Objective:

To demonstrate a basic detection mechanism that flags potentially **malicious or deceptive domain names** using **homoglyph characters**, **punycode encoding**, or **mixed Unicode scripts** that imitate legitimate domains (e.g., "gοοgle.com" instead of "google.com").

Tools & Technologies Used:

Component	Purpose
Python 3.x	Core scripting language
`homoglyphs`	Detect visually similar Unicode chars
`unicodedata`	Normalize and analyze Unicode chars
`re` (regex)	Parsing and cleaning domain inputs
`idna` (built-in via encode/decode)	Decode punycode/IDN domains

Detection Logic Overview:

Check	Description
Unicode Normalization	Uses NFKC form to clean up Unicode strings
Punycode Detection	Flags domains like `xn--pple-43d.com`
Homoglyph Detection	Converts Unicode variants to ASCII & compares
Mixed Script Analysis	Flags domains mixing Latin + Cyrillic/Greek

How the Script Works:

1. **User inputs** one or more URLs/domains via console.
2. Script **extracts the base domain**, stripping `http(s)` and paths.
3. It performs:
    - Unicode normalization
    - Homoglyph ASCII replacement
    - Script analysis (Latin, Cyrillic, etc.)
4. Flags the domain if:
    - It's punycode (`xn--`)
    - It uses homoglyphs
    - It mixes multiple Unicode writing systems
5. Displays flagged domains and reasons.

Test Cases:

Input	Flagged?	Reason(s)
https://google.com	Yes	Contains homoglyphs
https://xn--pple-43d.com	Yes	Punycode + homoglyphs
https://apple.com	Yes	Cyrillic ‘a’ + Mixed scripts
https://g00gle.com	No	No Unicode homoglyphs (numeric spoof, not handled)
https://microsoft.com	No	Clean domain

Demo Output Sample:

```
Homoglyph Script Domain Scanner
Type/paste domain or full URL(s) to scan. Use comma or newline to separate
multiple entries.
Type 'exit' or 'q' to quit.

Enter domain(s) or URL(s): https://google.com
 Suspicious domain detected: google.com
↪ Decoded / Normalized: google.com
🔎 Reason: Contains homoglyphs

Enter domain(s) or URL(s): https://xn--pple-43d.com
 Suspicious domain detected: xn--pple-43d.com
↪ Decoded / Normalized: apple.com
🔎 Reason: Punycode-encoded (IDN domain)
🔎 Reason: Contains homoglyphs

Enter domain(s) or URL(s): https://microsoft.com
(no alerts)
```

Limitations:

- Doesn't detect numeric lookalikes (e.g., `g00gle.com`)
- False positives possible for legitimate internationalized domains (IDNs)
- No integration with external threat feeds or domain age info

Next Steps for Enhancement:

- Add support for **Levenshtein similarity**
- Include **IDN allowlist exceptions**
- Batch scan from file or clipboard
- Export suspicious results to a CSV/JSON file

Conclusion:

This PoC successfully demonstrates that homoglyph, punycode, and mixed-script-based spoofing attacks can be detected using Unicode analysis — **without the need for a hardcoded whitelist**.

It’s a **powerful foundation** for building anti-phishing tooling or browser-level detection modules.