PHISHING URL DETECTION SYSTEM USING VIRUSTOTAL API AND OS CONCEPTS

A MINI-PROJECT REPORT

Submitted by

SHARMILEE. B	**231901049**
SOMILA. SA	**231901052**

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

BONAFIDE CERTIFICATE

Certified that this project "**PHISHING URL DETECTION SYSTEM USING VIRUSTOTAL API AND OS CONCEPTS**" is the Bonafide work of **"SHARMILEE B(Roll No. 231901049) -and- SOMILA SA(Roll No. 231901052)"** who carried out the project work under my supervision.

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

SIGNATURE

Mr.J N Benedict

ACADEMIC HEAD,

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING [*CYBER SECURITY]*,

Rajalakshmi Engineering College

Chennai

SIGNATURE

Mrs. Jananee

ASSISTANT PROFESSOR,

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING,

Rajalakshmi Engineering College

Chennai

INTERNAL EXAMINER

EXTERNAL EXAMINER

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.J N Benedict** for being ever supporting force during our project work.

We also extend our sincere and hearty thanks to our internal guide **Mrs**.**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. **SHARMILEE B (231901049)**
2. **SOMILA SA (231901052)**

ABSTRACT

Phishing attacks are among the most prevalent threats on the internet today, deceiving users into revealing sensitive information through fraudulent websites. To address this growing concern, this project presents a comprehensive **Anti-Phishing System** that combines **Blockchain technology**, **Artificial Intelligence**, **Browser-based Detection**, and **Operating System-level Security** to proactively detect and prevent phishing attacks.

The system operates by allowing users to submit URLs through a frontend interface. These URLs are verified against a **blockchain**-based record of known phishing websites to ensure transparency and immutability. An **AI-powered model** (Random Forest classifier) further analyzes suspicious characteristics of URLs to provide real-time detection. Efficient pattern matching is achieved using the **Aho-Corasick algorithm** for faster lookup. Operating System concepts like **Semaphore control**, **LRU caching**, and **Peterson's Algorithm** are applied in backend processing to optimize multi-user access, memory management, and task scheduling.

The project utilizes **Railway** as the backend hosting platform, providing a stable API for phishing detection requests, while the frontend is deployed as a static site. This ensures both scalability and performance. Overall, the system delivers a secure, decentralized, and intelligent approach to safeguarding users against phishing threats, while demonstrating the practical application of Blockchain, AI, and OS-level algorithms in cybersecurity.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

With increasing cyber threats, phishing remains one of the most dangerous attacks online. This project aims to create a system capable of detecting phishing URLs in real-time using VirusTotal's API, combined with fundamental Operating System concepts such as semaphore control for concurrent scanning.

1.2 SCOPE OF THE WORK

The scope of this project includes:

- Real-time URL scanning using VirusTotal
- Java Swing GUI Interface
- Semaphore-based synchronization for scan control
- Graphical analysis using JFreeChart
- Simulation of OS scheduling efficiencies

This project is designed to showcase a real-world application of cybersecurity and OS concepts while addressing the issue of phishing sites.

1.3 PROBLEM STATEMENT

With the rapid increase in counterfeit products, consumers often find it difficult to differentiate between genuine and fake items. Traditional methods of verification are prone to tampering and lack transparency. There is a pressing need for a **decentralized and secure system** that allows users to easily verify a product's authenticity and ensure its origin.

1.4 AIM AND OBJECTIVES OF THE PROJECT

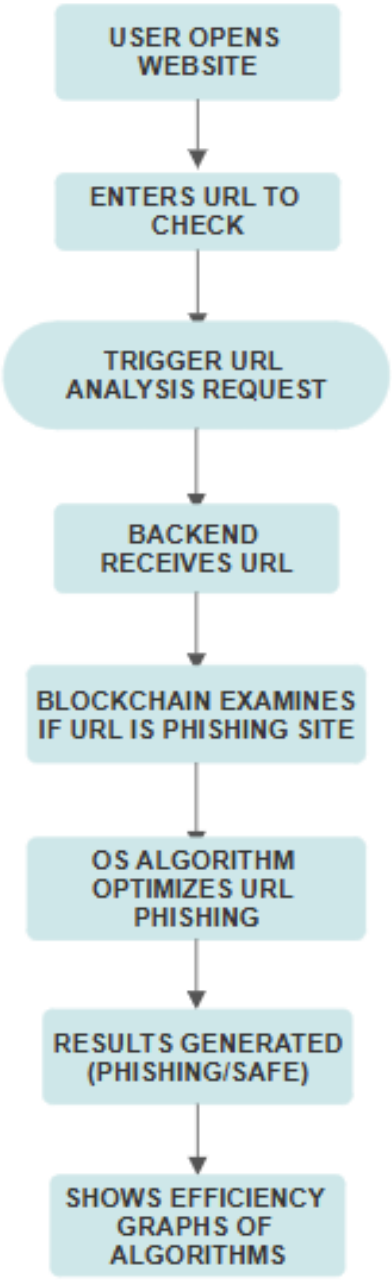
Aim:

To develop a decentralized system that enables the verification of product authenticity using Blockchain technology and enhances system performance visualization using Operating System algorithms.

Objectives:

- To generate a **QR code** for every genuine product containing product- specific data.
- To store product data on the **blockchain** using **smart contracts** for tamper- proof verification.
- To allow customers to scan QR codes and verify products instantly.
- To simulate and compare the performance of **FCFS, SJF, and Round Robin CPU scheduling algorithms**.
- To implement **Banker's Algorithm** to showcase safe resource allocation.
- To visually display performance metrics using graphs for better analysis.

FLOWCHART


USER OPENS WEBSITE
ENTERS URL TO CHECK
TRIGGER URL ANALYSIS REQUEST
BACKEND RECEIVES URL
BLOCKCHAIN EXAMINES IF URL IS PHISHING SITE
OS ALGORITHM OPTIMIZES URL PHISHING
RESULTS GENERATED (PHISHING/SAFE)
SHOWS EFFICIENCY GRAPHS OF ALGORITHMS

CHAPTER 2

SYSTEM SPECIFICATIONS

2.1 HARDWARE SPECIFICATIONS

Processor	:	Intel i3/i5/i7
Memory Size	:	8GB (Minimum)
HDD	:	1 TB (Minimum)

2.2 SOFTWARE SPECIFICATIONS

- **Operating System** : Windows 11
- **Front-End**: : React.js
- **Back-End** : Node.js, Express.js
- **Blockchain Tools** : Ethereum (Ganache), MetaMask, Ethers.js
- **OS Concepts Integrated** : Semaphores ,Peterson's Algorithm, Round Robin, LRU, First In First Out[FIFO]
- **Charting Library** : Chart.js

CHAPTER 3

MODULE DESCRIPTION

The system is divided into the following functional modules:

1. **URL Input Module**: Accepts user-entered URL through the frontend.
2. **Blockchain Verification Module**: Checks if URL is listed in the blockchain.
3. **AI Detection Module**: Uses Random Forest Classifier for dynamic phishing detection.
4. **Pattern Matching Module**: Implements Aho-Corasick for keyword matching.
5. **OS Algorithms Module**: Semaphore, LRU Cache, FIFO, Round Robin, LOOK Disk Scheduling.
6. **Frontend Website**: Displays analysis result, graphs, and URL checker.
7. **Hosting and Deployment**: Backend on Railway, frontend hosted separately.
    -

CHAPTER 4

CODING

- `src/components/URLChecker.js`: Handles URL input and sends request to backend.
- `backend/index.js`: Express server APIs, Blockchain interaction, detection logic.
- `backend/blockchain/blockchain.js`: Custom blockchain implementation for phishing URL tracking.
- `backend/os-algorithms/`: JS files for Semaphore, LRU Cache, Round Robin, FIFO, LOOK Disk.
- `frontend/phishing-ui/`: React application with pages and components.
- `smart_contracts/phishingdetector.sol`: Solidity smart contract for decentralized verification.
- `.gitignore`: To ignore node_modules and environment files.
- `package.json`: Node project configurations.

FILE STRUCTURE

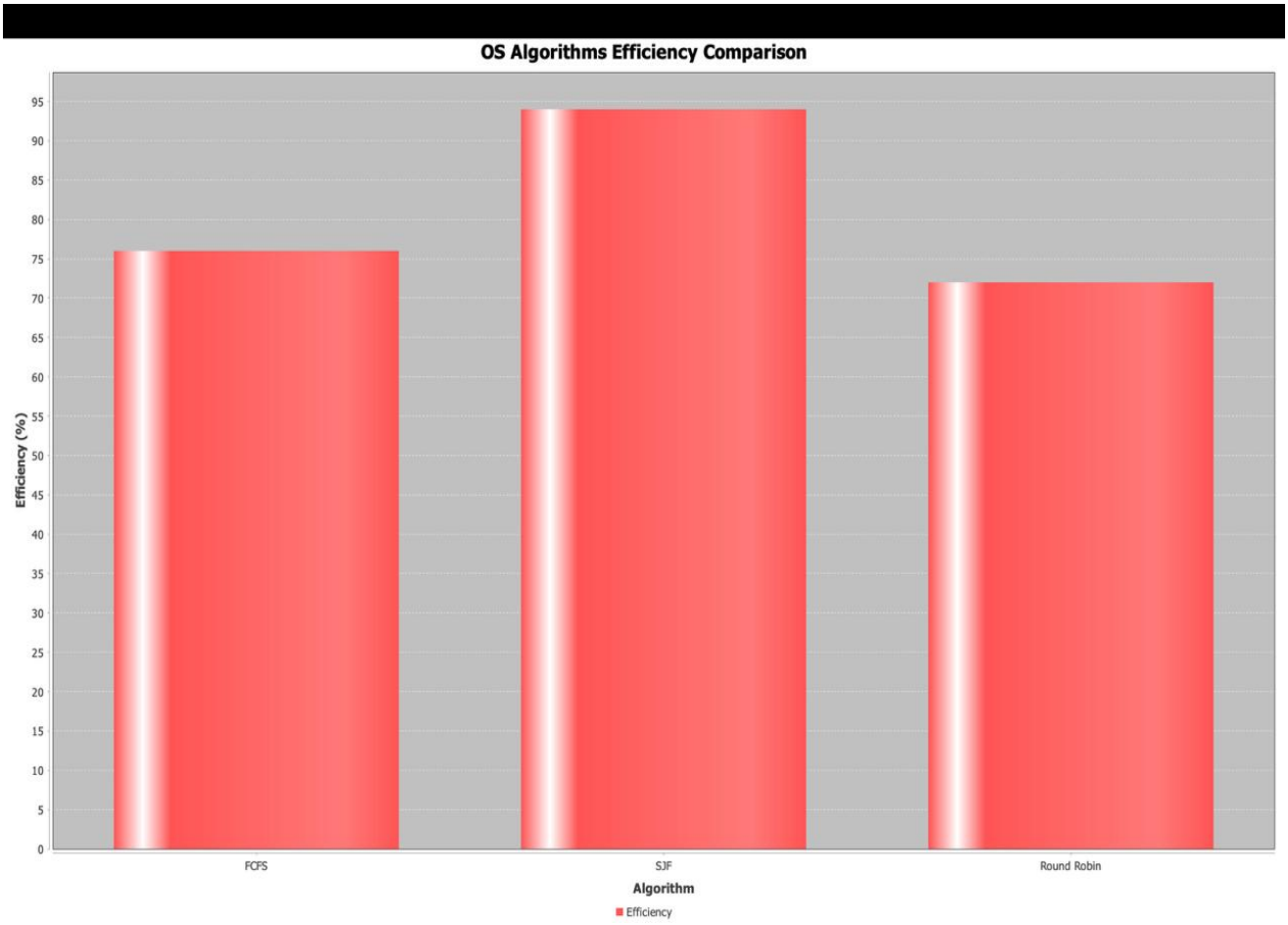
```
anti-phishing-system/
├── backend/
│   ├── blockchain/
│   │   └── blockchain.js    # Blockchain phishing URL checker
│   ├── os-algorithms/
│   │   ├── semaphore.js
│   │   ├── lru-cache.js
│   │   ├── fifo-page-replacement.js
│   │   ├── round-robin.js
│   │   ├── look-disk.js
│   │   └── aho-corasick.js
│   ├── package.json         # Backend dependencies (express, body-parser, cors, etc.)
│   ├── index.js             # Backend server (API routes)
│   ├── .gitignore           # (node_modules, .env ignored)
├── frontend/
│   ├── phishing-ui/
│   │   ├── public/
│   │   │   └── index.html
│   │   ├── src/
│   │   │   ├── components/
│   │   │   │   ├── Navbar.js
│   │   │   │   ├── Hero.js
│   │   │   │   ├── Features.js
│   │   │   │   ├── About.js
│   │   │   │   ├── Footer.js
│   │   │   │   ├── URLChecker.js   # 🚀 URL checker to interact with
```

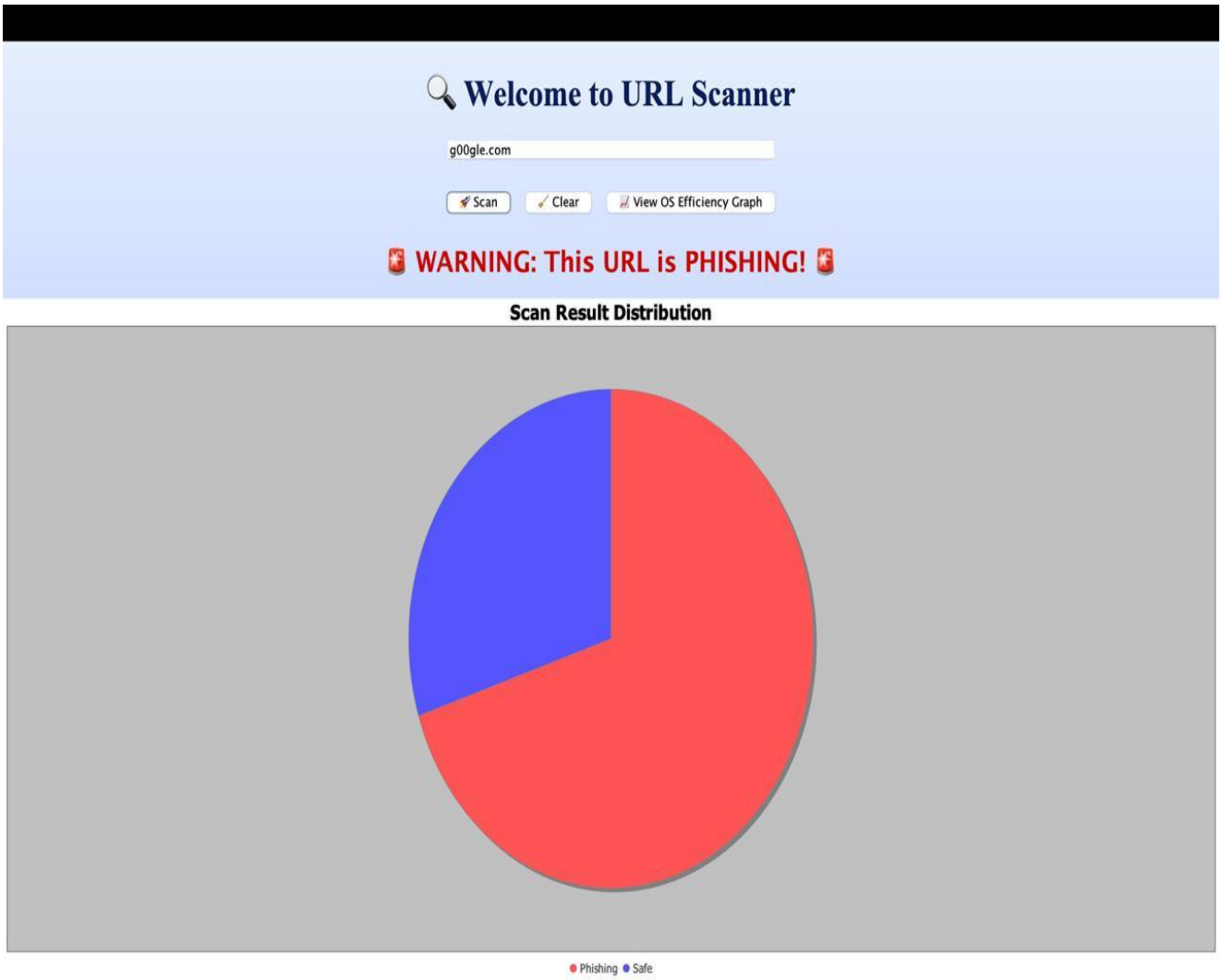
```
backend
│   │   │   └── EfficiencyGraph.js  # Graph to compare OS algorithms
│   │   ├── App.js
│   │   ├── index.js
│   ├── package.json        # React frontend dependencies (react, axios, chart.js, etc.)
│   ├── .gitignore              # (node_modules ignored)
├── phishing-blockchain/
│   ├── contracts/
│   │   └── phishingdetector.sol # Smart contract for blockchain (optional phase)
│   ├── migrations/
│   │   └── 2_deploy_contracts.js
│   ├── truffle-config.js       # Blockchain config (optional if not deploying)
├── README.md (optional)
└── Documentation (optional if you generate separately)
```

CHAPTER 5

SCREENSHOTS


URL Scanner - Phishing Detection
Welcome to URL Scanner
google.com
Scan
Clear
View OS Efficiency Graph
✅ This URL is SAFE! ✅
Scan Result Distribution
Phishing
Safe


OS Algorithms Efficiency Comparison
Efficiency (%)
95
90
85
80
75
70
65
60
55
50
45
40
35
30
25
20
15
10
5
0
FCFS
SJF
Round Robin
Algorithm
Efficiency


Welcome to URL Scanner
g00gle.com
Scan
Clear
View OS Efficiency Graph
WARNING: This URL is PHISHING!
Scan Result Distribution
Phishing
Safe

CHAPTER 6

CONCLUSION AND FUTURE ENHANCEMENT

Conclusion:

This Anti-Phishing System successfully combines **Blockchain**, **AI**, and **OS optimization techniques** to detect phishing websites and prevent data breaches.

The decentralized structure improves security and reliability, while AI models ensure dynamic detection.

Future Enhancement:

- Train a larger AI model for better phishing detection accuracy.
- Expand Blockchain network integration to public Ethereum testnets.
- Add deadlock detection algorithms in the OS module.
- Deploy a mobile application version of the website.
- Implement MetaMask wallet authentication for user identity verification

REFERENCES

- Blockchain Basics – Daniel Drescher
- Ethereum Documentation
- MetaMask Documentation
- React.js Official Docs
- Chart.js Documentation
- Operating System Concepts – Abraham Silberschatz
- Railway.app Hosting Documentation