

**1. Testing areas:**

1.1. What would be tested:

- Creation of a new account;
- Sign in form;
- Adding an item to the cart using search;
- Adding an item to the cart from page of current type of items;
- Basket basic features;

1.2. What would not be tested:

- Footer features;
- Header features;
- Home page features;
- All categories of items;
- Payment;
- Tracking order;

**2. Testing methodology:**

2.1. Approaches:

- Functional automation testing;
- Smoke testing;
- Negative testing;
- Black box testing techniques (equivalent partitioning, boundary values analysis, pairwise testing)
- UI testing;
- API testing.

2.2. Tools:

- JDK corretto-17 (java version "17.04.14");
- Gradle DSL (build-system);
- Selenium (UI);
- Rest-assured (API);
- Jackson (Serialization and deserialization);
- Lombok (code autogeneration using annotations);
- Owner (properties);
- Datafaker (test data);
- Allure (reports);
- GitHub Actions (CI/CD).

2.3. Environment:

- Production;
- Chrome (137.0.7151.120);
- Katana GF66 12UE (Intel(R) Core (TM) i5-12450H; NVIDIA GeForce RTX 3060 Laptop; 32.0 GB RAM) – for local run;
- Ubuntu (22.04/24.04), 4 vCPU, 16 GB RAM, 14 GB SSD – for CI/CD

**3. Tester – Volnov Alex.**

#### 4. Reasoning of testing:

##### 4.1. Risks:

- Defects in areas and features of the Website, which would not be tested;
- The integrational tests layer is represented only by API testing (without Database testing);
- Absence of Swagger or other documentation for API testing, due to which not all critical and important test scenarios can be identified;
- Limited ability to test certain critical scenarios due to restrictions imposed by the website's anti-bot protection system, which may block or interfere with automated test execution.

##### 4.2. Justification:

- the selected features ensure the fulfillment of the main objectives of the site;
- lack of sufficient human resources and time to conduct more in-depth testing;
- current product availability.

#### 5. Period of testing:

5.1. Entry Requirements - Prepared mind maps (to represent full scope of auto tests) and test tables (to track and manage the progress of test-coverage);

##### 5.2. Output requirements:

- successful review of all code branches with tests;
- successful completion of 90% of P0 tests (according to test tables) run locally and in CI/CD.

##### 5.3. Terms:

No	Stage	Terms
1.	API-testing	23.06.2025 - 29.06.2025
2.	UI-testing	30.06.2025 - 06.07.2025
3.	Run and debug tests in the CI/CD pipeline	07.07.2025 - 14.07.2025
4.	Designing the project and posting it to GitHub	15.07.2025 - 21.07.2025