements do not have associated labels (1 element found). - Links do not have a discernible name (1 element found).

**Contrast:**

- Background and foreground colors do not have a sufficient contrast ratio (9 elements found).

**Navigation:**

- Links do not have a discernible name, which can hinder navigation for assistive technologies.

**ARIA:**

- ARIA input fields do not have accessible names (1 element found).
- *Note: Other ARIA-related audits passed, but these are identified failures.*

### Best Practises (79)

**General Issues:**
- Detected 1 JavaScript error in the console.
- Image elements do not have explicit width and height (1 element found, which can lead to layout shifts).

**Trust and Safety Issues:**
- External anchors (links) do not use rel="noopener" (1 element found), which can pose a security risk when opening links in new tabs.
- *Note: The site uses HTTPS and has a valid doctype, which are positive indicators.*

**User Experience Issues:**
- Image elements without explicit width and height can negatively impact the visual stability of the page during loading.

## Conclusion

In summary, we conducted a comprehensive evaluation of **www.kitapyurdu.com** as part of our SE 2226 project, leveraging **Python**, **Selenium WebDriver**, **JMeter**, **JUnit**, and **Google Lighthouse**. Our primary objective was to thoroughly test various scenarios by meticulously preparing **Equivalence Class Partitioning**, **Boundary Value Analysis**, **Use Case-based testing**, and **Decision Table Testing** for our key functionalities. To enhance efficiency and repeatability, we automated several of these tests using Python.

Our assessment covered multiple critical facets of the website, including its **functionality, compatibility, performance, accessibility, and adherence to best practices**. Despite the website's suboptimal performance and identified accessibility gaps, it generally provides a functionally satisfactory user experience for core operations. Key findings and detailed recommendations for improvements, particularly concerning performance optimization and accessibility enhancements, have been meticulously documented in this report.

By diligently addressing the identified issues, especially in terms of performance optimization, improved accessibility, and enhanced error handling, **www.kitapyurdu.com** can significantly elevate its overall user experience and reliability. This report provides a robust foundation for targeted enhancements, ensuring that the platform can better meet evolving user expectations and operational demands.

```
import org.junit.After; import org.junit.Before; import
org.junit.Test; import org.openqa.selenium.By; import
org.openqa.selenium.JavascriptExecutor; import
org.openqa.selenium.WebDriver; import
org.openqa.selenium.WebElement; import
org.openqa.selenium.firefox.FirefoxDriver; import
org.openqa.selenium.support.ui.ExpectedConditions;
import org.openqa.selenium.support.ui.WebDriverWait;

import java.time.Duration;
import java.util.HashMap;
import java.util.Map;

import static org.junit.Assert.assertNotNull;

public class KitapyurduTest {

    private WebDriver driver;
private JavascriptExecutor js;
    private Map<String, Object> vars;

    @Before    public void setUp()
{        driver = new FirefoxDriver();
js = (JavascriptExecutor) driver;
        vars = new HashMap<>();
    }

    @After
    public void tearDown() {
        //driver.quit();
    }

    @Test
    public void loginValidUser() {
        driver.get("https://www.kitapyurdu.com/");
driver.manage().window().maximize();

        js.executeScript("window.scrollTo(0,47)");
        driver.findElement(By.linkText("Giriş Yap")).click();


        driver.findElement(By.id("login-email")).sendKeys("test_user@example.com"); // Placeholder for test
user email
        driver.findElement(By.id("login-password")).sendKeys("password"); // Use environment variables or
placeholders for security
        driver.findElement(By.id("login-button")).click();
```

```
        WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(10));
        WebElement userArea =
wait.until(ExpectedConditions.visibilityOfElementLocated(By.cssSelector(".user")));
assertNotNull(userArea);
    }
}
```

```
import unittest
from selenium import webdriver from
selenium.webdriver.chrome.service import Service from
selenium.webdriver.common.by import By from
selenium.webdriver.common.keys import Keys from
selenium.webdriver.support.ui import WebDriverWait from
selenium.webdriver.support import expected_conditions as EC
import time import os
from config import CHROMEDRIVER_PATH, BASE_URL

class TestCategorySelectionDropdown(unittest.TestCase):

   def setUp(self):
      service = Service(executable_path=CHROMEDRIVER_PATH)
      self.driver = webdriver.Chrome(service=service)
self.driver.maximize_window()
self.driver.get(BASE_URL)
      search_box = self.driver.find_element(By.ID, "search-input")
search_box.send_keys("kitap")
search_box.send_keys(Keys.RETURN)
      WebDriverWait(self.driver, 10).until(
         EC.presence_of_element_located((By.XPATH, "//a[contains(@href, '/kategori/')]"))
      )

   def tearDown(self):
self.driver.quit()

   def test_category_valid_selection(self):        print("\n--- Test Case 5:
Category Selection Dropdown - Valid ---")        try:
         children_books_category = self.driver.find_element(By.XPATH, "//a[text()='Çocuk
   Kitapları']")
children_books_category.click()
WebDriverWait(self.driver, 10).until(
            EC.url_contains("cocuk-kitaplari")
         )
         self.assertIn("çocuk kitapları", self.driver.title.lower(), "FAIL: Çocuk Kitapları kategorisi
   filtrelenmedi.")
         print("PASS: 'Çocuk Kitapları' kategorisi seçimi başarılı, ilgili sayfa yüklendi.")
except Exception as e:            self.fail(f"FAIL: 'Çocuk Kitapları' kategorisi linki bulunamadı
veya tıklanamadı: {e}")        print("Test Case 5: Category Selection Dropdown - Valid -
BAŞARILI")


   def test_category_invalid_no_selection(self):        print("\n--- Test Case 5: Category
Selection Dropdown - Invalid (No Selection) ---")
```

```
        self.driver.get(BASE_URL)
WebDriverWait(self.driver, 10).until(
EC.presence_of_element_located((By.ID, "search-input"))
        )

        self.assertNotIn("çocuk kitapları", self.driver.title.lower(), "FAIL: Kategori seçilmediği
    halde 'Çocuk Kitapları' başlığı görünüyor.")
        print("PASS: Kategori seçilmediğinde, belirli bir kategoriye özel filtre uygulanmadı.")
print("Test Case 5: Category Selection Dropdown - Invalid (No Selection) - BAŞARILI")

    def test_category_invalid_disabled(self):         print("\n--- Test Case 5: Category
Selection Dropdown - Invalid (Disabled) ---")

try:
            self.driver.find_element(By.XPATH, "//a[text()='Geçersiz
Kategori']").click()             self.fail("FAIL: Var olmayan bir kategoriye
tıklanabildi.")         except:
            print("PASS: Var olmayan bir kategoriye tıklanamadı (element bulunamadı veya
    etkileşim engellendi).")
        print("Test Case 5: Category Selection Dropdown - Invalid (Disabled/Non-existent) -
    BAŞARILI")


if __name__ == "__main__":
unittest.main()




import unittest
from selenium import webdriver from
selenium.webdriver.chrome.service import Service from
selenium.webdriver.common.by import By from
selenium.webdriver.common.keys import Keys from
selenium.webdriver.support.ui import WebDriverWait from
selenium.webdriver.support import expected_conditions as EC
import time import os
from config import CHROMEDRIVER_PATH, BASE_URL

class TestUserRegistrationEmailValidation(unittest.TestCase):

    def setUp(self):
        service = Service(executable_path=CHROMEDRIVER_PATH)
        self.driver = webdriver.Chrome(service=service)
self.driver.maximize_window()
```

```
        # Doğrudan verilen "üye-ol" URL'sini kullanıyoruz
        self.driver.get("https://www.kitapyurdu.com/index.php?route=account/register")

        # Email input'u ve password input'u yüklenene kadar bekleyelim
        WebDriverWait(self.driver, 10).until(
            EC.presence_of_element_located((By.ID, "email"))
        )
        self.email_input = self.driver.find_element(By.ID, "email")
self.password_input = self.driver.find_element(By.ID, "password")

        self.email_error_msg_locator = (By.CSS_SELECTOR, "div[data-valid-id='email']")


    def tearDown(self):
self.driver.quit()

    def test_email_valid_format(self):          print("\n--- Test Case 2: User Registration –
Email Format Validation - Valid ---")          self.email_input.clear()
        self.email_input.send_keys("user@example.com")
        self.password_input.click()
time.sleep(0.5)

try:
            WebDriverWait(self.driver, 2).until_not(
                EC.visibility_of_element_located(self.email_error_msg_locator)
            )
            print("PASS: Geçerli email kabul edildi, hata mesajı
yok.")          except Exception as e:                self.fail(f"FAIL:
Geçerli email için hata mesajı çıktı: {e}")
        print("Test Case 2: User Registration – Email Format Validation - Valid - BAŞARILI")     def
test_email_invalid_no_at_symbol(self):
        print("\n--- Test Case 2: User Registration – Email Format Validation - Invalid (No '@') ---
    ")
self.email_input.clear()
        self.email_input.send_keys("ilaydaexample.com")
        self.password_input.click()
        time.sleep(1)

        error_message_element = WebDriverWait(self.driver, 5).until(
EC.visibility_of_element_located(self.email_error_msg_locator)
        )
        error_message = error_message_element.text
        self.assertIn("Geçerli bir E-Posta adresi yazınız!", error_message, "FAIL: Email format
    hatası mesajı bulunamadı veya farklı.")
        print("PASS: '@' sembolü eksik email için doğru hata mesajı gösterildi.")
        print("Test Case 2: User Registration – Email Format Validation - Invalid (No '@') -
    BAŞARILI")
```

```
   def test_email_invalid_no_domain(self):         print("\n--- Test Case 2: User Registration –
Email Format Validation - Invalid (No domain)
   ---")
self.email_input.clear()
      self.email_input.send_keys("test@")
self.password_input.click()
time.sleep(1)

     error_message_element = WebDriverWait(self.driver, 5).until(
EC.visibility_of_element_located(self.email_error_msg_locator)
      )
      error_message = error_message_element.text
      self.assertIn("Geçerli bir E-Posta adresi yazınız!", error_message, "FAIL: Email format
   hatası mesajı (domain eksik) bulunamadı.")
      print("PASS: Domain eksik email için doğru hata mesajı gösterildi.")
      print("Test Case 2: User Registration – Email Format Validation - Invalid (No domain) -
   BAŞARILI")


   def test_email_invalid_empty(self):         print("\n--- Test Case 2: User Registration – Email
Format Validation - Invalid (Empty) ---")         self.email_input.clear()
self.password_input.click()         time.sleep(1)

      error_message_element = WebDriverWait(self.driver, 5).until(
EC.visibility_of_element_located(self.email_error_msg_locator)
      )
      error_message = error_message_element.text
      self.assertIn("Bu alan zorunludur!", error_message, "FAIL: Boş email alanı hata mesajı
   bulunamadı veya farklı.")
      print("PASS: Boş email alanı için doğru hata mesajı gösterildi.")
      print("Test Case 2: User Registration – Email Format Validation - Invalid (Empty) -
   BAŞARILI")


if __name__ == "__main__":
unittest.main()
```

```
import org.junit.jupiter.api.*; import
org.openqa.selenium.*; import
org.openqa.selenium.chrome.ChromeDriver; import
org.openqa.selenium.support.ui.ExpectedConditions;
import org.openqa.selenium.support.ui.WebDriverWait;

import java.time.Duration;

import static org.junit.jupiter.api.Assertions.*;

public class InputLengthTest {

    private WebDriver driver;

    @BeforeEach
void setUp() {
        driver = new ChromeDriver();
```

```
        driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(5));
driver.manage().window().maximize();
        driver.get("https://www.kitapyurdu.com");
    }

    @AfterEach
void tearDown() {
driver.quit();
    }

    public void search(String text) {
        WebElement input = driver.findElement(By.id("search-
input"));        input.clear();        input.sendKeys(text);
        input.sendKeys(Keys.RETURN);
    }

    boolean isSearchPerformed() {
        String currentUrl = driver.getCurrentUrl();
        return !currentUrl.equals("https://www.kitapyurdu.com/");
    }

    boolean isResultFound() {
if (!isSearchPerformed()) {
            return false;
}
        WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(5));
try {
            wait.until(ExpectedConditions.presenceOfElementLocated(By.className("productcr")));
            return true;
        } catch (TimeoutException e) {
return false;
        }
    }



    @Test
    void testE1_TooShort_NoResult() {
        boolean searches = searchFor0Len() || searchFor1Len();
assertFalse(searches, "Expected no result for short input");
    }

    @Test
    void testE2_ValidLength_RelatedResults() {
        boolean searches = searchFor2Len() && searchFor99Len() && searchFor100Len();
assertTrue(searches, "Expected related results for valid input");
    }
```

```
    //It is impossible to do a search query with negative input because a String cannot has negative
    length.
    //So the testU1_NegativeLength_Impossible func is a kind of placeholder for negative
valued search query.     @Test
    void testU1_NegativeLength_Impossible() {
search("");
        assertFalse(isResultFound(), "Expected no result for empty string (interpreted as
    impossible)");
    }


    @Test     void
testU2_TooLong_ErrorMessage() {
String longQuery = "a".repeat(165);
search(longQuery);

        WebElement body = driver.findElement(By.tagName("body"));
        assertTrue(body.getText().contains("hata") || !doesInputLengthMatch(165),
"Expected an error or no result for overly long input");
    }



    private boolean searchFor2Len() {
        search("ab");
        return isResultFound() && doesInputLengthMatch(2);
    }

    private boolean searchFor1Len() {
        search("a");
        return isResultFound();
    }

    private boolean searchFor0Len() {
search("");
        return isResultFound();
    }

    private boolean searchFor99Len() {
String query = "a".repeat(99);
search(query);
        return doesInputLengthMatch(99);
    }
```

```
    private boolean searchFor100Len() {
String query = "a".repeat(100);
search(query);
        return doesInputLengthMatch(100);
    }

    private boolean doesInputLengthMatch(int inputLen){
        WebElement searchInput = driver.findElement(By.id("search-input"));
        String inputValue =
searchInput.getAttribute("value");         int length =
inputValue.length();         return length == inputLen;
    }


}
```

```
import org.junit.jupiter.api.*;
import org.openqa.selenium.*;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.support.ui.*;

import java.time.Duration;

public class MaxPriceHorrorTest {
```

```
    private static WebDriver driver;
private static WebDriverWait wait;
    private static final int WAIT_TIMEOUT_SECONDS = 10;

    @BeforeAll
    public static void setup() {
driver = new ChromeDriver();
        wait = new WebDriverWait(driver, Duration.ofSeconds(WAIT_TIMEOUT_SECONDS));
driver.manage().window().maximize();
    }

    @BeforeEach
    public void goToHorrorGenre() {

     driver.get("https://www.kitapyurdu.com/index.php?route=product/search&filter_name=korku
    ");
wait.until(ExpectedConditions.urlContains("filter_name=korku"));
        wait.until(ExpectedConditions.presenceOfElementLocated(By.cssSelector(".product-cr")));
    }

    @AfterAll
    public static void teardown() {
        if (driver != null) {
driver.quit();
        }
    }

    private void setMaxPrice(String maxPrice) {
        WebElement maxPriceInput = findMaxPriceInput();

        scrollToElement(maxPriceInput);
        clearAndSetPrice(maxPriceInput, maxPrice);

        waitForLoadingToDisappear();
waitForResultsToLoad();
    }

    private WebElement findMaxPriceInput() {
try {
            return
    wait.until(ExpectedConditions.visibilityOfElementLocated(By.id("selected_sell_price_max"))
    );
        } catch (TimeoutException e) {
            return
    wait.until(ExpectedConditions.visibilityOfElementLocated(By.name("selected_sell_price_ma
    x")));
        }
```

```
    }

    private void scrollToElement(WebElement element) {
        ((JavascriptExecutor) driver).executeScript("arguments[0].scrollIntoView(true);", element);
    }

    private void clearAndSetPrice(WebElement maxPriceInput, String maxPrice) {
maxPriceInput.clear();
        String formattedPrice = maxPrice.replace('.', ',');
maxPriceInput.sendKeys(formattedPrice);
maxPriceInput.sendKeys(Keys.RETURN);
    }

    private void waitForLoadingToDisappear() {
try {
            wait.until(ExpectedConditions.invisibilityOfElementLocated(By.className("loading")));
} catch (TimeoutException ignored) {
        }
    }

    private void waitForResultsToLoad() {
        wait.until(driver -> driver.findElements(By.cssSelector(".product-cr")).size() > 0 ||
driver.findElements(By.cssSelector(".no-result")).size() > 0);
    }

    private int getResultCount() {
        WebElement productsCountElement = wait.until(
                ExpectedConditions.visibilityOfElementLocated(By.cssSelector("#faceted-search-listtotal
    h2"))
        );

        String text = productsCountElement.getText();
        System.out.println("Product count text: " + text);

        java.util.regex.Pattern pattern = java.util.regex.Pattern.compile("(\\d+)");
java.util.regex.Matcher matcher = pattern.matcher(text);

        if (matcher.find()) {
            return Integer.parseInt(matcher.group(1));
        } else {
            System.out.println("No number found in product count text");
return 0;
        }
    }

    private String getMaxPriceInputValue() {
java.util.List<WebElement> elements =
```

```
    driver.findElements(By.id("selected_sell_price_max"));
        if (elements.isEmpty()) {
            return null;
        }
        return elements.get(0).getAttribute("value");
    }

    @Test
    public void testE1_MaxPriceWithinRange() {
setMaxPrice("1");
        Assertions.assertEquals(19, getResultCount());

        setMaxPrice("2");
        Assertions.assertEquals(20, getResultCount());

        setMaxPrice("958");
        Assertions.assertEquals(492, getResultCount());

        setMaxPrice("958.40");
        Assertions.assertEquals(493, getResultCount());
    }

    @Test
    public void testE2_MaxPriceAboveRange() {
setMaxPrice("986");
        Assertions.assertEquals(493, getResultCount());

        setMaxPrice("999999999999999999998");
Assertions.assertEquals(493, getResultCount());

        setMaxPrice("9999999999999999999999");
Assertions.assertEquals(493, getResultCount());
    }

    @Test
    public void testU1_NegativeMaxPrice_Error() {
String price = "-5";
        setMaxPrice(price);

        wait.until(driver -> {
            String currentValue = getMaxPriceInputValue();
            return currentValue == null || !price.equals(currentValue);
        });

        String currentValue = getMaxPriceInputValue();
        if (currentValue == null) {
Assertions.assertTrue(true);
```

```
        } else {
            Assertions.assertFalse(price.equals(currentValue),
                    "Expected the input to reject the negative price and change its value");
            Assertions.assertEquals("5", currentValue,
                    "Expected the input value to be corrected to 5");
        }
    }



    @Test
    public void testU2_TooLargeMaxPrice_TypeError() {
String price = "10000000000000000000000";
        setMaxPrice(price);

        wait.until(driver -> {
            String currentValue = getMaxPriceInputValue();
return currentValue == null || !price.equals(currentValue);
        });

        String currentValue = getMaxPriceInputValue();
if (currentValue == null) {
Assertions.assertTrue(true);
        } else {
            Assertions.assertFalse(price.equals(currentValue),
                    "Expected the input to reject the too large price and change its value");
        }
    }

}
```


```
{
 "id": "eba80701-4127-4b7a-bdf7-0078f29daaf0",
 "version": "2.0",
 "name": "KitapYurdu",
 "url": "https://www.kitapyurdu.com",
 "tests": [{
  "id": "0630cda5-94c7-4955-bd9d-34279ad88b2e",
  "name": "Untitled",
  "commands": [{
   "id": "18f74ba8-737a-4742-8562-678d471a86f8",
   "comment": "",
   "command": "open",
   "target": "/index.php?route=account/account",
   "targets": [],
   "value": ""
  }, {
```

```
  "id": "f8ebe3f7-b545-437a-a6a3-2daa8c933b6a",
  "comment": "",
  "command": "setWindowSize",
  "target": "1303x961",
  "targets": [],
  "value": ""
}, {
  "id": "5424b53e-6873-4b29-8878-6706c62e379b",
  "comment": "",
  "command": "click",
  "target": "id=search-input",
  "targets": [
    ["id=search-input", "id"],
    ["name=search_keyword", "name"],
    ["css=#search-input", "css:finder"],
    ["xpath=//input[@id='search-input']", "xpath:attributes"],
    ["xpath=//div[@id='search']/input", "xpath:idRelative"],
    ["xpath=//div[3]/input", "xpath:position"]
  ],
  "value": ""
}, {
  "id": "806b1fd3-1a66-44c0-829a-aaf7c575d632",
  "comment": "",
  "command": "type",
  "target": "id=search-input",
  "targets": [
    ["id=search-input", "id"],
    ["name=search_keyword", "name"],
    ["css=#search-input", "css:finder"],
    ["xpath=//input[@id='search-input']", "xpath:attributes"],
    ["xpath=//div[@id='search']/input", "xpath:idRelative"],
    ["xpath=//div[3]/input", "xpath:position"]
  ],
  "value": "nutuk"
```

```
}, {
  "id": "c218a1b3-23e6-4334-b7c3-92b0e21379ce",
  "comment": "",
  "command": "sendKeys",
  "target": "id=search-input",
  "targets": [
    ["id=search-input", "id"],
    ["name=search_keyword", "name"],
    ["css=#search-input", "css:finder"],
    ["xpath=//input[@id='search-input']", "xpath:attributes"],
    ["xpath=//div[@id='search']/input", "xpath:idRelative"],
    ["xpath=//div[3]/input", "xpath:position"]
  ],
  "value": "${KEY_ENTER}"
}, {
  "id": "c66b609c-6b53-45da-9972-6795cac2a2ba",
  "comment": "",
  "command": "click",
  "target": "css=#product-645662 > .grid_7 > .name span",
  "targets": [
    ["css=#product-645662 > .grid_7 > .name span", "css:finder"],
    ["xpath=//div[@id='product-645662']/div/div[2]/a/span", "xpath:idRelative"],
    ["xpath=//div[2]/div/div[2]/a/span", "xpath:position"]
  ],
  "value": ""
}, {
  "id": "fcd1d8d9-6c3b-4e3d-bb1a-90c145410e75",
  "comment": "",
  "command": "click",
  "target": "css=.icon__cta-fast",
  "targets": [
    ["css=.icon__cta-fast", "css:finder"],
    ["xpath=//a[@id='button-cart']/i", "xpath:idRelative"],
    ["xpath=//div[3]/a/i", "xpath:position"]
  ],
  "value": ""
}, {
  "id": "11c0f1b2-c377-496b-857e-32d99fce7c2f",
  "comment": "",
  "command": "click",
```

```
    }, {
      "target": "id=cart-items",
      "targets": [
       ["id=cart-items", "id"],
       ["css=#cart-items", "css:finder"],
       ["xpath=//span[@id='cart-items']", "xpath:attributes"],
       ["xpath=//div[@id='sprite-cart-icon']/span", "xpath:idRelative"],
       ["xpath=//div[4]/div/div/span", "xpath:position"],
       ["xpath=//span[contains(.,'1')]", "xpath:innerText"]
      ],
      "value": ""
      "id": "acb45e5e-d917-42a4-9b31-0a464a05c592",
      "comment": "",
      "command": "click",
      "target": "id=js-cart",
      "targets": [
       ["id=js-cart", "id"],
       ["linkText=Go to Cart", "linkText"],
       ["css=#js-cart", "css:finder"],
       ["xpath=//a[contains(text(),'Go to Cart')]", "xpath:link"],
       ["xpath=//a[@id='js-cart']", "xpath:attributes"],
       ["xpath=//div[@id='cart-content']/div[2]/a", "xpath:idRelative"],
["xpath=//a[contains(@href,
    'https://www.kitapyurdu.com/index.php?route=checkout/cart')]", "xpath:href"],
       ["xpath=//div[2]/div[2]/a", "xpath:position"],
       ["xpath=//a[contains(.,'Go to Cart')]", "xpath:innerText"]
      ],
      "value": ""
    }, {
      "id": "36c3c698-14ab-4865-aebb-7979d7b19987",
      "comment": "",
      "command": "click",
      "target": "linkText=Checkout",
      "targets": [
       ["linkText=Checkout", "linkText"],
       ["css=.py-12px", "css:finder"],
       ["xpath=//a[contains(text(),'Checkout')]", "xpath:link"],
       ["xpath=//div[@id='cart-sidebar']/div/div[2]/a", "xpath:idRelative"],
["xpath=//a[contains(@href,
    'https://www.kitapyurdu.com/index.php?route=checkout/delivery')]", "xpath:href"],
```

```
    }, {
      ["xpath=//div[2]/div/div[2]/div/div[2]/a", "xpath:position"],
      ["xpath=//a[contains(.,'Checkout')]", "xpath:innerText"]
     ],
     "value": ""
    }, {
     "id": "fbd4c670-5274-4b3f-9c9a-ca956146a016",
     "comment": "",
     "command": "click",
     "target": "linkText=Add New",
     "targets": [
      ["linkText=Add New", "linkText"],
      ["css=.mb-10px > .text-green-dark", "css:finder"],
      ["xpath=(//a[contains(text(),'Add New')])[2]", "xpath:link"],
      ["xpath=//div[@id='collapseShippingAddress']/div/div/div[3]/a[2]", "xpath:idRelative"],
["xpath=(//a[contains(@href,
    'https://www.kitapyurdu.com/index.php?route=account/address/insert&is_checkout=1')])[2]",
    "xpath:href"],
      ["xpath=//div[3]/a[2]", "xpath:position"]
     ],
     "value": ""
     "id": "38c36515-23f1-4ecb-a314-6000fdc9e0db",
     "comment": "",
     "command": "click",
     "target": "id=title",
     "targets": [
      ["id=title", "id"],
      ["name=name", "name"],
      ["css=#title", "css:finder"],
      ["xpath=//input[@id='title']", "xpath:attributes"],
      ["xpath=//form[@id='js-address-insert-form']/div/div/div/input", "xpath:idRelative"],
      ["xpath=//form/div/div/div/input", "xpath:position"]
     ],
     "value": ""
    }, {
     "id": "9f054f40-4d6d-4eca-9c8b-895ebdd42f43",
     "comment": "",
     "command": "type",
     "target": "id=title",
     "targets": [
```

```
}, {
    ["id=title", "id"],
    ["name=name", "name"],
    ["css=#title", "css:finder"],
    ["xpath=//input[@id='title']", "xpath:attributes"],
    ["xpath=//form[@id='js-address-insert-form']/div/div/div/input", "xpath:idRelative"],
    ["xpath=//form/div/div/div/input", "xpath:position"]
  ],
  "value": "okul"
}, {
  "id": "8b45b23c-b2c5-4a70-a6c6-3925f32a5a91",
  "comment": "",
  "command": "click",
  "target": "id=firstname_companyname",
  "targets": [
    ["id=firstname_companyname", "id"],
    ["css=#firstname_companyname", "css:finder"],
    ["xpath=//input[@id='firstname_companyname']", "xpath:attributes"],
    ["xpath=//form[@id='js-address-insert-form']/div/div/div[2]/input", "xpath:idRelative"],
    ["xpath=//div[2]/input", "xpath:position"]
  ],
  "value": ""
}, {
  "id": "6e1063d7-5a24-48d3-9fba-c9c6fe9eace7",
  "comment": "",
  "command": "type",
  "target": "id=firstname_companyname",
  "targets": [
    ["id=firstname_companyname", "id"],
    ["css=#firstname_companyname", "css:finder"],
    ["xpath=//input[@id='firstname_companyname']", "xpath:attributes"],
```

```
    ["xpath=//form[@id='js-address-insert-form']/div/div/div[2]/input", "xpath:idRelative"],
    ["xpath=//div[2]/input", "xpath:position"]
   ],
   "value": "okul1"
  }, {
   "id": "57a366b4-2b2f-4e24-854a-7c34b1bb35a7",
   "comment": "",
   "command": "click",
   "target": "id=lastname_title",
   "targets": [
    ["id=lastname_title", "id"],
    ["name=surname", "name"],
    ["css=#lastname_title", "css:finder"],
    ["xpath=//input[@id='lastname_title']", "xpath:attributes"],
    ["xpath=//form[@id='js-address-insert-form']/div/div/div[3]/input", "xpath:idRelative"],
    ["xpath=//form/div/div/div[3]/input", "xpath:position"]
   ],
   "value": ""
  }, {
   "id": "e687056e-1e66-4fb7-b13c-4e47ee3c5cfc",
   "comment": "",
   "command": "type",
   "target": "id=lastname_title",
   "targets": [
    ["id=lastname_title", "id"],
    ["name=surname", "name"],
    ["css=#lastname_title", "css:finder"],
    ["xpath=//input[@id='lastname_title']", "xpath:attributes"],
    ["xpath=//form[@id='js-address-insert-form']/div/div/div[3]/input", "xpath:idRelative"],
    ["xpath=//form/div/div/div[3]/input", "xpath:position"]
   ],
   "value": "okul2"
  }, {
   "id": "abe2ff80-34b1-4cdd-9915-5a09484b484f",
   "comment": "",
   "command": "runScript",
   "target": "window.scrollTo(0,231)",
   "targets": [],
   "value": ""
  }, {
   "id": "0cf1422d-6221-4acb-bdb3-4e805b19ff8c",
   "comment": "",
   "command": "click",
   "target": "id=btn-modal-zone",
   "targets": [
    ["id=btn-modal-zone", "id"],
    ["linkText=Select City", "linkText"],
```

```
    ["css=#btn-modal-zone", "css:finder"],
    ["xpath=//a[@id='btn-modal-zone']", "xpath:attributes"],
    ["xpath=//form[@id='js-address-insert-form']/div/div/div[5]/a", "xpath:idRelative"],
                          ["xpath=//form/div/div/div[5]/a", "xpath:position"],
    ["xpath=//a[contains(.,'Select City')]", "xpath:innerText"]
   ],
   "value": ""
  }, {
   "id": "59472412-6615-4f5b-93c8-82e5ee162fde",
   "comment": "",
   "command": "click",
   "target": "css=#js-list-zone > .js-list-group-item:nth-child(7)",
   "targets": [
    ["css=#js-list-zone > .js-list-group-item:nth-child(7)", "css:finder"],
    ["xpath=//ul[@id='js-list-zone']/li[7]", "xpath:idRelative"],
    ["xpath=//div[5]/div[3]/div/div/div[2]/ul/li[7]", "xpath:position"],
    ["xpath=//li[contains(.,'Ankara')]", "xpath:innerText"]
   ],
   "value": ""
  }, {
   "id": "f4d3ac07-7ad5-4e67-997b-be2cd186503b",
   "comment": "",
   "command": "click",
   "target": "id=btn-modal-county",
   "targets": [
    ["id=btn-modal-county", "id"],
    ["linkText=Select County", "linkText"],
    ["css=#btn-modal-county", "css:finder"],
    ["xpath=//a[@id='btn-modal-county']", "xpath:attributes"],
    ["xpath=//div[@id='county']/a", "xpath:idRelative"],
    ["xpath=//form/div/div/div[6]/a", "xpath:position"],
    ["xpath=//a[contains(.,'Select County')]", "xpath:innerText"]
   ],
   "value": ""
  }, {
   "id": "4032c989-19f0-48f9-b103-cca146cb3d52",
   "comment": "",
   "command": "click",
   "target": "css=#js-list-county > .js-list-group-item:nth-child(9)",
   "targets": [
    ["css=#js-list-county > .js-list-group-item:nth-child(9)", "css:finder"],
    ["xpath=//ul[@id='js-list-county']/li[9]", "xpath:idRelative"],
    ["xpath=//div[6]/div[3]/div/div/div[2]/ul/li[9]", "xpath:position"],
    ["xpath=//li[contains(.,'ELMADAĞ')]", "xpath:innerText"]
   ],
   "value": ""
  }, {
```

```
  "id": "53048e38-7e03-40ef-9570-01d09fe807eb",
  "comment": "",
  "command": "mouseDownAt",
  "target": "id=btn-modal-district",
  "targets": [
   ["id=btn-modal-district", "id"],
   ["linkText=Select District", "linkText"],
   ["css=#btn-modal-district", "css:finder"],
   ["xpath=//a[@id='btn-modal-district']", "xpath:attributes"],
   ["xpath=//div[@id='district']/a", "xpath:idRelative"],
   ["xpath=//form/div/div/div[7]/a", "xpath:position"],
   ["xpath=//a[contains(.,'Select District')]", "xpath:innerText"]
  ],
  "value": "158.25,13"
}, {
  "id": "9bc4916e-e59a-45bb-9ca6-1e7843bbd925",
  "comment": "",
  "command": "mouseMoveAt",
  "target": "id=btn-modal-district",
  "targets": [
   ["id=btn-modal-district", "id"],
   ["linkText=Select District", "linkText"],
   ["css=#btn-modal-district", "css:finder"],
   ["xpath=//a[@id='btn-modal-district']", "xpath:attributes"],
   ["xpath=//div[@id='district']/a", "xpath:idRelative"],
   ["xpath=//form/div/div/div[7]/a", "xpath:position"],
   ["xpath=//a[contains(.,'Select District')]", "xpath:innerText"]
  ],
  "value": "158.25,13"
}, {
  "id": "b208b1ce-5fa0-4d18-9382-fe1112bd58a5",
  "comment": "",
  "command": "mouseUpAt",
  "target": "id=btn-modal-district",
  "targets": [
   ["id=btn-modal-district", "id"],
   ["linkText=Select District", "linkText"],
   ["css=#btn-modal-district", "css:finder"],
   ["xpath=//a[@id='btn-modal-district']", "xpath:attributes"],
   ["xpath=//div[@id='district']/a", "xpath:idRelative"],
   ["xpath=//form/div/div/div[7]/a", "xpath:position"],
   ["xpath=//a[contains(.,'Select District')]", "xpath:innerText"]
  ],
  "value": "158.25,13"
}, {
  "id": "99fff744-c9ba-4d19-a406-15045a864afb",
  "comment": "",
```

```
  "command": "click",
  "target": "id=btn-modal-district",
  "targets": [
   ["id=btn-modal-district", "id"],
   ["linkText=Select District", "linkText"],
   ["css=#btn-modal-district", "css:finder"],
   ["xpath=//a[@id='btn-modal-district']", "xpath:attributes"],
   ["xpath=//div[@id='district']/a", "xpath:idRelative"],
   ["xpath=//form/div/div/div[7]/a", "xpath:position"], ["xpath=//a[contains(.,'Select
   District')]", "xpath:innerText"]
  ],
  "value": ""
}, {
  "id": "4756505e-1000-46be-a29b-40699d56d78e",
  "comment": "",
  "command": "mouseDown",
  "target": "css=#js-list-district > .js-list-group-item:nth-child(12)",
  "targets": [
   ["css=#js-list-district > .js-list-group-item:nth-child(12)", "css:finder"],
   ["xpath=//ul[@id='js-list-district']/li[12]", "xpath:idRelative"],
   ["xpath=//div[7]/div[3]/div/div/div[2]/ul/li[12]", "xpath:position"],
   ["xpath=//li[contains(.,'LALABEL MAH')]", "xpath:innerText"]
  ],
  "value": ""
}, {
  "id": "6953aec2-76ca-40f6-8c4e-cf083abca713",
  "comment": "",
  "command": "mouseUp",
  "target": "css=#js-list-district > .js-list-group-item:nth-child(11)",
  "targets": [
   ["css=#js-list-district > .js-list-group-item:nth-child(11)", "css:finder"],
   ["xpath=//ul[@id='js-list-district']/li[11]", "xpath:idRelative"],
   ["xpath=//div[7]/div[3]/div/div/div[2]/ul/li[11]", "xpath:position"],
   ["xpath=//li[contains(.,'KUŞÇUALİ MAH')]", "xpath:innerText"]
  ],
  "value": ""
}, {
  "id": "c5152250-5132-4e4e-b70a-b4d47891788f",
  "comment": "",
  "command": "click",
  "target": "id=js-list-district",
  "targets": [
   ["id=js-list-district", "id"],
   ["css=#js-list-district", "css:finder"],
   ["xpath=//ul[@id='js-list-district']", "xpath:attributes"],
   ["xpath=//div[@id='modal-district']/div/div/div[2]/ul", "xpath:idRelative"],
   ["xpath=//div[7]/div[3]/div/div/div[2]/ul", "xpath:position"]
```

```
    ],
    "value": ""
   }, {
    "id": "413c4122-84d5-4faf-a8dc-5f99b58094af",
    "comment": "",
    "command": "click",
    "target": "css=#js-list-district > .js-list-group-item:nth-child(11)",
    "targets": [
     ["css=#js-list-district > .js-list-group-item:nth-child(11)", "css:finder"],
     ["xpath=//ul[@id='js-list-district']/li[11]", "xpath:idRelative"],
     ["xpath=//div[7]/div[3]/div/div/div[2]/ul/li[11]", "xpath:position"],
     ["xpath=//li[contains(.,'KUŞÇUALİ MAH')]", "xpath:innerText"]
    ],
    "value": ""
   }, {
    "id": "701ce644-8341-41c1-a29e-96c55c91671e",
    "comment": "",
    "command": "click",
    "target": "id=address",
    "targets": [
     ["id=address", "id"],
     ["css=#address", "css:finder"],
     ["xpath=//textarea[@id='address']", "xpath:attributes"],
     ["xpath=//form[@id='js-address-insert-form']/div/div/div[8]/textarea",
"xpath:idRelative"],      ["xpath=//textarea", "xpath:position"]
    ],
    "value": ""
   }, {
    "id": "ed068ba9-aee7-4c18-89bc-4d61dcfdb380",
    "comment": "",
    "command": "type",
    "target": "id=address",
    "targets": [
     ["id=address", "id"],
     ["css=#address", "css:finder"],
     ["xpath=//textarea[@id='address']", "xpath:attributes"],
     ["xpath=//form[@id='js-address-insert-form']/div/div/div[8]/textarea",
"xpath:idRelative"],
     ["xpath=//textarea", "xpath:position"]
    ],
    "value": "okul123"
   }, {
    "id": "6cf0ea2a-3113-45a7-ae19-4f27dc50561b",
    "comment": "",
    "command": "click",
    "target": "id=mobile_telephone",
    "targets": [
```

```
      ["id=mobile_telephone", "id"],
      ["name=mobile_telephone", "name"],
      ["css=#mobile_telephone", "css:finder"],
      ["xpath=//input[@id='mobile_telephone']", "xpath:attributes"],
      ["xpath=//form[@id='js-address-insert-form']/div/div/div[10]/input", "xpath:idRelative"],
      ["xpath=//div[10]/input", "xpath:position"]
     ],
     "value": ""
    }, {
     "id": "2232a371-c023-4641-a500-43e3b79d9aa3",
     "comment": "",
     "command": "type",
     "target": "id=mobile_telephone",
"targets": [
      ["id=mobile_telephone", "id"],
      ["name=mobile_telephone", "name"],
      ["css=#mobile_telephone", "css:finder"],
      ["xpath=//input[@id='mobile_telephone']", "xpath:attributes"],
      ["xpath=//form[@id='js-address-insert-form']/div/div/div[10]/input", "xpath:idRelative"],
      ["xpath=//div[10]/input", "xpath:position"]
     ],
     "value": "556 657 83 26"
    }, {
     "id": "d6463e71-28e7-4b29-a5f1-bdcc377ea296",
     "comment": "",
     "command": "click",
     "target": "id=telephone",
     "targets": [
      ["id=telephone", "id"],
      ["name=telephone", "name"],
      ["css=#telephone", "css:finder"],
      ["xpath=//input[@id='telephone']", "xpath:attributes"],
      ["xpath=//form[@id='js-address-insert-form']/div/div/div[11]/input", "xpath:idRelative"],
      ["xpath=//div[11]/input", "xpath:position"]
     ],
     "value": ""
    }, {
     "id": "a96a717a-a11a-4d16-accf-b13cd2fc4fa0",
     "comment": "",
     "command": "click",
     "target": "id=insert-address-button",
     "targets": [
      ["id=insert-address-button", "id"],
      ["css=#insert-address-button", "css:finder"],
      ["xpath=//button[@id='insert-address-button']", "xpath:attributes"],
      ["xpath=//form[@id='js-address-insert-form']/div[4]/button", "xpath:idRelative"],
      ["xpath=//div[4]/button", "xpath:position"]
```

```
  ],
  "value": ""
}, {
  "id": "1c7dc490-1ed1-4bf9-b188-e3a59944936d",
  "comment": "",
  "command": "click",
  "target": "id=non-23",
  "targets": [
    ["id=non-23", "id"],
    ["css=#non-23", "css:finder"],
    ["xpath=//input[@id='non-23']", "xpath:attributes"],
    ["xpath=//div[@id='shipping-company-non']/div/input", "xpath:idRelative"],
    ["xpath=//div[3]/div/div/div[2]/div/div/div/div[2]/div/input", "xpath:position"]
  ],
  "value": ""
}, {
  "id": "ead59330-040d-4abe-a9ae-0bf41c63617b",
  "comment": "",
  "command": "mouseOver",
  "target": "css=#headingCargo > .accordion-button",
  "targets": [
    ["css=#headingCargo > .accordion-button", "css:finder"],
    ["xpath=(//button[@type='button'])[17]", "xpath:attributes"],
    ["xpath=//h2[@id='headingCargo']/button", "xpath:idRelative"],
    ["xpath=//div[3]/div/div/div[2]/div/h2/button", "xpath:position"],
    ["xpath=//button[contains(.,'Shipping Selection')]", "xpath:innerText"]
  ],
  "value": ""
}, {
  "id": "adfe3796-0a7e-4e30-befb-3eb051de4bb4",
  "comment": "",
  "command": "click",
  "target": "id=continue-button",
  "targets": [
    ["id=continue-button", "id"],
    ["css=#continue-button", "css:finder"],
    ["xpath=//button[@id='continue-button']", "xpath:attributes"],
    ["xpath=//div[@id='checkout-sidebar']/div/div[2]/button", "xpath:idRelative"],
    ["xpath=//div[2]/button", "xpath:position"],
    ["xpath=//button[contains(.,'Continue')]", "xpath:innerText"]
  ],
  "value": ""
}, {
  "id": "c0b6507d-be00-4284-ac2f-5f67e5f8406e",
  "comment": "",
  "command": "click",
  "target": "id=credit_card_number",
```

```
  "targets": [
    ["id=credit_card_number", "id"],
    ["name=credit_card_number", "name"],
    ["css=#credit_card_number", "css:finder"],
    ["xpath=//input[@id='credit_card_number']", "xpath:attributes"],
    ["xpath=//div[@id='credit-card-form']/div/input", "xpath:idRelative"],
    ["xpath=//div/input", "xpath:position"]
  ],
  "value": ""
}, {
  "id": "49ac8431-1395-4319-a3ea-3548abcbcb09",
  "comment": "",
  "command": "type",
  "target": "id=credit_card_number",
  "targets": [
    ["id=credit_card_number", "id"],
    ["name=credit_card_number", "name"],
    ["css=#credit_card_number", "css:finder"],
    ["xpath=//input[@id='credit_card_number']", "xpath:attributes"],
    ["xpath=//div[@id='credit-card-form']/div/input", "xpath:idRelative"],
    ["xpath=//div/input", "xpath:position"]
  ],
  "value": "1231 2312 3123 1233"
}, {
  "id": "9bd9067f-01ca-426a-99c5-7ea4538cc6e7",
  "comment": "",
  "command": "click",
  "target": "id=credit_card_name",
  "targets": [
    ["id=credit_card_name", "id"],
    ["name=credit_card_name", "name"],
    ["css=#credit_card_name", "css:finder"],
    ["xpath=//input[@id='credit_card_name']", "xpath:attributes"],
    ["xpath=//div[@id='credit-card-form']/div[2]/input", "xpath:idRelative"],
    ["xpath=//div[2]/input", "xpath:position"]
  ],
  "value": ""
}, {
  "id": "2aac61ce-62a6-442f-82f7-b213ee14f74c",
  "comment": "",
  "command": "type",
  "target": "id=credit_card_name",
  "targets": [
    ["id=credit_card_name", "id"],
    ["name=credit_card_name", "name"],
    ["css=#credit_card_name", "css:finder"],
    ["xpath=//input[@id='credit_card_name']", "xpath:attributes"],
```

```
      ["xpath=//div[@id='credit-card-form']/div[2]/input", "xpath:idRelative"],
      ["xpath=//div[2]/input", "xpath:position"]
     ],
     "value": "pkuldfkd dwkıj"
    }, {
     "id": "6e52c52f-7fa0-49be-827c-33597c21201e",
     "comment": "",
     "command": "click",
     "target": "id=credit_card_expires",
     "targets": [
      ["id=credit_card_expires", "id"],
      ["name=credit_card_expires", "name"],
      ["css=#credit_card_expires", "css:finder"],
      ["xpath=//input[@id='credit_card_expires']", "xpath:attributes"],
      ["xpath=//div[@id='credit-card-form']/div[3]/div/input", "xpath:idRelative"],
["xpath=//div[3]/div/input", "xpath:position"]
     ],
     "value": ""
    }, {
     "id": "2b6406a6-8441-40f4-a777-4d4e5a10f3c8",
     "comment": "",
     "command": "type",
     "target": "id=credit_card_expires",
     "targets": [
      ["id=credit_card_expires", "id"],
      ["name=credit_card_expires", "name"],
      ["css=#credit_card_expires", "css:finder"],
      ["xpath=//input[@id='credit_card_expires']", "xpath:attributes"],
      ["xpath=//div[@id='credit-card-form']/div[3]/div/input", "xpath:idRelative"],
["xpath=//div[3]/div/input", "xpath:position"]
     ],
     "value": "12/31"
    }, {
     "id": "f56ee3cb-a931-4b24-82f6-9f452ec278dd",
     "comment": "",
     "command": "click",
     "target": "id=credit_card_security_code",
     "targets": [
      ["id=credit_card_security_code", "id"],
      ["name=credit_card_security_code", "name"],
      ["css=#credit_card_security_code", "css:finder"],
      ["xpath=//input[@id='credit_card_security_code']", "xpath:attributes"],
      ["xpath=//div[@id='credit-card-form']/div[3]/div[2]/div/input", "xpath:idRelative"],
["xpath=//div[2]/div/input", "xpath:position"]
     ],
     "value": ""
    }, {
```

```
      "id": "bc4fa955-9f9d-4188-9ed5-1cc579ce0bc8",
      "comment": "",
      "command": "type",
      "target": "id=credit_card_security_code",
      "targets": [
        ["id=credit_card_security_code", "id"],
        ["name=credit_card_security_code", "name"],
        ["css=#credit_card_security_code", "css:finder"],
        ["xpath=//input[@id='credit_card_security_code']", "xpath:attributes"],
        ["xpath=//div[@id='credit-card-form']/div[3]/div[2]/div/input", "xpath:idRelative"],
["xpath=//div[2]/div/input", "xpath:position"]
      ],
      "value": "123"
    }, {
      "id": "0ac8d09a-d6d2-4a9a-ab66-9bf64bb51cfb",
      "comment": "",
      "command": "click",
      "target": "id=continue-button",
      "targets": [
        ["id=continue-button", "id"],
        ["css=#continue-button", "css:finder"],
        ["xpath=//button[@id='continue-button']", "xpath:attributes"],
        ["xpath=//div[@id='checkout-sidebar']/div/div[2]/button", "xpath:idRelative"],
        ["xpath=//div[2]/div/div[2]/button", "xpath:position"],
        ["xpath=//button[contains(.,'Continue')]", "xpath:innerText"]
      ],
      "value": ""
    }]
  }],
  "suites": [{
    "id": "ef84ac61-b5f1-4000-a94d-14b546145035",
    "name": "Default Suite",
    "persistSession": false,
    "parallel": false,
    "timeout": 300,
    "tests": ["0630cda5-94c7-4955-bd9d-34279ad88b2e"]
  }],
  "urls": ["https://www.kitapyurdu.com/"],
  "plugins": []
}
```