

## Assignment 2

Software Quality Assurance and Testing | Tuseeva Inara

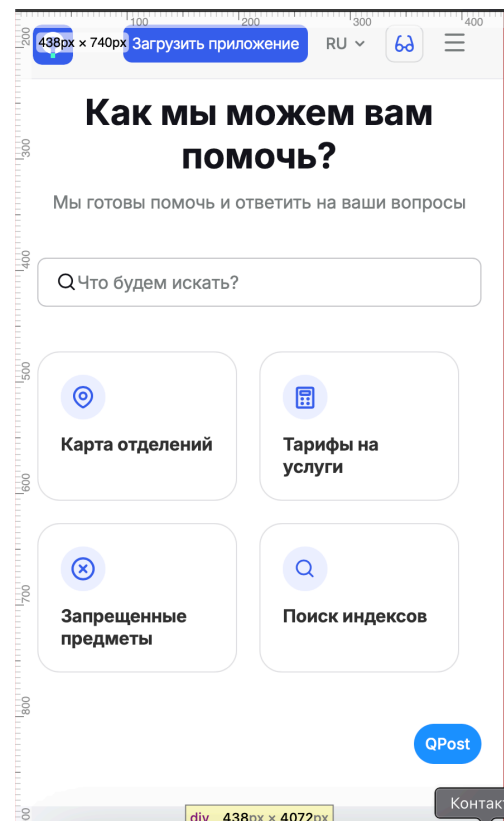
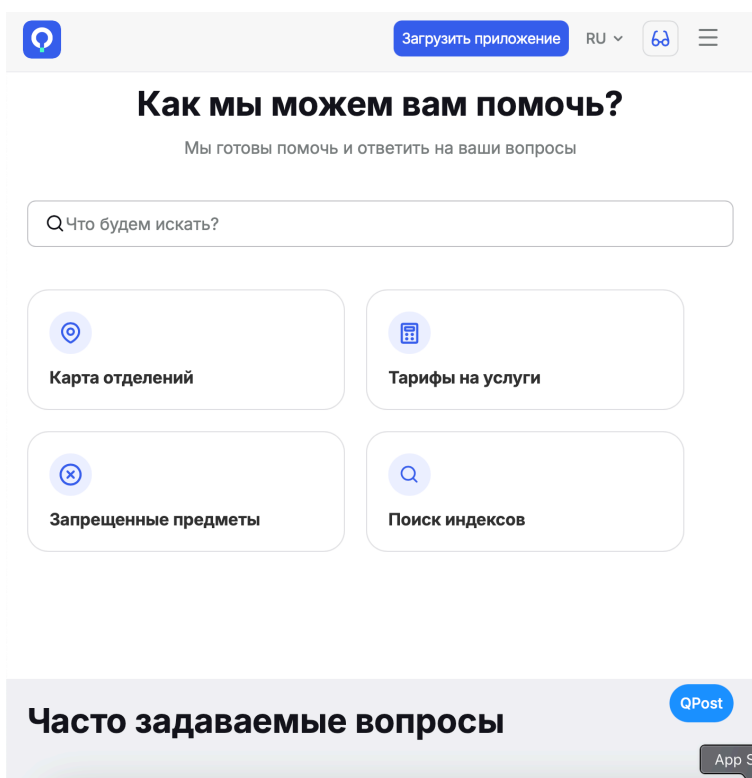
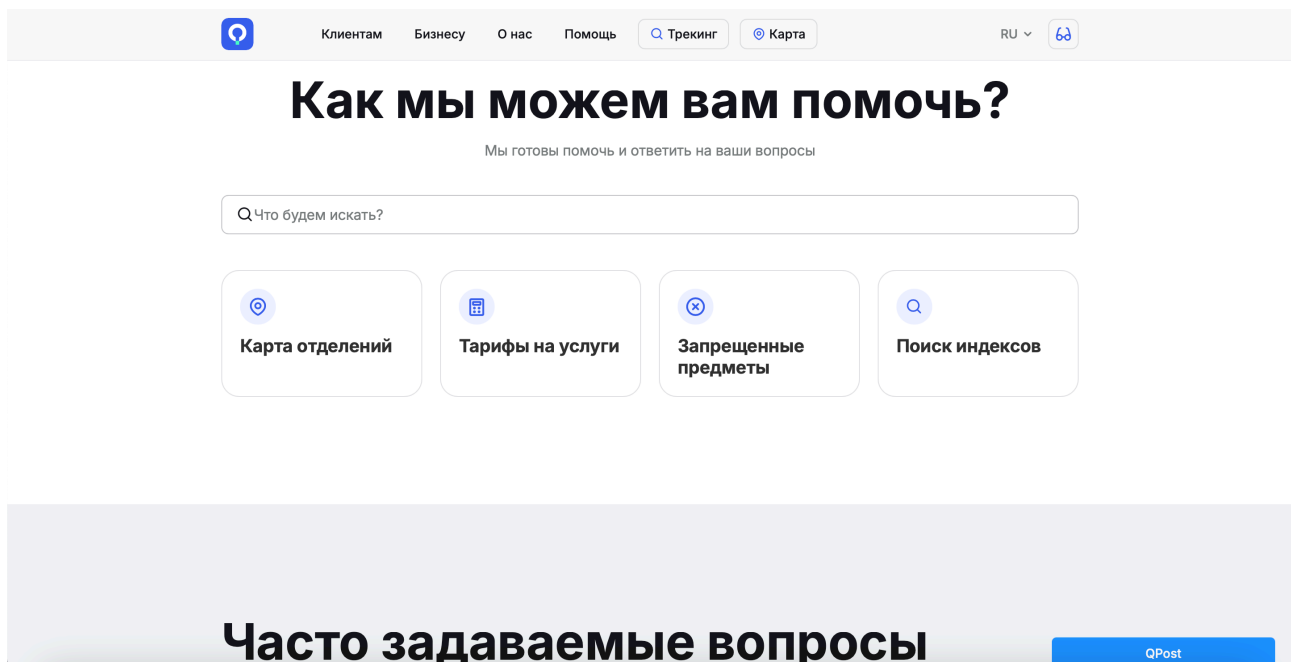
Zhaishylyk Rakhat, SE-2217, 221249

1)

Step 1: Access the Homepage

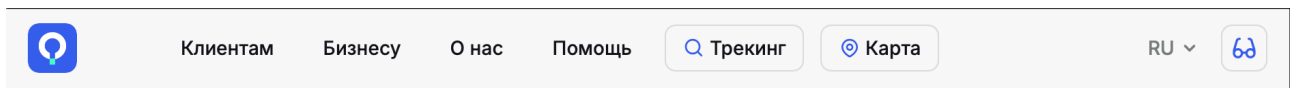
**Action:**

- Open the URL <https://www.kazpost.kz/en/> in a web browser.



### Checked:

- Ensure the homepage loads properly without errors.
- Verify the responsiveness of the design (check on both desktop and mobile views).
- Check if all main navigation elements are displayed.



### Expected Results:

- Homepage should load without any visual or functional issues.
- All navigation elements should be visible and clickable.

### Step 2: Language Selection

#### Action:

- Test the language switcher to ensure the site changes between available languages (e.g., English, Kazakh, Russian).

#### What to check:

- Verify that all content changes correctly without layout distortion.
- Ensure no untranslated sections remain after switching.

#### Expected Results:

- Content should appear fully translated and aligned correctly.

### Step 3: Search Functionality

#### Action:

- Use the search bar to look up common terms, such as "parcel tracking" or "postal services."

#### What to check:

- Verify that the search results are accurate and relevant.
- Test invalid inputs or edge cases, such as special characters, to check error handling.

#### Expected Results:

- Search results should match the query.

- Invalid inputs should display a proper error message or return "no results found."

#### **Step 4: Parcel Tracking**

##### **Action:**

- Click on the "Track Parcel" option and test tracking with:
  1. A valid tracking number.
  2. An invalid tracking number.
  3. A blank input.

##### **What to check:**

- Ensure that valid numbers return the correct details.
- Verify that invalid or blank inputs display appropriate error messages.
- Test loading speed and UI responsiveness during the query.

##### **Expected Results:**

- Correct parcel details for valid numbers.
- Proper error handling for invalid or blank inputs.

Not checked.

#### **Step 5: Navigation Links**

##### **Action:**

- Click on all main menu links.

##### **What to check:**

- Ensure the linked pages load without errors.
- Verify the functionality of buttons and hyperlinks on each page.
- Check the alignment and readability of content.

##### **Expected Results:**

- All links should navigate to the correct pages.
- Content should load correctly and be user-friendly.

#### **Step 6: User Account Login**

##### **Action:**

- Attempt to log in using:
  4. Valid credentials.
  5. Invalid credentials.
  6. Blank inputs.

## What to check:

- Test error messages and redirection for invalid credentials.
- Verify successful login and access to account details for valid credentials.
- Check for session timeouts or logout functionality.

## Expected Results:

- Smooth login process with valid credentials.
- Proper error handling for invalid inputs.

No Login Page found

## Step 7: Contact Page

### Action:

- Navigate to the "Contact Us" page and test the contact form by submitting:
  7. Valid information.
  8. Incomplete information.
  9. Invalid information (e.g., special characters in name fields).

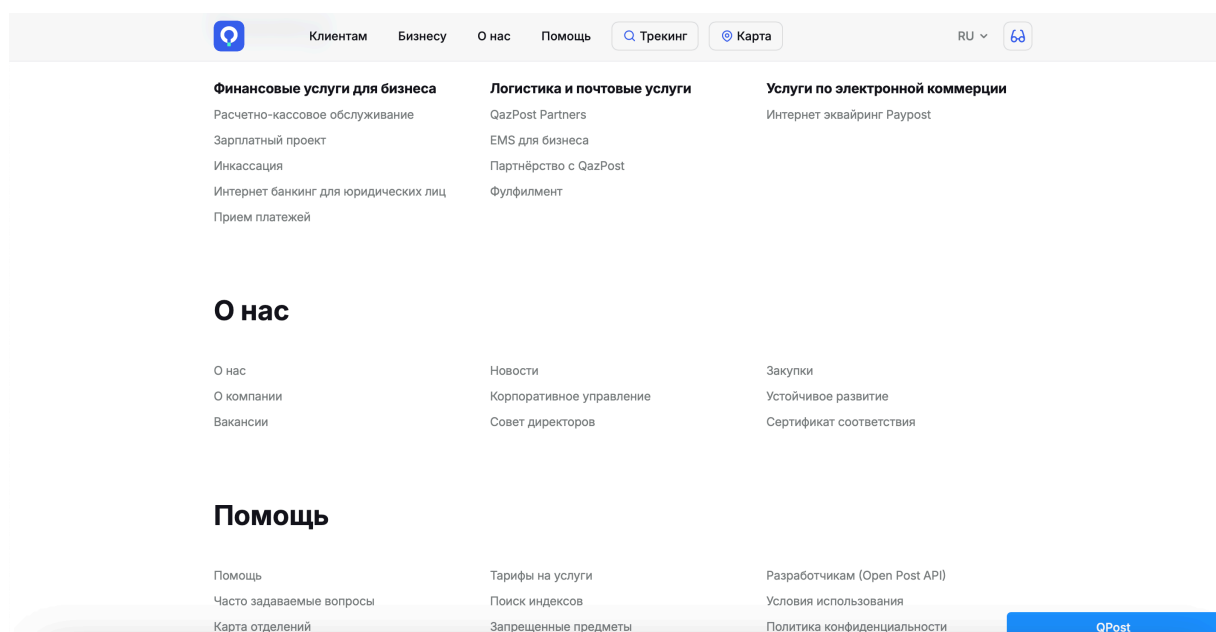
### What to check:

- Ensure successful submission with valid data.
- Verify error handling for incomplete or invalid inputs.
- Check if a confirmation message or email is received after submission.

## Expected Results:

- Successful submission for valid data.
- Proper error messages for invalid or incomplete entries.

## Step 8: Footer Links



**Action:**

- Click on all footer links (e.g., "Privacy Policy," "Terms of Use").

**What to check:**

- Verify that all links redirect to the correct pages.
- Check the readability and layout of these pages.

**Expected Results:**

- Footer links should function as expected without any errors.

**Step 9: Stress Testing****Action:**

- Open multiple tabs with the website and test navigation simultaneously.
- Test the website under slow internet conditions.

**What to check:**

- Ensure the website remains responsive and doesn't crash.
- Verify the loading time and error handling under low bandwidth.

**Expected Results:**

- Website should handle multiple tabs and slow connections gracefully.

## 2) Test Plan for <https://egov.kz/cms/kk>

### 1. Test Objectives

The purpose of testing is to ensure that the eGov.kz website meets both functional and non-functional requirements, providing a seamless user experience with accurate functionality and high performance.

- Functional Objectives:
  - Verify login and authentication mechanisms (user and admin roles).
  - Test the search functionality for locating government services.
  - Validate form submissions for requesting government services (e.g., applying for certificates).
  - Check the navigation and accessibility of major modules/pages.
- Non-Functional Objectives:
  - Ensure the website performs efficiently under high user loads.
  - Conduct security testing to prevent data breaches and unauthorized access.
  - Verify usability for users with diverse skill levels, including accessibility compliance.

### 2. Scope

- In-Scope:
  - Main modules/pages: Authentication, Search, Services Directory, Feedback Forms, and Payment System.
  - Responsive design on mobile and desktop browsers.
  - API integrations for service requests and payment processing.
- Out-of-Scope:
  - Internal administrative portals and back-end operations.

### 3. Schedule and Logistics

- Timeline:
  - Test Planning and Environment Setup: 1 week.
  - Functional Testing: 2 weeks.
  - Non-Functional Testing: 1 week.
  - Regression and Final Testing: 1 week.
- Resources Needed:
  - Human Resources: 2 QA engineers and 1 test manager.

- Technical Resources: Test devices (laptops, mobile phones, and tablets), virtual machines, network simulators, and testing tools (e.g., Selenium, Postman).

#### 4. Test Environment

- Hardware:
  - Desktops with Windows/Linux.
  - Smartphones with Android/iOS.
- Software:
  - Browsers: Chrome, Edge, Firefox, Safari.
  - Tools: Selenium for UI testing, JMeter for performance testing, and Burp Suite for security testing.
- Network Configurations:
  - High-speed internet for standard testing.
  - Simulated low bandwidth for performance testing.
- Dependencies:
  - Access to the production-like staging environment.
  - Test accounts with user roles and preconfigured API endpoints.

#### 5. Test Strategies/Methods

- Functional Testing:
  - Unit Testing: Verify individual components (e.g., login form, search bar).
  - Integration Testing: Test how modules interact, such as login integration with the user dashboard.
  - System Testing: End-to-end testing of user workflows.
- Non-Functional Testing:
  - Performance Testing: Test the website's behavior under load using JMeter.
  - Security Testing: Use tools like Burp Suite to check for vulnerabilities.
  - Usability Testing: Assess ease of use for diverse user groups.
- Tools and Techniques:
  - Selenium for automated UI testing.
  - Postman for API testing.
  - BrowserStack for cross-browser testing.

#### 6. Test Cases

Test Case	Test Objective	Pre-conditions	Test Steps	Expected Results	Actual	Status
TC001	Verify login functionality	User account	1. Navigate to the login page. 2. Enter valid credentials and submit.	User is redirected to the dashboard after valid login. Error message displayed for invalid login attempts.		
TC002	Validate search	Website homepage	1. Enter a valid search term and press Enter. 2. Enter invalid or blank terms and verify the	Accurate search results are displayed. Proper "no results found" message or error handling is		
TC003	Test form submission	User logged in	1. Navigate to any service form and fill in valid	Form submits successfully, and a confirmation message		

## 7. Bug Report

Bug ID	Description	Steps to Reproduce	Severity	Priority	Status
BUG001	Search bar crashes on special character input	1. Open the homepage. Enter special characters (e.g., @@@###) into the search bar.	Critical	High	Open

## 8. Risks

- Technical Risks:
  - API downtime during testing.
  - Browser compatibility issues.
- Resource Risks:
  - Insufficient QA resources for peak testing periods.
  - Limited access to real devices for responsive testing.

## 9. Procedures for Handling Problems

1. Bug Reporting:
  - Report issues via the bug-tracking tool (e.g., Jira).
  - Assign severity and priority levels for each issue.
2. Issue Resolution:
  - Collaborate with the development team to fix critical and high-priority bugs first.
3. Re-testing:
  - Perform retesting after fixes are implemented.
4. Regression Testing:



- Verify that fixes did not introduce new issues in the existing functionalities.

This structured plan ensures comprehensive coverage of the critical functionalities and features of the eGov.kz website, delivering a high-quality product for end-users.

3)

## Difference Between Smoke Testing and Regression Testing

Criteria	Smoke Testing	Regression Testing
<b>Purpose</b>	Verify critical functionalities after changes.	Ensure no new changes break existing functionality.
<b>Scope</b>	Focuses only on critical functionalities.	Covers the entire application or affected components.
<b>Test Type</b>	Quick, high-level verification.	Comprehensive and detailed testing.
<b>Result Analysis</b>	Identifies immediate blockers for a release.	Compares current and previous test results.
<b>Reporting</b>	Summary report on critical issues.	Detailed report on tests and issues.

### Smoke Test Suite for <https://gov.kz/>

Test Case	User Requirements Addressed	Expected Result	Actual	Pass/Fail	Status	Corrective Action if
Check Homepage	Homepage should load within 2 seconds.	Homepage loads completely without		Pass		Optimize assets for faster
Verify Search Bar	Users can search for services using	Results related to the search term are		Pass		Resolve backend issues
Validate Navigation	All navigation links redirect to correct	Links work as expected without 404 errors.		Pass		Update broken links.

### Regression Test Suite for <https://gov.kz/>

Test Case	User Requirements Addressed	Expected Result	Actual	Pass/Fail	Status	Corrective Action if
Validate Service	Users can submit requests for services.	Form submissions are successful, with		Fail		Fix submission or
Test User Profile	Users can update their profile information.	Updates are saved, and changes are reflected		Fail		Debug database write
Test Mobile Responsiveness	The website renders properly on mobile	All elements are accessible and usable on		Pass		Adjust CSS for mobile
Verify Performance	The website handles 1000 concurrent users	No crashes, lags, or errors under the specified load.		Fail		Optimize server
Check Multi-Language	Users can switch between Kazakh,	Content updates to match the selected language.		Pass		Fix language toggling
Validate Secure	Payments and sensitive data are encrypted.	Data encryption works; no unauthorized access is		Fail		Address vulnerabilities