



THE YENEPOYA INSTITUTE OF ARTS SCIENCE COMMERCE AND MANAGEMENT

(a constituent unit of Yenepoya Deemed to be University)

**Email Security and Anti-Spam Solutions**

# PROJECT SYNOPSIS

MASTER OF COMPUTER APPLICATIONS

Presented by : Guided by :

Berlin Jacob 24MCA205 Mr. Shashank

Helen Mary 24MCA210

Karpaga Dharshini K 24MCA213



**TITLE PAGE**

1. Name of the student: Helen Mary
2. Class Roll No. 24MCA210
3. Campus ID: 35862
4. Present official Address: YIASCM Balmatta, Mangalore 575002
5. Email: [35862@yenepoya.edu.in](mailto:35862@yenepoya.edu.in)
6. Phone No. +91 8848521484
7. Branch: Computer Science
8. Batch: 2024-2026
9. Proposed Topic: Email Security and Anti-Spam Solutions



# TABLE OF CONTENTS

Cover page - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - 1

Title - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - 2

Content - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - 3

1. Introduction - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - 4
   1. **Key Features**
   2. **Technology Stack**
   3. **Specialized Field: Cybersecurity and Ethical Hacking**
2. Methodology- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - 5
   1. **Requirement Analysis and Tool Selection**
   2. **System Architecture and Design**
   3. **Frontend Development (Tkinter)**
   4. **Backend Integration**
   5. **Final Testing and Documentation**
3. Facilities required for proposed work - - - - - - - - - - - - - - - - - - 6
   1. **Development Environment**
   2. **Vulnerability Scanning Tools**
   3. **Testing and Deployment**
   4. **Reporting Tools**



1. **Introduction**

Email communication is a critical part of modern enterprises but is often exploited for malicious purposes, such as spam, phishing, and malware delivery. This project aims to build a comprehensive system that safeguards email communications using intelligent content analysis and integrated malware scanning. Leveraging Natural Language Processing (NLP), machine learning, and third-party threat detection APIs, the system can identify, block, and report malicious emails in real time.

1**.1 Key Features**

 Spam and phishing detection using NLP-based content filtering.

 Malicious URL and attachment scanning via VirusTotal API.

 Rule-based and blacklist filtering.

 Secure IMAP/SMTP integration for real email fetching.

 Admin dashboard for reviewing flagged emails and setting policies.

 Manual override option for wrongly flagged emails.

 Real-time threat statistics and reporting.

1.2 **Technology stack**

* + - Programming Language: Python
    - Libraries/Frameworks: Scikit-learn, NLTK/spaCy, Tkinter, Flask/Django
    - OpenVAS: For in-depth vulnerability assessment.
    - OWASP ZAP: For web application security scanning.
    - Metasploit: For automating exploitation.
    - SQLMap: For detecting and exploiting SQL injection vulnerabilities.



* + - CVE Integration: Fetches the latest CVE data from the NVD, linking vulnerabilities with their respective CVE entries.
  1. **Specialized Field: Cybersecurity and Ethical Hacking**

This project falls under Cybersecurity, with a strong emphasis on ethical hacking and threat analysis. It provides practical insights into how attackers exploit emails and how defenders can build automated tools to detect and neutralize threats effectively.



1. **Methodology**

The system will be developed in incremental stages using an agile approach:

1. Requirement analysis and tool selection.

2. Building NLP models and integrating VirusTotal API.

3. Designing UI and backend.

4. Testing with real email samples.

5. Final documentation and deployment.

* 1. **Requirement Analysis and Tool Selection**

Requirements were determined based on real-world email threats. Tools like NLTK, VirusTotal, and Scikit-learn were chosen for their proven effectiveness in email content and threat analysis. Tkinter was selected for creating a simple GUI.

* 1. **System Architecture and Design**

The architecture includes:

* Input Layer: Email retrieval via IMAP.
* Analysis Layer:NLP model, blacklist, VirusTotal scanner.
* Decision Layer: Flag, allow, or quarantine email.
* Presentation Layer:Admin UI for review and statistics.

A modular, service-oriented architecture ensures scalability and maintainability.

* 1. **Frontend Development (Tkinter)**

The desktop interface allows users to:

* Set filtering rules.
* View flagged emails.



* Review threat statistics.
* Perform manual overrides.

Tkinter was chosen for its simplicity and integration with Python.

* 1. **Backend Integration**

The backend handles:

* Email fetching and parsing.
* NLP classification and API-based scanning.
* Database logging of emails and results.
* Communication between frontend and backend via Flask routes (or within a single Python app using threading).
  1. **Final Testing and Documentation**

Comprehensive testing includes:

* Unit tests for NLP components.
* API response validation.
* Accuracy testing on real phishing/spam datasets.

Documentation will cover setup, usage, and extension guides.



1. **Facilities Required For Proposed Work**

* Internet access for API and email server connectivity.
* Access to VirusTotal API (free key or academic license).
* Sample datasets of spam and phishing emails.
* Secure test email account (IMAP/SMTP).
  1. **Development Environment**

Hardware: Standard laptop with 8GB+ RAM.

Software: Python 3.x, required libraries, IDE (VS Code/PyCharm), SQLite/MySQL, VirusTotal API key.

* 1. **Vulnerability Scanning Tools**

VirusTotal API for file and URL scanning.

Optionally integrate with tools like SpamAssassin or PhishTank for broader detection.

**3.3 Testing and Deployment**

Unit and integration testing using custom and public email datasets.

Testing performance on real-time email streams.

Deployment in a secure, sandboxed environment or as a desktop application for SMEs.

* 1. **Reporting Tools**
* Built-in dashboard for logging and visualization of threats.
* Graphs and statistics on:
* Number of threats detected.
* Type of threat (phishing, malware, spam).
* Time-based trends.