

Configuration Manual

MSc Research Project
Programme Name

Suraj Rakesh Gupta
Student ID: x23292717

School of Computing
National College of Ireland

Supervisor: Eugene Mclaughlin

National College of Ireland
MSc Project Submission Sheet
School of Computing



Student Name: Suraj Rakesh Gupta

Student ID: x23292717

Programme: MSCCYB1_A

Year: 2024-2025

Module: MSc Research Project

Lecturer: Eugene McLaughlin

Submission

Due Date: 30/05/2025

Project Title: Implementation of Cyrillic Script Checker to Detect Phishing Attacks
URLs/links
.....

Word Count: 758

Page Count: 10

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Suraj Rakesh Gupta

Date: 30/05/2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Suraj Rakesh Gupta

Student ID: x23292717

<https://cyrillic-guard.vercel.app/>

1 Introduction

This manual will provide an extensive overview for setting up and launching the "Cyrillic Script Checker," a tool aimed at identifying phishing URLs that utilize Cyrillic homoglyph characters. The system detects visually alike Cyrillic-Latin character replacements (e.g., facebook.com vs. facebook.com) and marks possible phishing attempts. This document includes dependencies of tools, laboratory configuration, and steps for implementation

2 Tools /Languages

Software Stack:

- **Frontend:**
 - HTML/CSS/JavaScript (for user interface and client-side logic)
 - Unicode normalization libraries (e.g., punycode.js for URL parsing)
- **Backend:** None (client-side only)
- **Testing:**
 - Browser Developer Tools (Chrome/Firefox)
 - VSCode live server
 - Burp Suite Community Edition (optional for HTTP request analysis)

Key Libraries:

- **Homoglyph Detection:** Custom JavaScript functions for Cyrillic-Latin character mapping.
- **URL Parsing:** URL API for domain extraction.

3 Hardware Configuration

Component	Specification
Operating System	Windows 11 / macOS Ventura / Linux Ubuntu
Processor	Intel i5 or equivalent (64-bit)
RAM	8 GB DDR4 or higher
Storage	256 GB SSD (minimum)

Component	Specification
Browser	Chrome v116+ or Firefox v115+

4 Lab Setup

Local Server Configuration (VSCode):

1. Install VSCode and Live server extension:
 - VSCode live server
2. Deploy Project:
 - Open index.html.
3. Start Servers:
 - Launch live server “Go Live”.
4. Access Application:
 - Automatically open in browser

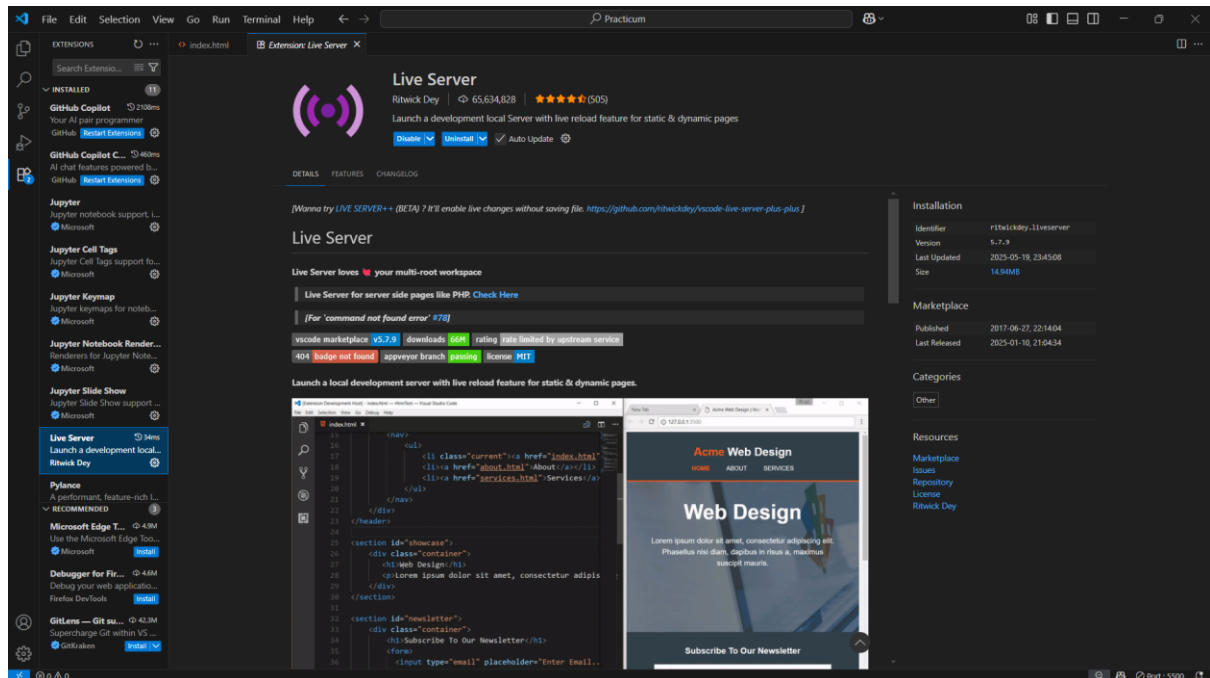


Figure 1: Live server to see the project live

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1.0">
6 <title>Cyrillic Phishing Detection Framework</title>
7 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">
8 <style>
9 :root {
10   --gradient-start: #2c3e50;
11   --gradient-end: #3498db;
12 }
13
14 body {
15   font-family: 'Segoe UI', system-ui, -apple-system;
16   background: linear-gradient(135deg, var(--gradient-start), var(--gradient-end));
17   color: #ffff;
18 }
19
20 .detection-box {
21   background: rgba(255, 255, 255, 0.1);
22   border-radius: 15px;
23   backdrop-filter: blur(10px);
24   border: 1px solid rgba(255, 255, 255, 0.2);
25 }
26
27 .cyrillic-char {
28   color: #ff7f50;
29   font-weight: bold;
30 }
31 </style>
32 </head>
33 <body>
34 <!-- Navigation -->
35 <nav class="navbar navbar-expand-lg navbar-dark bg-dark fixed-top">
36   <div class="container">
37     <a class="navbar-brand" href="#>Cyrillic Guard</a>
38     <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav">
39       <span class="navbar-toggler-icon"></span>
40     </button>
41   </div>
42 </nav>
43
44 <!-- Hero Section -->
45 <header class="py-5 mt-5">
46   <div class="container text-center">
47     <h1 class="display-4 mb-4">Detecting Cross-Cultural Phishing Attacks</h1>
48     <p class="lead">Advanced detection system for Cyrillic script-based homograph attacks</p>

```

Figure 2: Final Code view index.html

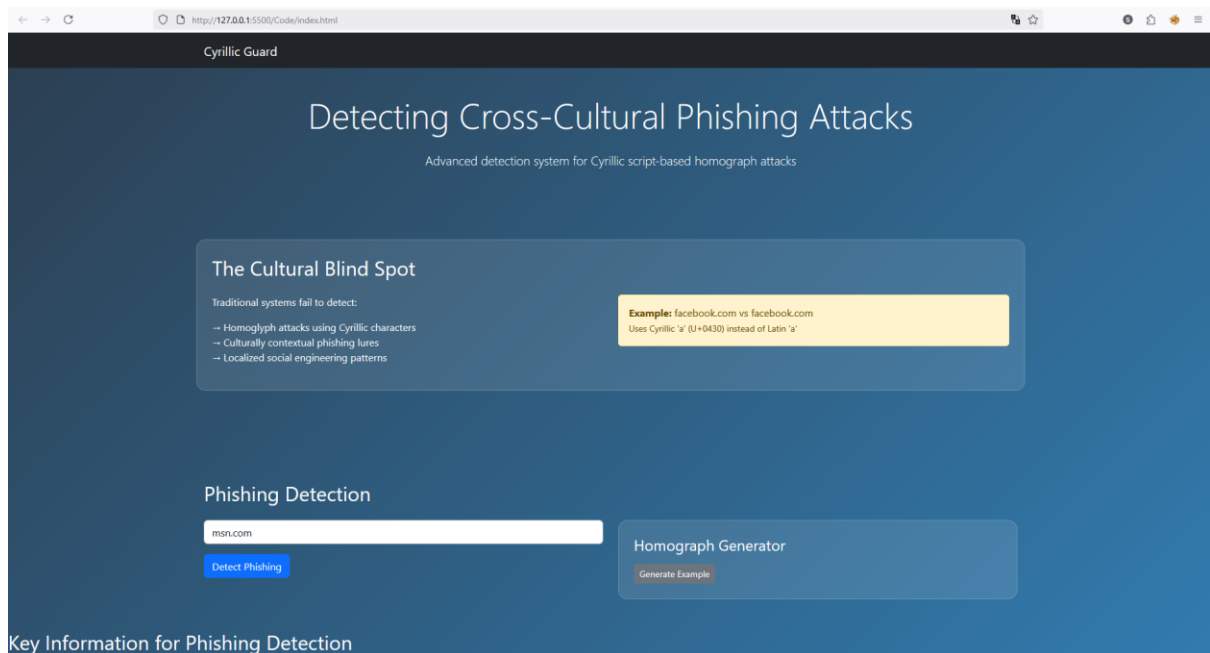


Figure 3: Final Website View

Key Information for Phishing Detection

Common Homograph Substitutions:

- a → ɑ Latin 'a' (U+0061) vs Cyrillic 'а' (U+0430)
- e → ɛ Latin 'e' (U+0065) vs Cyrillic 'е' (U+0435)
- o → ɑ Latin 'o' (U+006F) vs Cyrillic 'о' (U+043E)
- p → Ɔ Latin 'p' (U+0070) vs Cyrillic 'р' (U+0440)
- x → ɣ Latin 'x' (U+0078) vs Cyrillic 'х' (U+0441)
- ɣ → ɣ Latin 'y' (U+007B) vs Cyrillic 'у' (U+044C)

Detection Range: Cyrillic characters fall within Unicode range U+0400 to U+04FF (1024-1279 decimal)

English Alphabet	Unicode	Russian Alphabet	Unicode
A	U+0041	А	U+0410
B	U+0042	Б	U+0411
C	U+0043	В	U+0412
D	U+0044	Г	U+0413
E	U+0045	Д	U+0414
F	U+0046	Е	U+0415
G	U+0047	Ж	U+0416
H	U+0048	З	U+0417
I	U+0049	И	U+0418
J	U+004A	Й	U+0419
K	U+004B	К	U+041A
L	U+004C	Л	U+041B
M	U+004D	М	U+041C
N	U+004E	Н	U+041D
O	U+004F	О	U+041E
P	U+0050	П	U+041F
Q	U+0051	Р	U+0420
R	U+0052	С	U+0421
S	U+0053	Т	U+0422
T	U+0054	У	U+0423
U	U+0055	Ф	U+0424
V	U+0056	Х	U+0425
W	U+0057	Ц	U+0426
X	U+0058	Ч	U+0427
Y	U+0059	Ш	U+0428
Z	U+005A	Щ	U+0429

Figure 4: Comparison on Website for clear understanding

5 Implementation

5.1 Core Workflow:

1. **URL Input:** Users enter a URL into the web interface.
2. **Homoglyph Detection:** The script converts the URL to Unicode, checking for Cyrillic characters in the domain.
3. **Example code snippet:**

```
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>
<script>
  // Cyrillic Detection Logic
  function detectCyrillic(text) {
    return /\u0400-\u04FF/i.test(text);
  }

  // Homograph Generator
  const homographMap = {
    'a': 'ɑ', 'c': 'Ɔ', 'e': 'ɛ',
    'o': 'ɑ', 'p': 'Ɔ', 'x': 'ɣ'
  };

  function generateHomograph() {
    const example = 'facebook';
    const homograph = example.split('').map(c =>
      homographMap[c.toLowerCase()] || c
    ).join('');
    document.getElementById('homographOutput').innerHTML = `
      <span class="text-danger">${homograph}</span>.com
    `;
  }

  // URL Analysis
  function analyzeURL() {
    const input = document.getElementById('urlInput').value;
    const hasCyrillic = detectCyrillic(input);

    const resultDiv = document.getElementById('result');
    resultDiv.innerHTML = hasCyrillic ?
      `<div class="alert alert-danger">Cyrillic characters detected!</div>` :
      `<div class="alert alert-success">No Cyrillic detected</div>`;
  }
</script>
```

6 Testing & Results

In this section we will run some live test using the own created datasets you can find in here : <https://github.com/x23292717/Cyrillic-Guard/blob/main/URLs.csv>. In here we will compare original and fake Cyrillic URL and see if the created logic can distinguish between fake and real and detect Cyrillic script present and URL that we have input. Below is the Unicode range shown for example:

Unicode Ranges

English Alphabet: U+0041-U+005A (uppercase), U+0061-U+007A (lowercase)

Russian Alphabet: U+0410-U+044F (basic Cyrillic range)

Test case 1: Testing on google real website and a fake website

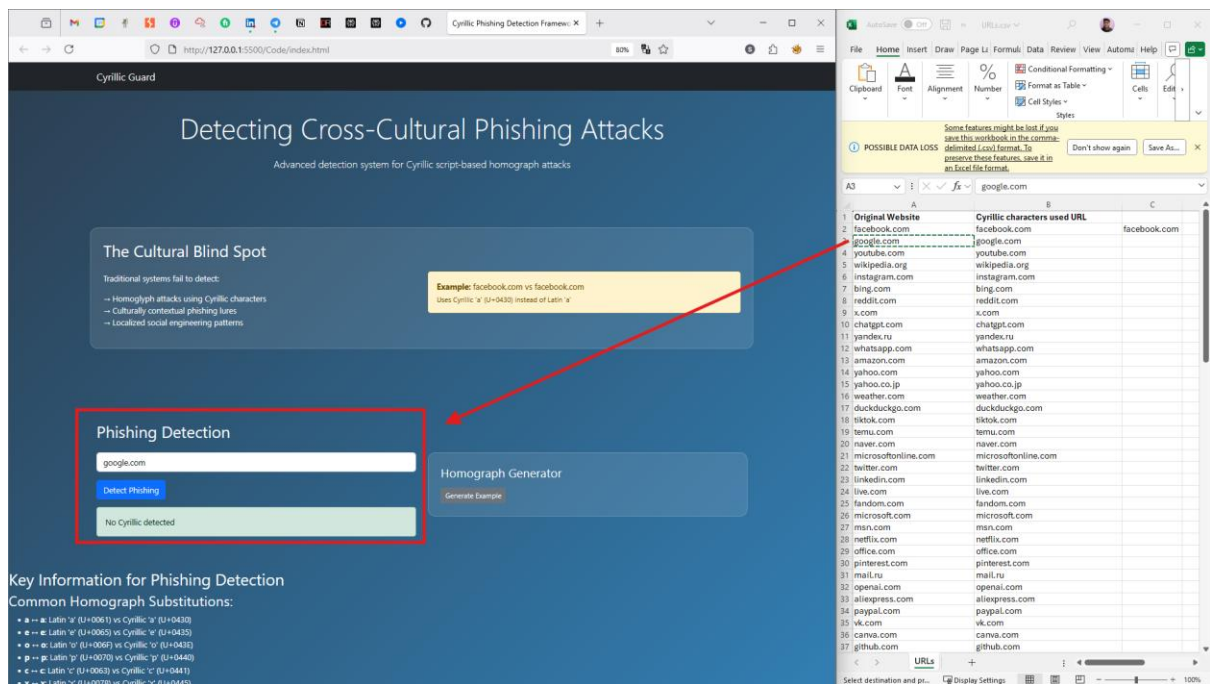


Figure 5: Real google.com URL show "No Cyrillic detected"

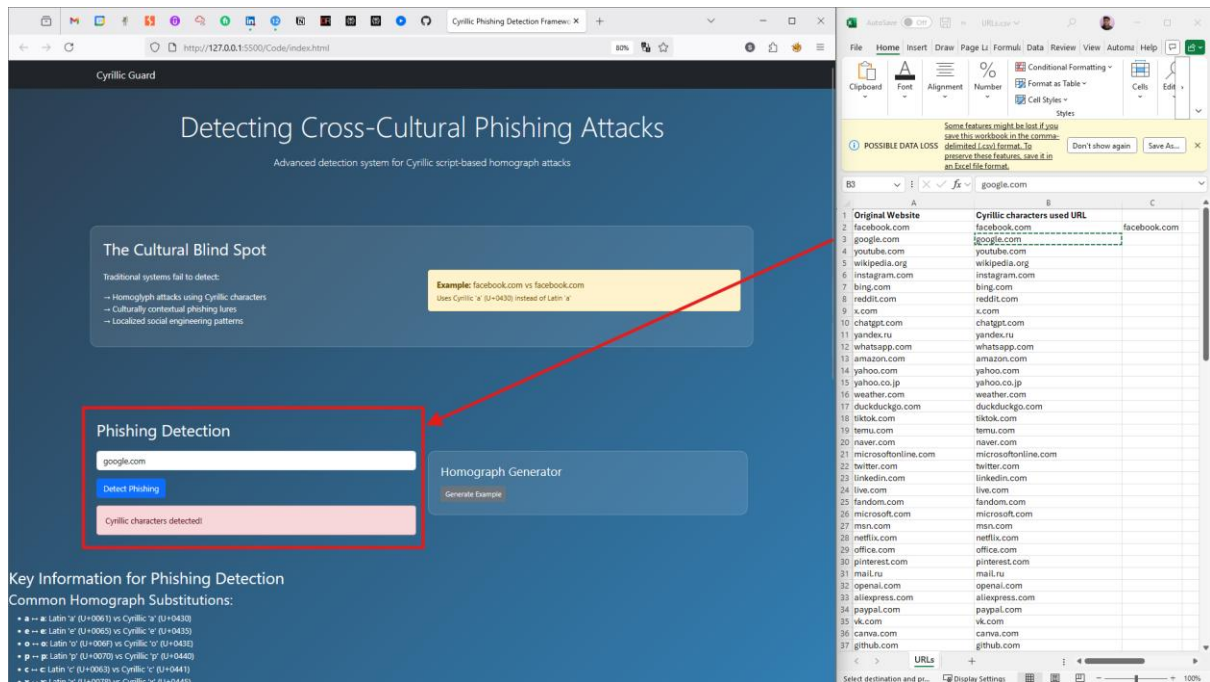


Figure 6: Fake Cyrillic URL of google.com show "Cyrillic characters detected!"

Test case 2: Testing on x.com real website and a fake website

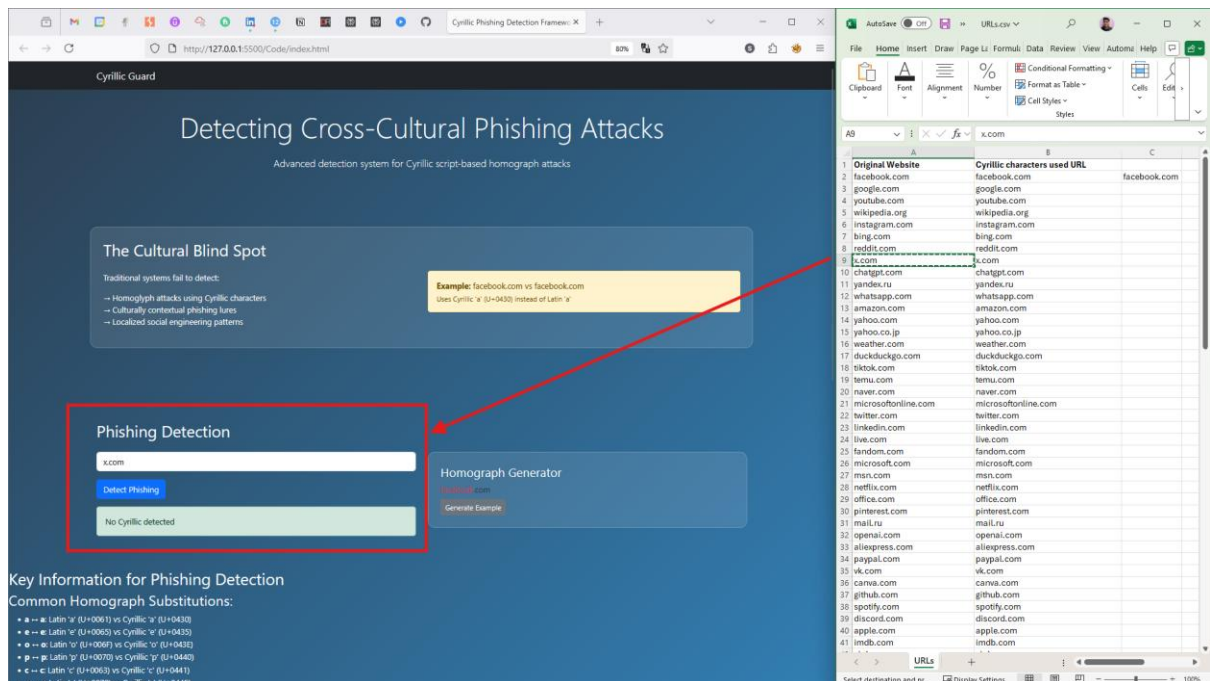


Figure 7: Real x.com URL show "No Cyrillic detected"

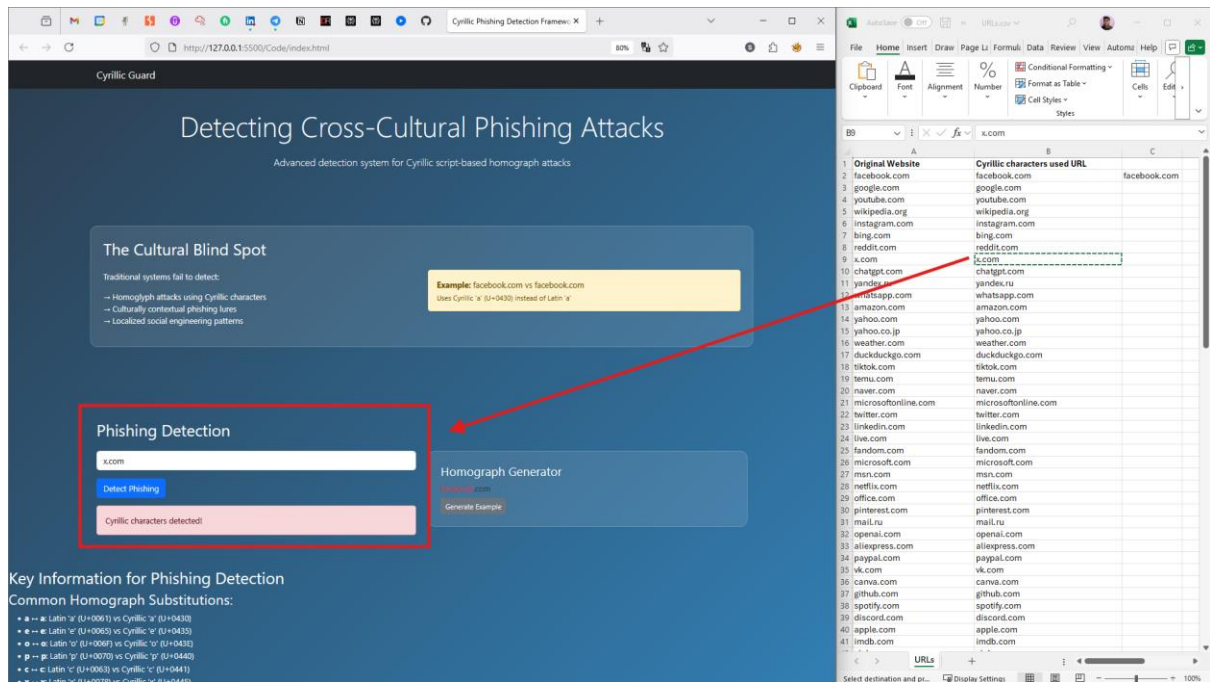


Figure 8: Fake Cyrillic URL of x.com shows "Cyrillic characters detected!"

Test case 3: Testing on instagram.com real website and a fake website

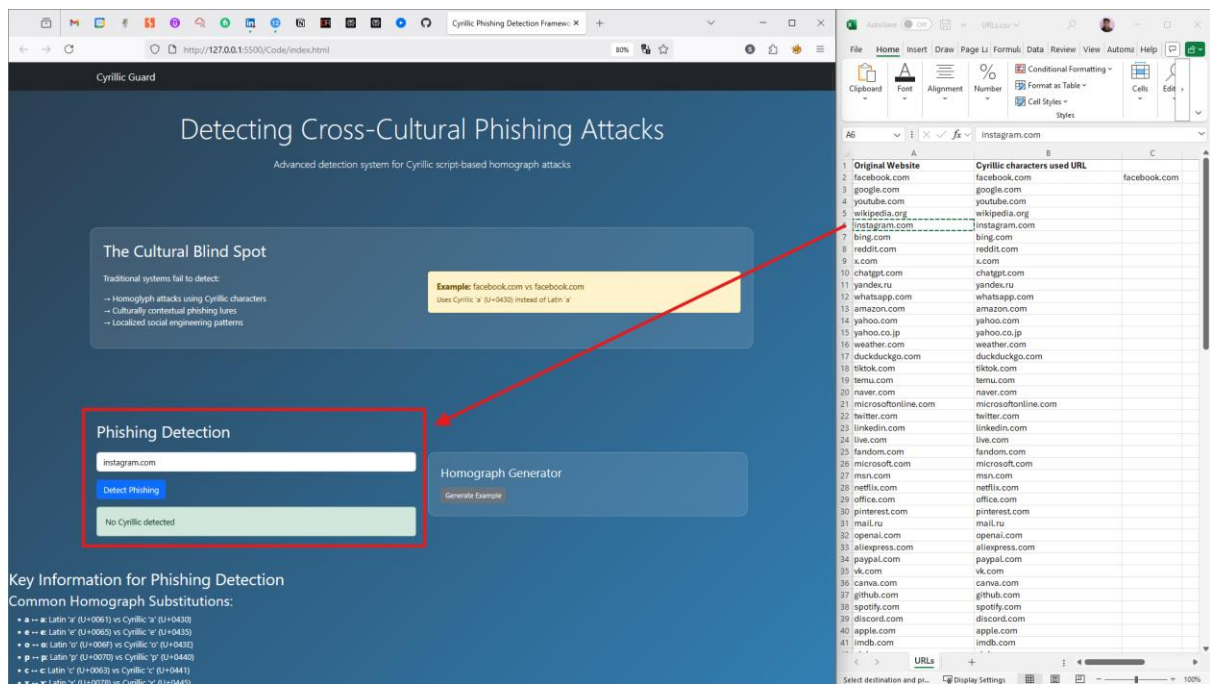


Figure 9: Real instagram.com URL shows "No Cyrillic detected"

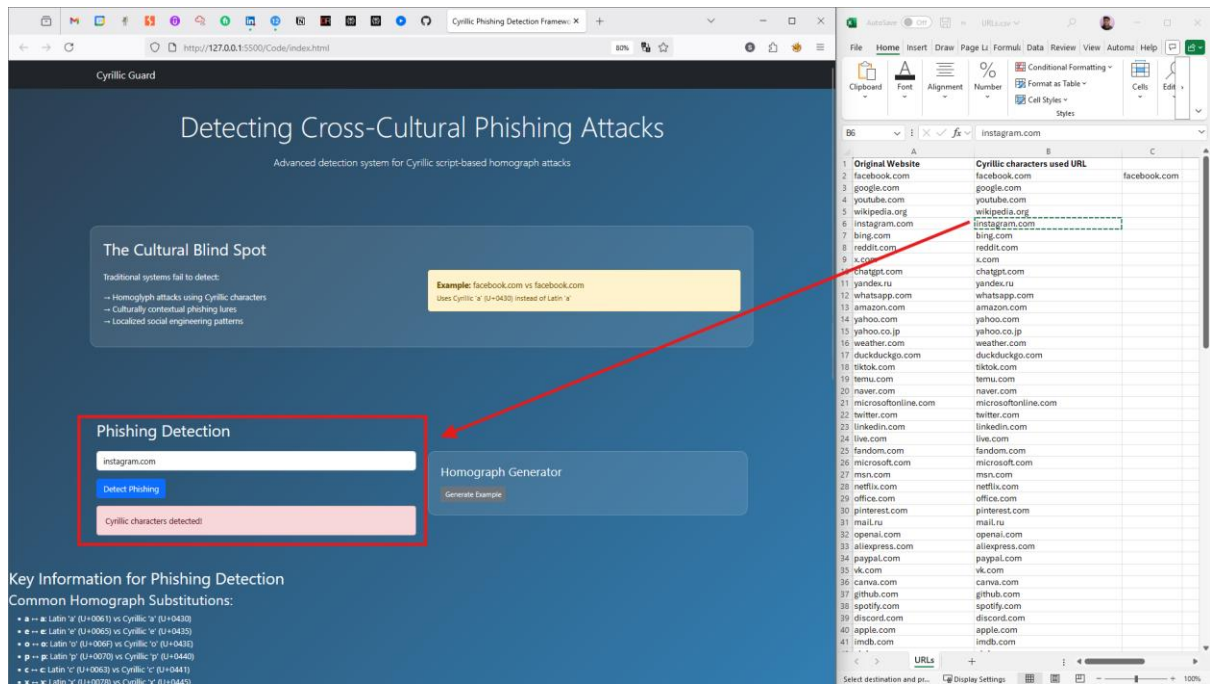


Figure 10: Fake Cyrillic URL of instagram.com shows "Cyrillic characters detected!"

References

- [1] Microsoft, "Visual Studio Code," Visualstudio.com, 2024. <https://code.visualstudio.com/> (accessed Jun. 17, 2025).
- [2] timofurrer, "idn-homograph-attack/data at master · timofurrer/idn-homograph-attack," GitHub, 2017. <https://github.com/timofurrer/idn-homograph-attack/tree/master/data> (accessed Jun. 17, 2025).
- [3] x23292717 - Suraj Rakesh Gupta, "Cyrillic-Guard/index.html at main · x23292717/Cyrillic-Guard," GitHub, 2025. <https://github.com/x23292717/Cyrillic-Guard/blob/main/index.html> (accessed Jul. 08, 2025).