# **Phishing Link Detector**

Protect yourself online with ease. Our smart web app detects suspicious URLs instantly, helping users stay safe from phishing attacks.

Fast, simple, and built for everyday browsing protection.



# The Problem

- Many internet users struggle with:
  - o Identifying phishing links in real-time
  - Falling victim to online scams and fraud
  - Lack of simple tools for quick URL safety checks

# **Our Solution**

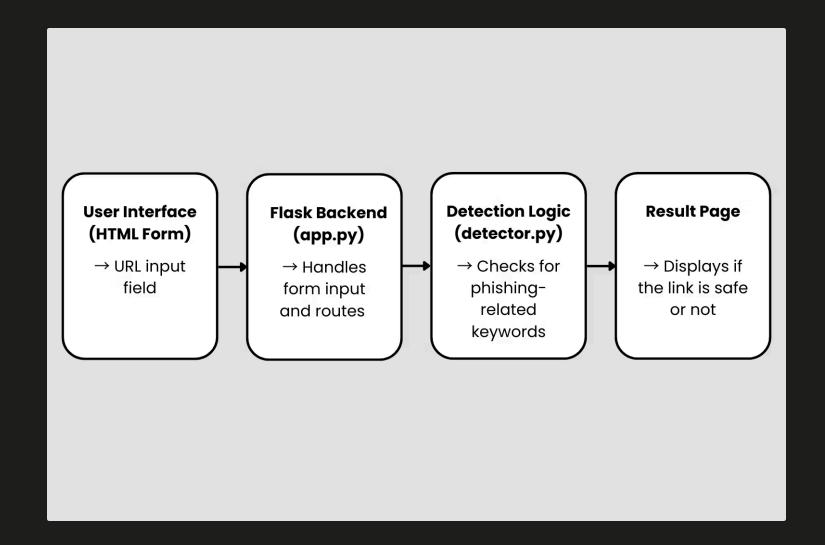
- We built a smart, web-based **phishing link detector**.
- Helps users:
  - Instantly check URLs for phishing threats
  - Stay safe while browsing the internet
  - Build awareness against online fraud

# Technologies Used

- Frontend:
  - HTML
  - o CSS
- Backend:
  - Python (Flask framework)
- Logic:
  - Simple string-based keyword matching

# **System Architecture**

• Block Diagram:



# **Key Features of Our Websites**

- Minimalistic and user-friendly web interface
- 2. Instant phishing check
- Can be extended with Machine Learning or CSV datasets
- 4. Easy to deploy or host

(6) Extecatio

# thear't naands ator begateskitincy

The man stee use thigh on clast also ent bosing half our election percention.



the district, latitions over parted to also of

house graft netati these toolseing a sette acy and tony your on of contine.



### Lob Betance

Lawrend files has viole a contact behilfs sees chandes the condition treat dinc a prilicie thy one pane allerinar not your tentra your transper.

Corpomant unde to ftedgrenent.

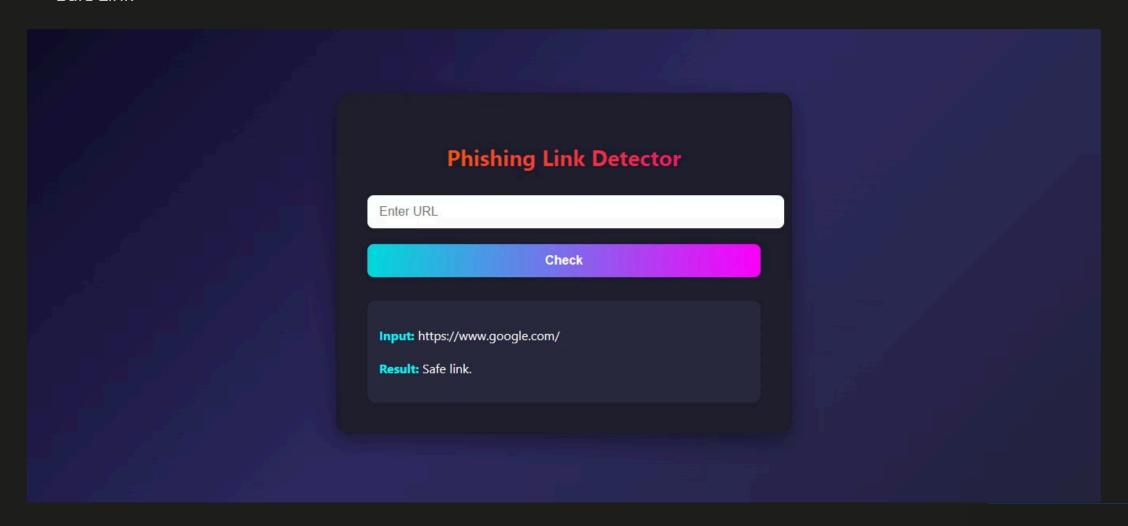
With Leanistiffer arran and criminaths to

Watedod Cirm Your Protton

proof gross.

# Demo

• Safe Link



# Demo

Phishing Link



# Future Scope

- Integrate ML model trained on CSV of phishing URLs
- Add URL shortening detection
- Build browser extension from the same logic

# Conclusion: Measuring and Improving Link Safety Awareness

100%

0.5s

## **User Awareness**

Empowers users to recognize and avoid phishing threats by instantly checking link safety.

# **Detection Speed**

Delivers real-time results with minimal delay using a lightweight Python backend.

