UNIVERSITY OF ECONOMICS - VARNA

CATHEDRA „INFORMATIX“

ИКОНОМИЧЕСКИ УНИВЕРСИТЕТ - ВАРНА

С АКАДЕМИЧНИ ТРАДИЦИИ В БЪДЕЩЕТО

Основан 1920

COURSE WORK

ON AUTOMATED SOFTWARE TESTING

THEME: **The Internet Archive (eBooks and Texts section)**

Developed By:

127771 Nikita Romanovich Oechkin, 34th gr.

QA:..

/………………… /

Varna

2025

Table of Contents

I. Introduction

The Internet Archive is an American public, non-profit digital library founded in 1996 with the goal of preserving digital content and providing longterm public access to knowledge. Before becoming widely known as a digital archive, the organization originated from a web-crawling and indexing initiative created by Brewster Kahle. This early project focused on systematically collecting and storing snapshots of web pages, which later evolved into what is now known as the Wayback Machine. Over time, the scope of the Internet Archive expanded beyond web archiving to include books, academic texts, audio recordings, video materials, and software.

One of the main sections of the platform is the "Texts" collection, which as of 13.12.2025 contains 48,048,250 of digitized books, research publications, historical documents, and user-contributed written works. The Texts section allows users to browse available materials, perform keyword searches, apply filters etc. Each item page typically presents structured metadata such as title, author, publication year, subject classification, and a short description, as well as options for online reading and file downloads in various formats (TORRENT, PDF, EPUB, TXT, etc.). The core functionality of the section is publicly accessible and does not require user authentication.

The scope of this course work focuses on testing the public, read-only functionality of the Internet Archive's Texts section as an independent system under test. The testing covers: navigation to the Texts landing page, browsing and pagination of results, search behavior, opening individual text items, visibility and consistency of metadata, availability of download options, and basic error handling for invalid or empty search results.

Excluded from the scope is functionality related to: user accounts, content uploading, backend processing, and validation of file contents. The objective of the testing is to evaluate the reliability and usability of the core user-facing features of the Texts section from a quality assurance perspective.

II. Analysis & Test Planning

This section defines what is tested, what is explicitly excluded, and how testing is performed. It establishes clear boundaries for the coursework and avoids overlap with the detailed execution artifacts presented later in the Test Documentation section.

1. Scope Definition

1.1. In Scope

The following functionality of the Internet Archive Texts section is included in testing. All items listed below refer exclusively to public, read-only user interactions available without authentication.

- Loading and accessibility of the Texts landing page
- Browsing of text collections and result lists
- Keyword-based search functionality within the Texts section
- Application of available filters and collections
- Pagination of browsing and search results
- Opening and loading of individual text item pages
- Visibility and consistency of metadata elements, including: title, author, publication year, description, subject information (where available)
- Presence and accessibility of download options (PDF, EPUB, plain text, etc.), without validating file contents
- Navigation behavior, including internal links and browser back navigation
- Error handling: searches returning no results
- Error handling: non-existent or invalid item URLs

The scope focuses on validating user-facing behavior and interface stability, rather than internal system correctness or archival content accuracy.

1.2. Out of Scope

The following functionality is explicitly excluded from testing, either due to access restrictions, ethical considerations, or irrelevance to the coursework objectives:

- User authentication and account-related features (login, profiles, favorites, borrowing)

- Uploading, editing, or managing text items
- Validation of file contents, OCR quality, or document correctness
- Performance, load, stress, or security testing
- Mobile applications or native device behavior
- Administrative or moderation functionality
- Any functionality not explicitly listed in the In Scope section

This exclusion ensures that testing remains focused, non-invasive, and appropriate for a public production system.

1.3. Business-Critical Functions

The following functions are considered business-critical within the Texts section:

- Reliable search functionality
- Successful loading of individual text item pages
- Correct and consistent display of metadata
- Stable browsing and pagination of large result sets
- Clear visibility of available download formats

Failure in these areas significantly impacts usability and access to archived materials.

2. Test Strategy (Manual & Automation)

Testing is conducted using a black-box approach, where the system under test is evaluated solely through its externally observable behavior. No assumptions are made about backend services, databases, or internal implementation details. The Internet Archive Texts section is treated as an independent, public-facing web application. Manual and automated testing are used in a complementary manner:

- Manual testing is applied to exploratory testing, validation of UI behavior, and identification of edge cases that require human observation.

- Automated testing is implemented to verify repeatable and deterministic behavior such as page loading, navigation flows, search execution, pagination, and item page rendering.

Automation is implemented exclusively using Playwright.

2.1. Test Types Included

The following test types are applied within the defined scope:

- Functional testing
- Positive and negative scenario testing
- Navigation testing
- UI consistency validation
- Basic stability checks

Non-functional testing (performance, load, stress, and security testing) is intentionally excluded, as defined in the scope.

2.2. Test Environment

- Operating System: Linux (Arch btw)
- Browser: Firefox (executed via Playwright)
- Application Type: Public web application

2.3. Tools

- Playwright (used for browser automation and test execution)

No additional external tools, repositories, or CI systems are used. All testing artifacts are maintained locally and documented within the coursework.

III. Test Documentation

This section presents the concrete testing artifacts derived from the analysis and planning described in Section II. It documents how the defined scope was translated into executable test cases and how observed defects were recorded during test execution.

3.1 Test Plan

Test Objectives

The objective of this test plan is to validate the stability and correctness of the publicly accessible functionality of the Internet Archive Texts section. The testing aims to confirm that end users can browse, search, and access textual items reliably, and that the user interface behaves consistently under normal usage conditions.

The test plan focuses exclusively on observable, user-facing behavior and does not assess internal system logic, backend services, or the accuracy of archived content.

Test Coverage Overview

Based on the defined scope, testing covers the following functional areas:

- Texts landing page access and loading
- Browsing and pagination of text collections
- Keyword-based search behavior
- Filter and collection application
- Navigation between result lists and item pages
- Display of item metadata
- Visibility of available download options
- Handling of empty search results
- Handling of invalid or non-existent item URLs

Assumptions and Constraints

- The Internet Archive is a live production system; content and layout may change during testing
- Text items may differ in metadata completeness and available formats
- Tests are executed without authentication

- No destructive or intrusive actions are performed

3.2 Test Cases

The following test cases were designed to validate the functionality defined in the test coverage. All test cases are executed against the public Texts section and do not require user authentication.

TC-01 - Load Texts Landing Page

Preconditions: Internet connection available

Steps:

1. Navigate to the Internet Archive Texts page

Expected Result:

The Texts landing page loads successfully and displays a list of text items or collections.

TC-02 - Browse Text Collections

Preconditions: Texts landing page is loaded

Steps:

1. Scroll through the available list of text items

Expected Result:

Multiple text items are displayed, each represented by a title and visual identifier.

TC-03 - Search with Valid Keyword

Preconditions: User is on the Texts page

Steps:

1. Enter a common keyword in the search field
2. Submit the search

Expected Result:

A results page is displayed containing items relevant to the search keyword.

TC-04 - Search With No Results

Preconditions: User is on the Texts page

Steps:

1. Enter a random, non-existent string into the search field
2. Submit the search

Expected Result:

The system displays a message or empty state indicating that no results were found.

TC-05 - Pagination of Results

Preconditions: Search or browsing results span multiple pages

Steps:

1. Navigate to the next page using pagination controls

Expected Result: The next page of results loads successfully.

TC-06 - Apply Filter or Collection

Preconditions: Results page is visible

Steps:

1. Select an available filter or collection option

Expected Result:

The displayed results update according to the selected filter or collection.

TC-07 - Open Text Item Page

Preconditions: A list of text items is visible

Steps:

1. Select a text item from the results

Expected Result:

The corresponding item page loads successfully.

TC-08 - Verify Metadata Display

Preconditions: Text item page is open

Steps:

1. Locate the metadata section
2. Verify the presence of title and at least one additional metadata field

Expected Result:

Metadata elements are visible and readable.

TC-09 - Verify Download Options Visibility

Preconditions: Text item page is open

Steps:

1. Scroll to the download options section

Expected Result:

Available download formats are displayed as accessible links, where provided.

TC-10 - Browser Back Navigation

Preconditions: Text item page is open

Steps:

1. Use the browser’s Back button

Expected Result:

The user is returned to the previous results page.

TC-11 - Invalid Item URL Handling

Preconditions: None

Steps:

1. Navigate to a non-existent item URL

Expected Result:

An “No full-text search results were found for this query.” message is displayed.

3.3 Bug Reports

The following defect reports document issues observed during test execution. Defects are based solely on visible UI behavior and do not assume internal system failures.

IV. Automation Part

1. Setup & Environment

Automated Instalation of (Node.js, Playwright);

environment: Arch/Debian based linux distributions

```bash
#!/bin/bash

echo "Spying on your package manager..."

if command -v apt &> /dev/null; then

    echo "I spy with, my little eye APT!!! . Installing Node.js and npm..."

    sudo apt update

    sudo apt install -y nodejs npm

elif command -v pacman &> /dev/null; then

    echo "Pacman detected, I sure hope you're not running Manjaro. Installing Node.js and npm..."

    sudo pacman -Syu --noconfirm nodejs npm

else

    echo "Bruv, i dont even know what package manage r u runnin', just google how to install Node.js and npm"

    exit 1

fi

echo "Installing Playwright..."

npm install -D @playwright/test

echo "Installing browsers for Playwright..."

npx playwright install

echo "Done and done!!! You can now run tests with 'npx playwright test'."
```

2. Automated Tests

loadTextsPage.spec.ts

```
import { test, expect } from '@playwright/test';

test('TC-01 - Load Texts Landing Page', async ({ page }) => {

/*act

MOVE ME TO THE INTERNET ARCHIVE TEXTS LLLLLANDING PAGE*/

await page.goto('/details/texts', { waitUntil:'domcontentloaded' });

/*assert //dominance

check if evelysin is oke*/

await expect(page).toHaveURL(/texts/);

});
```

searchValid.spec.ts

```
import { test, expect } from '@playwright/test';

test('TC-03 - Search with Valid Keyword', async ({ page }) => {

/*act

to search results page for an nice book*/

await page.goto('/searchquery=Do+Androids+Dream+of+Electric+Sheep&tab=collection',{
waitUntil: 'domcontentloaded' });

/*assert

see if has the book*/

const results = page.locator('article.cell-container');

await expect(results.first()).toBeVisible({ timeout: 10000 });

const count = await results.count();

expect(count).toBeGreaterThan(0);

});
```

searchNoResults.spec.ts

```
import { test, expect } from '@playwright/test';

test('TC-04 - Search With No Results', async ({ page }) => {

/*act

goes to page vith an invalid key word search*/

await page.goto('/search?query=nonexistentrandomxyz123&and[]=mediatype%3A%22texts%22',{ waitUntil: 'domcontentloaded' });

/*assert

checks ifpage haz ZEROU items displayed*/

await expect(page.locator('.item-ia')).toHaveCount(0);

});
```

V. Test Summary Report

- Total Test Cases Executed: 11
    - Automated UI Tests (3):
        - TC-01 - Load Texts Landing Page
        - TC-03 - Search With Valid Keyword
        - TC-04 - Search With No Results
    - Manual Tests (8):
        - TC-02 - Browse Text Collections
        - TC-05 - Pagination of Results
        - TC-06 - Apply Filter or Collection
        - TC-07 - Open Text Item Page
        - TC-08 - Verify Metadata Display
        - TC-09 - Verify Download Options Visibility
        - TC-10 - Browser Back Navigation
        - TC-11 - Invalid Item URL Handling

- Successful Test Cases: 11
- Failed Test Cases: 0
- Defects Found: None
- Interpretation of Results:
    - All automated and manual tests passed.
    - Texts landing page, search, browsing, item pages, metadata, filters, pagination, download options, and error handling work as expected.
    - No inconsistencies or unexpected behavior were observed.
- Final Recommendations:
    - Maintain automated tests for critical paths; continue manual testing for new features.
    - Monitor UI and content updates that may require test updates.
    - Consider automating additional manual tests in the future for more coverage.