PHISHING DETECTION AND PREVENTION SYSTEM

A MINI-PROJECT REPORT

Submitted by

MOHANA M **231901031**

MONIKA E **231901032**

in partial fulfillment of the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING (CYBER SECURITY)

RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI

An Autonomous Institute

CHENNAI

MAY 2025

BONAFIDE CERTIFICATE

Certified that this project **"PHISHING DETECTION AND PREVENTION SYSTEM"** is the bonafide work of **"MOHANA M , MONIKA E"** who carried out the project work under my supervision.

SIGNATURE

Mrs. JANANEE V

ASSISTANT PROFESSOR

Dept. of Computer Science and Engg,

Rajalakshmi Engineering College

Chennai

This mini project report is submitted for the viva voce examination to be held on

---

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ABSTRACT

In the current digital era, phishing has emerged as one of the most prevalent and deceptive forms of cyberattack, aimed at stealing sensitive information such as login credentials, banking details, and personal data. To address this challenge, we present a **Phishing Detection and Prevention System** that offers a real-time, interactive, and system-integrated solution.

This project enables users to check the safety of URLs through a user-friendly web interface. Upon entering a URL, the system analyzes its legitimacy by comparing it against known phishing data sources. If the URL is identified as phishing, the system dynamically responds with a red background, warning sound, and screen shake animation to grab user attention. For safe URLs, the interface turns green, providing immediate and intuitive feedback.

Additionally, the system features graphical analytics using **Chart.js** to visualize the efficiency of operating system modules such as firewalls, sandboxing, AI-based detection, and real-time alerts. A logging module records each scanning event with timestamps, contributing to traceability and incident review.

This solution not only prevents potential phishing attempts but also educates users by providing visual insights into how system-level components contribute to cybersecurity. The integration of OS concepts makes this project both technically rich and practically valuable in the field of digital forensics and cybersecurity.

ACKNOWLEDGEMENT

We express our sincere thanks to our beloved and honorable chairman **MR. S. MEGANATHAN** and the chairperson **DR. M.THANGAM MEGANATHAN** for their timely support and encouragement.

We are greatly indebted to our respected and honorable principal **Dr. S.N. MURUGESAN** for his able support and guidance.

No words of gratitude will suffice for the unquestioning support extended to us by our Head Of The Department **Mr. BENEDICT JAYAPRAKASH NICHOLAS M.E Ph.D.,** for being ever supporting force during our project work.

We also extend our sincere and hearty thanks to our internal guide **Mrs**.V.**JANANEE,** for her valuable guidance and motivation during the completion of this project.

Our sincere thanks to our family members, friends and other staff members of computer science engineering.

1. **MOHANA M(231901031)**
2. **MONIKA E (231901032)**

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

Phishing Detection and Prevention System is designed to identify malicious URLs in real-time and provide immediate feedback to users. This project leverages modern web technologies and operating system-level concepts to alert users about threats through visual and auditory signals.

1.2 SCOPE OF THE WORK

The system is intended to help users identify phishing links and avoid falling prey to fraudulent websites. It ensures system-level awareness through graphs and logs and supports basic URL threat detection through keyword and list-based matching.

1.3 PROBLEM STATEMENT

Phishing is a common cyber threat that tricks users into visiting fake websites to steal sensitive information. Most existing tools rely on outdated databases or offer minimal interaction, making them less effective. Our project addresses this gap by providing a real-time phishing detection system with engaging feedback and system-integrated features to improve awareness and prevent data breaches.

1.4 AIM AND OBJECTIVES OF THE PROJECT

- To create a user-friendly web-based platform to detect phishing URLs.
- To incorporate visual feedback for immediate response.
- To graphically display how OS modules assist in phishing prevention.
- To log detection activities for review and learning.

CHAPTER 2

SYSTEM SPECIFICATIONS

2.1 HARDWARE SPECIFICATIONS

Processor	:	Intel i3 or higher
RAM	:	4GB (Minimum)
Hard Disk	:	100MB for project files

2.2 SOFTWARE SPECIFICATIONS

- Operating System: Windows/Linux
- Frontend: HTML, CSS, JavaScript
- Backend: Python (Flask)
- Visualization: Chart.js
- Editor: Visual Studio Code / Sublime

CHAPTER 3

MODULE DESCRIPTION

The system is divided into the following functional modules:

- **URL SCANNING MODULE:** Accepts URL, verifies safety, and provides UI feedback.
- **GRAPH MODULE:** Displays OS concept efficiency using line and bar graphs.
- **LOG MODULE:** Shows URL scan history and risk outcomes.
- **INTERFACE MODULE:** Handles smooth navigation and UI responsiveness.

CHAPTER 4

CODING

4.1 Style.css

```
body {
    font-family: 'Segoe UI', sans-serif;
    background-color: #f4f4f4;
    margin: 0;
    padding: 0;
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
}

.container {
    text-align: center;
    background-color: #fff;
    padding: 30px;
    border-radius: 12px;
    box-shadow: 0 0 15px rgba(0,0,0,0.1);
    width: 80%;
    max-width: 700px;
}
```

```
input[type="text"] {
    padding: 10px;
    width: 60%;
    margin-right: 10px;
    border: 2px solid #ccc;
    border-radius: 6px;
}

button, .nav-btn {
    padding: 10px 20px;
    margin-top: 15px;
    border: none;
    background-color: #007bff;
    color: white;
    font-weight: bold;
    border-radius: 6px;
    text-decoration: none;
    display: inline-block;
    cursor: pointer;
}

button:hover, .nav-btn:hover {
    background-color: #0056b3;
}
```

```
.result-box {
    margin-top: 20px;
    padding: 15px;
    font-size: 18px;
    border-radius: 8px;
}

.shake {
    animation: shake 0.3s;
}

@keyframes shake {
    0% { transform: translateX(0); }
    25% { transform: translateX(-5px); }
    50% { transform: translateX(5px); }
    75% { transform: translateX(-5px); }
    100% { transform: translateX(0); }
}

.popup {
    position: absolute;
    top: 20px;
    right: 20px;
```

```
    background: #ffc;
    padding: 10px;
    border: 1px solid #aaa;
    display: none;
    border-radius: 5px;
}
```

4.2 Server.py

```
from flask import Flask, request, jsonify, send_from_directory
from flask_cors import CORS
import os
app = Flask(__name__, static_url_path='', static_folder='.')
CORS(app)
# Load phishing list once
with open("phishing_feed.txt", "r") as f:
    phishing_list = [line.strip().lower().replace("https://", "").replace("http://", "").replace("www.", "").rstrip('/') for line in f.readlines()]


@app.route("/")
def root():
    return send_from_directory('.', 'index.html')


@app.route("/<path:path>")
def serve_file(path):
    return send_from_directory('.', path)
```

```
@app.route("/check_url")
def check_url():
    url = request.args.get("url", "").strip().lower()
    cleaned_url = url.replace("https://", "").replace("http://", "").replace("www.", "").rstrip('/')
    is_phishing = any(phish in cleaned_url for phish in phishing_list)
    with open("log.txt", "a") as log:
        log.write(f"Checked: {url} - {'PHISHING' if is_phishing else 'SAFE'}\n")
    return jsonify({"phishing": is_phishing})

@app.route("/log")
def get_log():
    if os.path.exists("log.txt"):
        with open("log.txt", "r") as f:
            return f.read()
    return "No logs found."

if __name__ == "__main__":
    app.run(debug=True)
```

4.3 Script.js

```
document.getElementById("urlForm").addEventListener("submit", function(e) {
    e.preventDefault();
    const url = document.getElementById("urlInput").value;
    const resultBox = document.getElementById("result");
    resultBox.textContent = "Checking...";

    fetch(`/check_url?url=${encodeURIComponent(url)}`)
        .then(response => response.json())
        .then(data => {
            if (data.phishing) {
                resultBox.textContent = "⚠ Phishing Detected!";
                resultBox.style.backgroundColor = "#ff4c4c";
                document.body.style.backgroundColor = "#ff4c4c"; // red background
                document.getElementById("result").classList.add("shake");
                new Audio('alert.mp3').play();
            } else {
                resultBox.textContent = "✅ Safe Site";
                resultBox.style.backgroundColor = "#4caf50";
                document.body.style.backgroundColor = "#4caf50"; // green background
            }
        });
});
```

4.4 log.js

```
document.addEventListener('DOMContentLoaded', function() {
    fetch('/log')
        .then(response => response.text())
        .then(data => {
            document.getElementById('log-content').textContent = data;
        });
});
```

4.5 graph.js

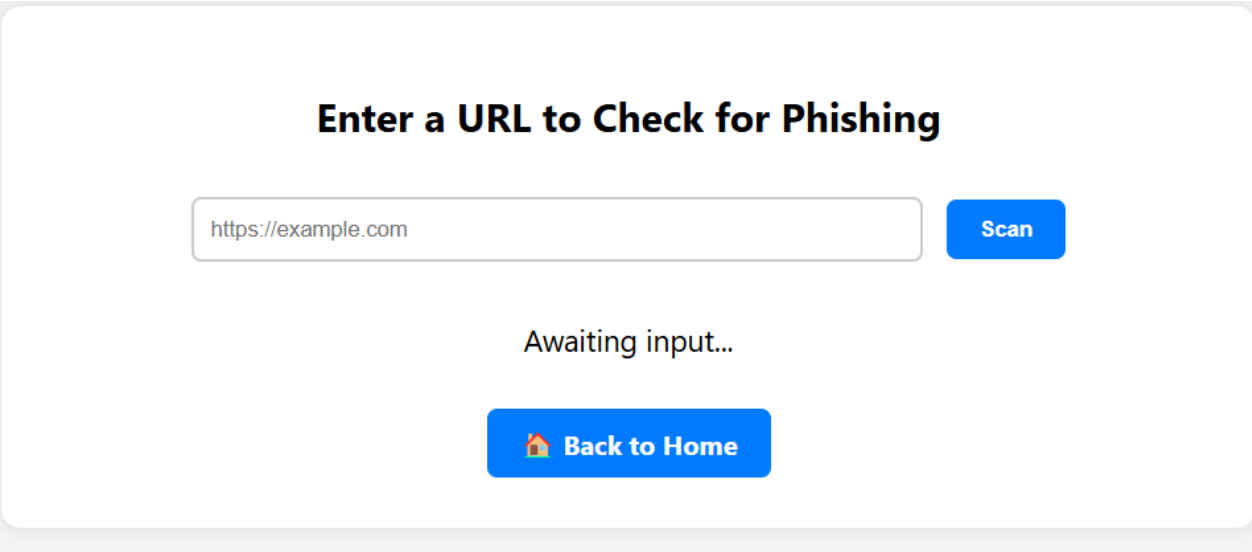
```
const ctx = document.getElementById('osChart').getContext('2d');
const popup = document.getElementById('osPopup');
const conceptData = {
    labels: ['Startup', 'URL Check', 'File Access', 'Logging', 'Response'],
    datasets: [{
        label: 'OS Concepts Activity',
        data: [2, 5, 3, 4, 3],
        borderColor: 'blue',
        fill: false
    }]
};
const chart = new Chart(ctx, {
    type: 'line',
    data: conceptData,
```

```
    options: {
        onClick: (e) => {
            const activePoints = chart.getElementsAtEventForMode(e, 'nearest', { intersect: true }, true);
            if (activePoints.length) {
                const index = activePoints[0].index;
                const label = conceptData.labels[index];
                const messages = {
                    'Startup': 'OS loads essential modules.',
                    'URL Check': 'Network stack and DNS involved.',
                    'File Access': 'Accessing local phishing feed file.',
                    'Logging': 'Writing result to log.txt.',
                    'Response': 'Generating JSON and returning via Flask.'
                };
                popup.textContent = messages[label];
                popup.style.display = 'block';
                setTimeout(() => popup.style.display = 'none', 3000);
            }
        }
    }
})
```

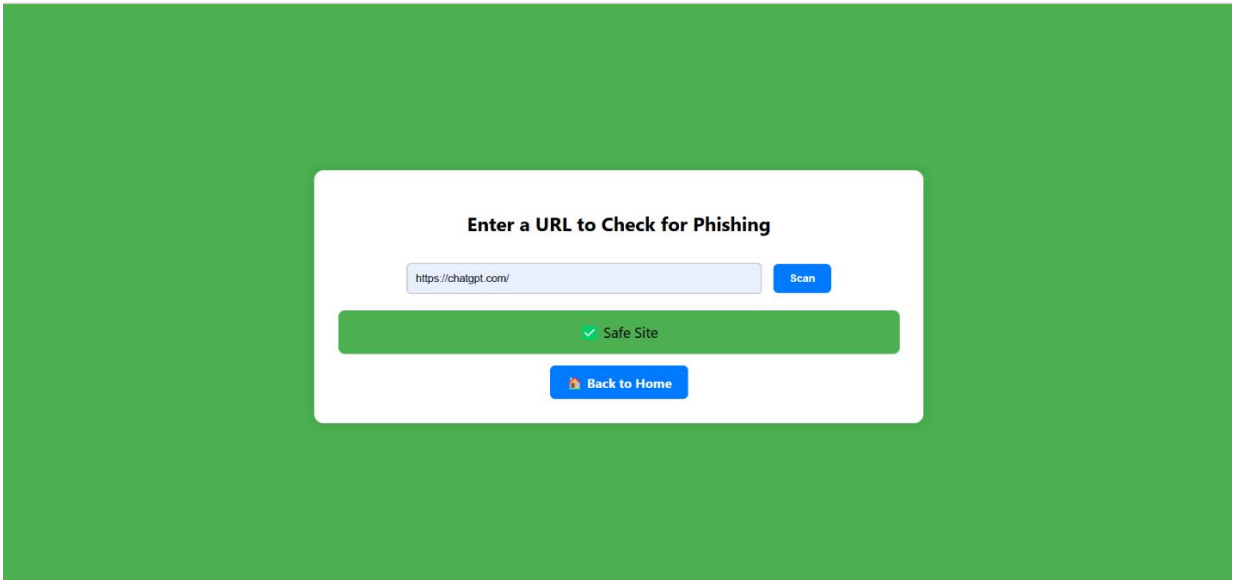
CHAPTER 5

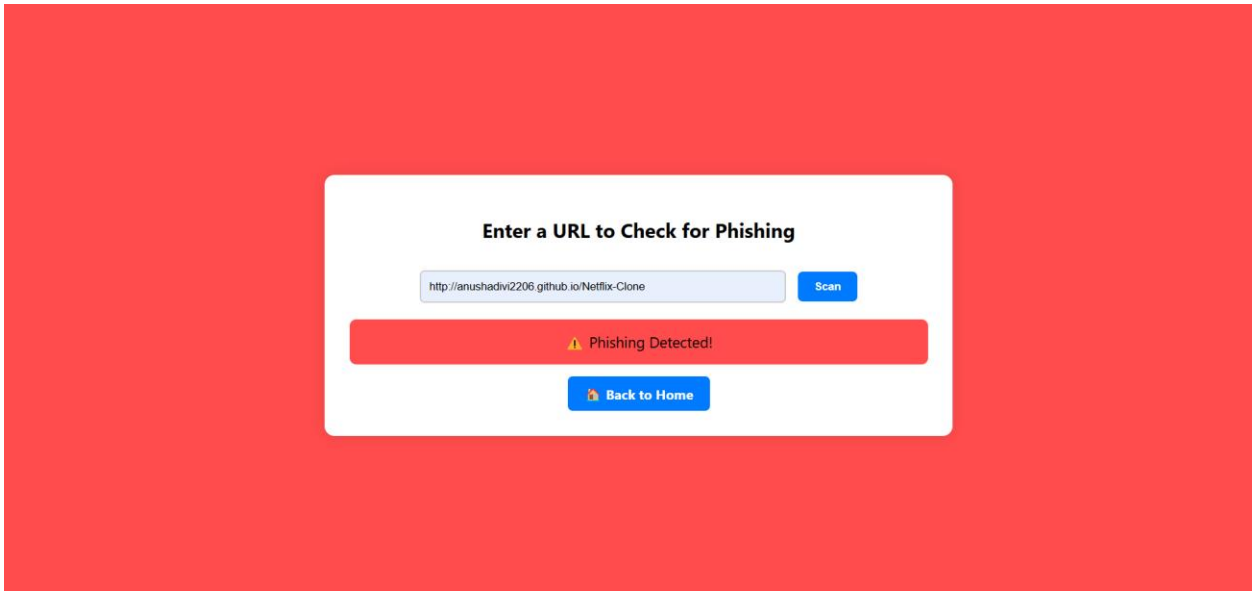
SCREENSHOTS

- **Homepage with navigation**

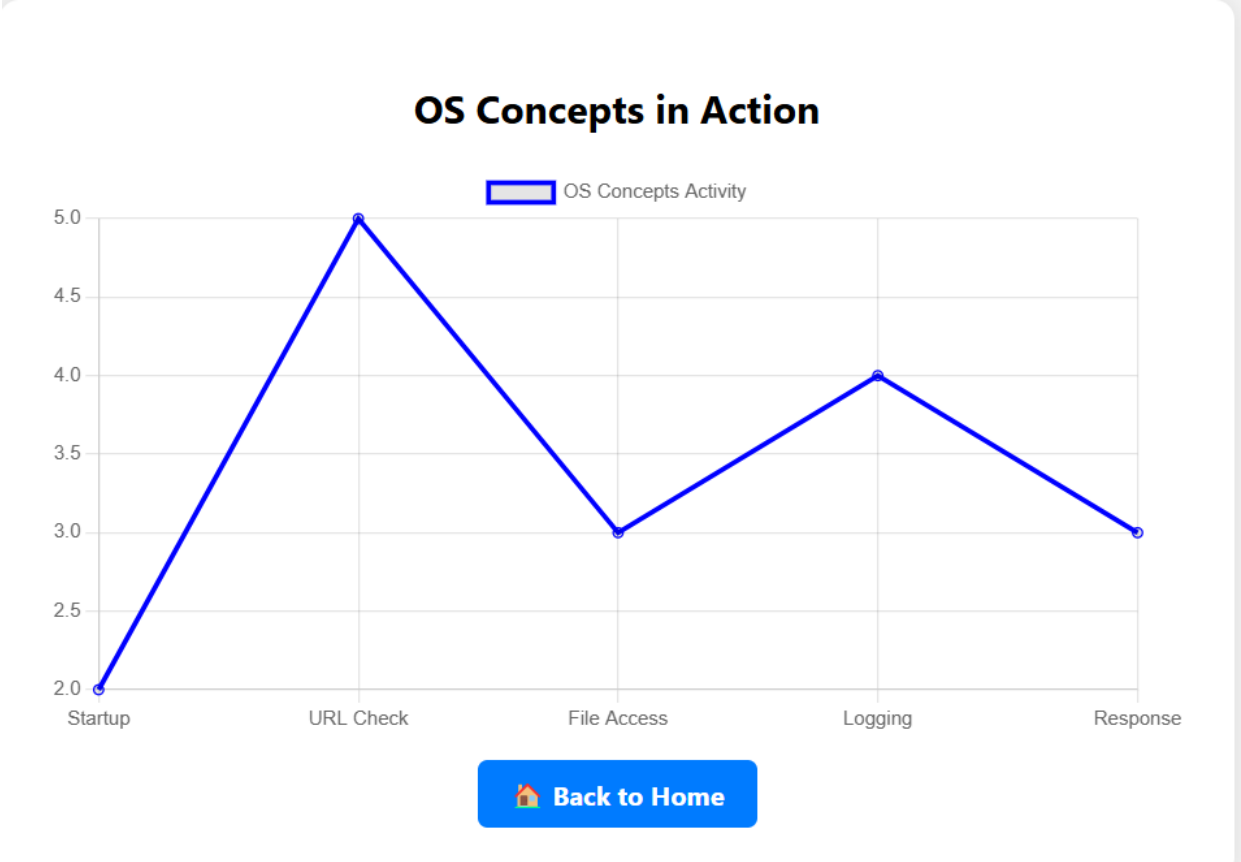


- **URL scanning interface (green/red background)**





- **Graph page (module performance)**



- **Log history page**

CHAPTER 6

CONCLUSION AND FUTURE ENHANCEMENT

This system successfully detects phishing links, notifies users through multiple senses, and logs the results for reference.

Future Enhancements:

- Integrate machine learning for predictive phishing detection.
- Enable scanning of email-based phishing links.
- Support for cloud-based logging and monitoring.
- Add role-based dashboards for admin and users.
- Include browser plugin integration for continuous protection.
- Enhance log visualization with charts and filtering options.

REFERENCES

1. **https://openphish.com/feed.txt**
2. **https://developer.mozilla.org/**
3. **https://flask.palletsprojects.com/**
4. **https://www.chartjs.org/**
5. **Python Official Documentation**
6. **Web Security Essentials by O'Reilly**