A picture containing application

Description automatically generated

**CIT499 Senior Project**

**Email Spam Detection Using Machine Learning**

**Algorithms**

# **Table of Contents**

[Abstract 3](#_Toc182344804)

[Chapter 1: Introduction 5](#_Toc182344805)

[1.1 Overview 5](#_Toc182344806)

[1.2 Motivation 7](#_Toc182344807)

[1.3 Aim and Objectives 8](#_Toc182344808)

[1.4 Scope of the Study 8](#_Toc182344809)

[1.5 Significance of the Project 8](#_Toc182344810)

[1.6 Structure of the report 9](#_Toc182344811)

[CHAPTER 2: LITERATURE REVIEW 11](#_Toc182344812)

[2.1 Introduction 11](#_Toc182344813)

[2.2 Digital Forensics (DF) 11](#_Toc182344814)

[2.2.1 DF Subdomain 12](#_Toc182344815)

[2.2.2 Digital Investigation 12](#_Toc182344816)

[2.3 Email System 13](#_Toc182344817)

[2.3.1 Email Structure 13](#_Toc182344818)

[2.3.2 Email Analysis 13](#_Toc182344819)

[2.3.3 Crime Using Email 14](#_Toc182344820)

[2.4 Email Forensics 14](#_Toc182344821)

[2.4.1 Header Forensics 15](#_Toc182344822)

[2.4.2 Body Forensics 15](#_Toc182344823)

[2.4.3 Signature Forensics 16](#_Toc182344824)

[2.5 Challenges in Email Forensics 16](#_Toc182344825)

[2.6 Summary 16](#_Toc182344826)

[CHAPTER 3: Methodology 18](#_Toc182344827)

[3.1 Overview 19](#_Toc182344828)

[3.2 Operational Framework 19](#_Toc182344829)

[3.3 Gemini Large Language Model 19](#_Toc182344830)

[3.4 Prompt Engineering 20](#_Toc182344831)

[3.5 Geolocation with IPinfo.io API 20](#_Toc182344832)

[3.6 Functional and Non-Functional Requirements 20](#_Toc182344833)

[3.6.1 Functional Requirements: 20](#_Toc182344834)

[3.6.2 Non-Functional Requirements: 21](#_Toc182344835)

[3.7 Diagrams and Framework Components 22](#_Toc182344836)

[3.7.1 Activity Diagram 22](#_Toc182344837)

[3.7.2 Block Diagram 23](#_Toc182344838)

[3.7.3 Entity Diagram 24](#_Toc182344839)

[3.7.4 Sequence Diagram 25](#_Toc182344840)

[3.7.5 Use Case Diagram 26](#_Toc182344841)

[3.8 Web Application 27](#_Toc182344842)

[3.8.1 Landing Page 27](#_Toc182344843)

[3.8.2 About Page 28](#_Toc182344844)

[3.8.3 Sign-Up / Sign-In Page 29](#_Toc182344845)

[3.8.4 Contact Page 30](#_Toc182344846)

[CHAPTER 4: Implementation and results 32](#_Toc182344847)

[4.1 Implementation 33](#_Toc182344848)

[4.2 Tools Used 33](#_Toc182344849)

[4.2.1 Gemini LLM 33](#_Toc182344850)

[4.2.2 LangChain Integration 33](#_Toc182344851)

[4.2.3 IPinfo.io API for Geolocation 33](#_Toc182344852)

[4.2.4 Flask Framework 34](#_Toc182344853)

[4.2.5 Frontend Technologies (HTML, CSS, JavaScript) 34](#_Toc182344854)

[4.2.6 SQLite for Data Storage 34](#_Toc182344855)

[4.3 Gemini Pro: LLM Model for Text Analysis 34](#_Toc182344856)

[4.4 Testing 35](#_Toc182344857)

[CHAPTER 5: Future Work and Recommendations 37](#_Toc182344858)

[5.1 Future Work 38](#_Toc182344859)

[5.2 Recommendations 39](#_Toc182344860)

[References 41](#_Toc182344861)

# **Abstract**

The application here is a Flask web application for detecting spam mails by scanning the contents and metadata of emails. The user can fill out an email address, body and footer that Google’s Gemini LLM (integrated with LangChain) eats up. This language model runs a comprehensive scan of the site — flagging possible spam patterns based on the language usage, greeting styles and footer text. Furthermore, application accesses ipinfo.io API to get the geographical position of platform user for better contextualization in the output. JWT authentication is used to secure access and user data is stored on SQLite database. This technique fuses natural language with geolocation to deliver comprehensive spam detection output and user-friendly interface.  
  
**Keywords**: Spam Recognition, Flask, LangChain, Gemini LLM, Natural Language Analysis, Email Recognition, Geolocation, JWT Authentication.

# **Chapter 1: Introduction**

## **1.1 Overview**

With the development and advancement of information and communication technology in the recent decades has led to the emergence of new application areas and use cases, one of the most common use cases of ICT technologies is in the communication domain (Graafland, 2018). For example, people make use of ICT technologies to communicate with each other, the most common type of this communication takes place through emails and email systems. Email is the most widely used form of communication between individuals, businesses and governments around the world. Today billions of emails are sent and received around the world which makes the emails one of the most important tools for communication between businesses, individuals and organizations (Patel, 2021).

Due to the rise and wide spread use of the emails has led to another problem which is commonly referred to as spam email. Spam email are unsolicited mails that are sent out to users in huge numbers without taking their consent and authorization. The purpose of spam emails ranges from products advertisements and services information, commercial and sometimes fraudulent schemes, malwares and many other purposes. The number of spam emails can overload the email inbox and often leads to significant security threats such as personal information theft, financial loss and many other forms of information security risks (Bujang, 2013).

In the current situation the issue of spam email is growing day by day and it makes a huge portion of email traffic, spam email generation and use is very simple and low cost and email services providers often blame and target the spammers for over traffic on their networks. Spam email can be classified into different types and categories, some of the most widely spams are promotional spams, phishing spams and malware spams. The promotional spams are marketing emails that only promote a business, or product or any other business product; these emails are sent without the user's consent. The phishing email is used to steal information or trick the users into providing their personal as well as financial details. The malware spams are sent with a purpose in order to inject a piece of malware code in order to perform different actions on the target's systems (Broadhurst, 2020).

Spammars uses different methods and techniques in order to obtain the emails address of the users in order to send emails, one of the most widely used method employed by the spammers to collect email address is the email harvesting, email harvesting is a technique to scrape the user email address from the websites, social networking sites and many other types of application. The spammers collect email addresses by using the email harvesting and then use those emails to send spam emails. Another method spammers use for email addresses is to purchase email lists, some companies collect the user email address and then sell those email addresses to spammers without their consent. Apart from this spammers use other advanced techniques such as email spoofing and randomization and obfuscation methods to avoid the detection of email service providers spam email filters.

In order to resolve the issue of spam email by the email service providers, different methods and approaches have been developed and implemented over the time, the most conventional method that has been used is the rule based system. The rule based system works by scanning the incoming emails. The incoming are scanned through the rules set in the system and filters the emails based on the outcome of the rules. This approach has several limitations and drawbacks because this approach is unable to detect and filter spams deployed by spammers using the advanced methods (Crawford, 2015).

Another powerful and effective method for spam filters is to implement spam filters and detection methods using machine learning algorithms. In this type of spam detection mechanism, the machine learning algorithms scans and processes large amounts of data sets in order to detect the patterns and information that resembles the spam emails. Some of the most widely used machine learning algorithms are support vector machines, decision trees, naïve Bayes, convolutional neural networks and many other types of algorithms. These methods and algorithms classify the spam email with greater accuracy and effectiveness as compared to the rules based systems.

However, due to the advancement in the spamming tools and approaches it becomes very difficult for a single system or method to detect every type of spam email, in order to make the spam detection and filtration more effective, the combination of different approaches and methods are important to effectively detect spam emails (Mohammad, 2024). Other parameters of emails such as IP address, and header information can also play an important role in detecting the spam emails. For example, some email services use blacklisting of IP addresses to spam emails, in order to allow the trusted IP address to bypass filters is called whitelisting of IP Addresses. Parameters like time zone and origin of the email can also improve the spam email detection and filtration.

Email spam involves the sending of unsolicited emails, typically in bulk, to various recipients without their consent. This chapter introduces the concept of email spam, the methods employed by spammers, and the combination of traditional machine learning-based methods with tools to identify the sender’s specifics, such as IP address, geographical location, and time zone for enhanced spam detection.

## **1.2 Motivation**

The basic intuition and motivation behind the proposed project of spam filtering and detection using machine learning is to develop effective methods that can limit the rapid increase of spam emails and its impact on personal and organizational security. Currently email is one of the most widely used means of communication, however, due to the significant rise and impact of spam emails has led to a very stressful problem that can have huge impact on businesses as well as individuals.

 Email spam not only contributes to the rise in email traffic on the internet, but poses multidimensional threats and security risks in today's connected world. Spam emails are not only used in promotional content and advertising, spam emails are used in malware attacks, phishing emails and targeting ordinary users on the internet and stealing their financial and other sensitive information (Siddique, 2021).

To counter the issue of widespread spam emails we have come up with the idea to develop an effective method for detecting spam emails using the advanced tools and methods such as machine learning algorithms. As the existing rules based systems for email detection and filtration are not sufficient to counter the threats and techniques applied by the spammers to send spam emails. in order to develop a more robust and effective method for spam email filtering and detection, other important email parameters such as email headers information, geolocation and origin of the email and temporal data can be included and be taken into consideration for detecting the spam filters. For example, Geolocation or origin of the email can play a vital role in classifying an email whether it is a spam or genuine email. For instance, if a company sends emails from one specific location every time and if an email comes from the same company address but with different location and origin, this behavior can indicate and classify an email as spam. Further information such as the author can also be considered for classifying an email as spam or genuine email (Yaseen, 2021).

To overcome the existing issues and to limit the spread of spam email to undertake this spam detection system that uses state of the art machine learning algorithm and other important parameters such as geolocation and email origin to classify an email as spam or genuine email.

.

## **1.3 Aim and Objectives**

Aim: To develop an advanced email spam detection system utilizing machine learning algorithms augmented with tools to identify sender’s IP address, geographical location, and time zone, to provide a comprehensive solution against email spam.

Objectives:

* To review and implement various machine learning algorithms for spam detection and to evaluate the effectiveness of these algorithms in detecting different types of email spam.
* To integrate tools for identifying the IP address, geographical location, and time zone of the sender to enhance the detection process and provide contextual security insights.
* To assess the feasibility and added value of including sender’s location and timing data in improving spam classification accuracy and preventing phishing attacks.

## **1.4 Scope of the Study**

The scope of this study encompasses the application of machine learning algorithms to detect spam emails, augmented with the capability to analyze and utilize the sender’s geographical and temporal data for a more sophisticated detection and analysis system.

## **1.5 Significance of the Project**

This Project is significant as it not only addresses the growing issue of email spam using advanced machine learning techniques but also innovates by incorporating the sender’s geographical and temporal data into the detection process. This approach aims to enhance the robustness and reliability of spam detection mechanisms, providing a multi-layered security framework that can adapt to different organizational needs. The following are the main stackholders that are invloved in this project. Some of the direct stackholders related to this project are the ordinary users, email service providers, business and orgnizations and the technical as well academic communities.

## **1.6 Structure of the report**

This report is structured as follows.

* + 1. **Chapter 1: Introduction**

The following are the main content and secitons inlcuded in first chapter,

* Introduction and Problem Identification: Discusses the rising issue of email spam with the growth of internet usage and its impacts on security and productivity.
* Benefits of the Project: Outlines how the project will help in accurately detecting spam emails, enhancing security, and improving email management.
* Project Objectives and Modifications: Details the primary goals of the project, including the development and implementation of a machine learning model for spam detection, and notes any modifications from the original project plan.
* Project Objectives: Specifies the targeted outcomes of the project, such as improved spam detection rates and reduced false positives.
* Project Plan and Milestones: Provides a timeline of the project, including key milestones from inception to completion, and scheduled reviews.
* Future Plan and Deliverable: Describes future enhancements and potential expansions of the project, along with the final deliverables.
  + 1. **Chapter 2: Literature Review**
* Chapter two discuss and presents a detailed overview of the literature review conducted in the spam filteration and detection directions.
  + 1. **Chapter 3: System analysis and Desgin specification**
* Description of the Delivered Product/Service: Offers a detailed description of the final product or service, including its features, functionalities, and user interface.
* Data Flow Diagram: Includes a diagram that illustrates the flow of data through the system, highlighting how inputs are processed, and outputs are generated.
* Prototype: Describes the prototype developed during the project, including its operational logic and the technologies used.
  + 1. **Chapter 4: System Implementation and Development**
* Review/Analysis/Test of Product/Service: Summarizes the testing and quality assurance processes used to validate the effectiveness and reliability of the final product.
* Resources: Lists the resources used during the project, including software tools, hardware, and datasets.
* Scenario: Provides a hypothetical scenario to demonstrate how the product/service can be used in a real-world situation.
  + 1. **Chapter 5: Lessons Learned and Recommendations**
* Lessons Learned and Recommendations: Reflects on the challenges faced during the project and offers recommendations for future projects based on these experiences.
* Tools and Definitions: Details the tools and technologies used in the project, along with definitions of key terms and concepts.
  + 1. **Chapter 6 : Conclusion and Future Work**
* Conclusion and Future Work: Concludes the report with a summary of findings and discusses potential future work to build upon the project’s success.
* References: Lists all the sources and references used to gather information and guide the project development.
* Appendices: Provides additional material that supports the main text, such as raw data, code snippets, and detailed charts.

# **CHAPTER 2: LITERATURE REVIEW**

## **2.1 Introduction**

The increase in digital communication has led to the widespread use of emails for personal and professional purposes. This growth in email usage has brought about a significant increase in email spam, which represents a range of problems from minor annoyances to major security threats. Email spam includes unsolicited messages sent in bulk, often with harmful content such as advertisements, malware, or phishing scams intended to deceive the recipients. Effective solutions are necessary to tackle these issues.

Traditionally, spam detection started with simple methods like filtering specific keywords or blocking known harmful senders. However, as spammers evolved, these methods became less effective. This shift necessitated more dynamic solutions that use advanced computational techniques such as machine learning and artificial intelligence.

Machine learning has proven particularly effective in spam detection. It uses large sets of emails to learn and identify patterns that might indicate spam. The literature discusses various machine learning techniques such as decision trees, support vector machines, and neural networks. These methods are evaluated on their ability to adapt and respond to new spamming techniques, highlighting the need for ongoing research and adaptation in spam detection technologies.

## **2.2 Digital Forensics (DF)**

Digital forensics concerning emails involves methods and technologies used to investigate and understand the origins and consequences of spam. This analysis is important for identifying threats, recovering data, and understanding attackers' methods. It is also essential for gathering evidence for legal actions against those responsible for spam.

The literature provides a broad overview of forensic techniques, from basic code analysis to complex pattern recognition using advanced machine learning algorithms. These tools dissect every part of an email to gather insights about its origin and purpose. The analysis of email headers and bodies can reveal much about the sender's identity and intentions, which is important for blocking spam emails and improving security measures.

Additionally, digital forensics also plays a vital role in legal scenarios where the integrity of email evidence must be maintained. Techniques for ensuring the authenticity of email data are discussed extensively in the literature, highlighting their importance in legal cases involving email fraud and other cybercrimes.

### **2.2.1 DF Subdomain**

Subdomains within email digital forensics focus (Al-Dhaqm et al. 2021; Devendran, Shahriar, and Clincy 2015)on specialized areas like phishing detection, malware analysis, and examining email headers and bodies. Each area uses specific techniques to target different threats associated with email spam .

For instance, phishing detection involves identifying emails that trick recipients into giving away sensitive information. The literature reviews text analysis techniques and machine learning models that distinguish between legitimate and malicious emails based on their content and structure. Malware analysis focuses on inspecting attachments and links within emails to detect harmful software.

The analysis of email headers and bodies is also important, as they contain valuable data for forensic investigation. Advanced techniques supported by machine learning help detect anomalies that may suggest malicious intent or spam. These specialized forensic areas are essential for a comprehensive approach to managing email threats (Al-Dhaqm, 2021).

### **2.2.2 Digital Investigation**

Digital investigation of emails involves systematic processes for examining emails to detect spam and other security threats. This includes everything from collecting data to analyzing it to gather actionable insights. Such investigations are important for understanding the scope of threats conveyed through emails and for collecting evidence in cybercrime cases.

The literature describes a range of manual and automated methods for analyzing emails. Machine learning algorithms are highlighted for their ability to handle new and evolving spam tactics, analyzing incoming emails in real-time to block spam more effectively.

Moreover, these investigations also deal with recovering deleted or damaged emails, which can be important for forensic and legal purposes. Advanced recovery techniques enable the retrieval of emails that have been deleted intentionally to hide illegal activities, showcasing the importance of these methods in comprehensive digital investigations.

## **2.3 Email System**

An email system's architecture is fundamental to understanding how spam attacks are conducted and mitigated. An email system includes servers, clients, and network infrastructure that supports message transmission. The protocols that form the structure of email systems, such as SMTP, IMAP, and POP, are essential for sending and receiving emails securely.

Each component of the email system can be exploited by spammers, who may take advantage of system vulnerabilities to send spam or malicious emails. Security measures and protocols like TLS and S/MIME are important for encrypting email data to prevent unauthorized access and ensure data integrity.

Machine learning algorithms significantly enhance the system's ability to detect and prevent spam by analyzing patterns in incoming emails and learning from past attacks. The literature includes case studies demonstrating the successful integration of machine learning into email systems, providing evidence of their effectiveness.

### **2.3.1 Email Structure**

Understanding an email's structure is vital for effective spam detection and forensic analysis. An email comprises different parts, including the header and body, which may contain various information types that spammers can exploit. The literature reviews techniques for analyzing these components to detect spam and other threats.

Email headers contain metadata such as sender and recipient information, which can be important for tracing the origin of spam emails. Techniques for extracting and analyzing this data are detailed in the literature, emphasizing their importance in identifying fraudulent activities.

The body of an email, where the message's content resides, is also analyzed for signs of phishing, malware, or scams. Machine learning algorithms play a significant role here, using natural language processing to understand text and identify harmful content effectively.

### **2.3.2 Email Analysis**

Analyzing emails is important for detecting spam and other malicious content. This process involves a range of techniques that look at the details within an email's content and structure to identify threats. The literature discusses various manual and automated methods that are used to analyze emails. Among these, machine learning algorithms stand out for their ability to adapt and improve their detection capabilities over time.

These algorithms can analyze emails based on learned patterns from vast amounts of data, making them highly effective at identifying both known and new types of spam. Techniques such as statistical analysis, pattern recognition, and machine learning are used to examine the content within emails, including the words used and the way messages are structured. The results from these analyses help in distinguishing between normal and potentially harmful emails.

The challenges of accurately detecting spam without misidentifying legitimate emails are also a significant focus in the literature. It discusses the importance of achieving a balance where the system effectively blocks as much spam as possible without affecting regular email communication. The advancements in machine learning algorithms have helped improve this balance, making email analysis more reliable and efficient (Chhabra, 2015).

### **2.3.3 Crime Using Email**

Emails are often used for criminal activities, including scams, phishing attacks, and spreading malware. These types of crimes use deceptive emails to trick recipients into exposing private information or downloading harmful software. The literature reviews various crimes that are committed using email and discusses the methods used to prevent and detect these activities.

Phishing attacks, for instance, are a common topic in these discussions(Ghafarian, Mady, and Park 2020; Singh Chhabra, Singh Chhabra Asst Professor, and Singh Bajwa Asst Professor n.d.). They involve emails that mimic legitimate requests from reputable sources to steal user data. The literature explains how text analysis and behavior monitoring can detect these fraudulent emails by spotting signs of deception that are commonly found in phishing attempts(Ghafarian, Mady, and Park 2020).

Legal aspects of email-based crimes are also discussed, emphasizing how important it is to handle email investigations properly to ensure that the evidence collected is valid in a court of law. The literature talks about the methods used to trace the origin of criminal emails and the techniques that help prove the intent and actions of the perpetrators.

## **2.4 Email Forensics**

Email forensics is a critical field that deals with investigating and analyzing emails to find evidence of cybercrimes. This section of the literature provides insights into the forensic processes that help understand the nature of email-based threats and find the culprits behind these activities. It involves examining every part of an email, such as the headers, body, and attachments, to gather useful forensic data.

Techniques used in email forensics include analyzing the email headers for origin and routing information, which can help trace the source of spam or malicious emails. The body of the email is also scrutinized for any signs of phishing, fraud, or malware. Machine learning techniques are increasingly used in this field to automatically analyze large volumes of emails quickly and accurately.

The literature also covers the challenges faced in email forensics, such as the need to continuously update and improve forensic tools to keep up with the evolving tactics of cybercriminals. It emphasizes the importance of developing more sophisticated and adaptive tools that can handle the complexity and volume of modern email traffic (Devendran, 2015).

### **2.4.1 Header Forensics**

The forensic analysis of email headers is an important practice in the investigation of email-related crimes. Email headers contain a wealth of data, including the sender's details, the route the email took, and the time it was sent. The literature discusses how forensic experts analyze this information to identify and trace the sources of spam and other malicious emails.

Various tools and techniques are used to parse and examine the headers to detect any anomalies or signs of tampering that might indicate fraudulent activities. Machine learning algorithms are also applied to help automate the detection process, making it faster and more accurate. The ability of these algorithms to learn from large datasets of email traffic allows them to identify unusual patterns that may be missed by manual analysis.

### **2.4.2 Body Forensics**

The body of an email is where the main content lies, and it is a key focus area in email forensics. This section of the literature explores the methods used to analyze the text, links, and other content within email bodies to detect signs of phishing, scams, and malware. Techniques range from simple keyword scanning to complex algorithms that analyze the semantics and intent behind the text.

Machine learning models are particularly useful in this aspect of forensics. They can analyze the content for typical phishing indicators or patterns known to be used in scams. These models improve over time as they learn from more examples, making them an essential tool in the ongoing fight against email-based crimes.

### **2.4.3 Signature Forensics**

Signature forensics focuses on verifying the authenticity of emails through digital signatures. The literature explains how these signatures help confirm that an email has not been altered and is genuinely from the purported sender. This process involves cryptographic techniques that secure the integrity of email communications.

The discussions include the technical details of how digital signatures work and the challenges involved in implementing and managing these security measures effectively. As cyber threats evolve, so do the techniques used in signature forensics, with ongoing research aimed at enhancing the reliability and security of digital signatures in email communications.

## **2.5 Challenges in Email Forensics**

Despite advancements, email forensics faces many challenges, which are extensively explored in the literature. Handling the massive volume of email data and keeping up with sophisticated spamming techniques are major issues. The discussions cover the need for scalable, efficient tools that can process large amounts of data without sacrificing accuracy.

Privacy concerns are also a major topic, with discussions on how to balance effective spam detection with the protection of users' privacy. The implications of laws like GDPR on forensic methods are considered, highlighting the need for compliant, ethical approaches in forensic practices (Ghafarian, 2020).

## **2.6 Summary**

This chapter thoroughly reviews the current methods and technologies employed in detecting and analyzing email spam through machine learning algorithms. The focus has been primarily on how these methods can be applied to enhance email security by identifying and filtering spam emails more efficiently.

The literature underscores the use of various machine learning algorithms, including Naïve Bayes, Support Vector Machines (SVM), and neural networks, which have been important in recognizing patterns and anomalies indicative of spam. These algorithms are appreciated for their ability to learn from data and improve their predictions over time, which is crucial given the ever-evolving nature of spam attacks.

Moreover, the chapter discusses the challenges encountered in email spam detection, such as the dynamic nature of spam tactics and the need for algorithms to adapt quickly to new threats without generating a high number of false positives. The integration of machine learning not only helps in adapting to these challenges but also aids in automating the detection process, thus increasing the efficiency and accuracy of spam filters.

In addition to technical discussions, the chapter also touches on the potential improvements in spam detection systems, such as enhancing data sets for training models and refining algorithms to reduce errors further. These advancements are critical as they contribute to developing more robust systems capable of handling the complexities of modern cyber threats.

# CHAPTER 3: Methodology

## **3.1 Overview**

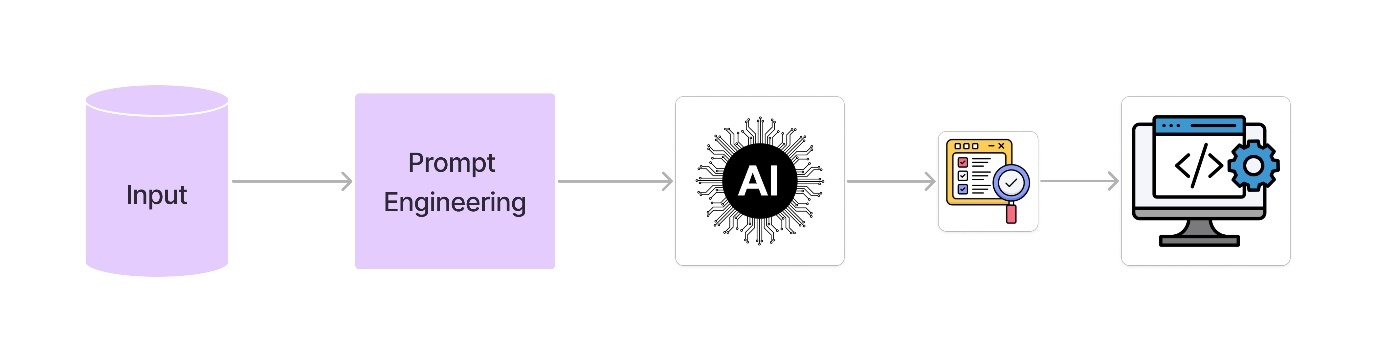
This is the technical process and technology stack that went into building the Spam Detector platform, bringing together the main building blocks: Gemini LLM, prompt engineering, and IP-based geolocation. It has diagrams to explain how the system works, how users work and how data are handled.

Spam Detector leverages Flask to develop an easy-to-use web-based app for identifying spam emails. Thanks to Google’s Gemini LLM and LangChain, the software scans the email address, body, and footer for spam attributes. And also IPinfo.io API makes context by telling the users where they are located. The authentication is JWT to allow user and the data is stored in SQLite. This NLP-geolocation architecture generates high-quality spam detection.

## **3.2 Operational Framework**

Spam Detector platform, natural language and geolocation processing integrated into one system of operations. It starts with the user authentication (JWT) to get an encrypted access. When logged in, they paste in the email, body and footer for analysis. Gemini LLM scores them and warns against spam, using language, tone, and formatting. IPinfo.io API: The location of the user gets returned which adds a contextual dimension to the analysis.

Then the system outputs the spam detection as well as the location to an intuitive interface. The system is built to scale, meaning future system upgrades (like a new model or more geolocation capabilities) can be easily implemented.



## **3.3 Gemini Large Language Model**

The core analysis engine of Spam Detector is Gemini LLM (in this case "gemini-pro") – which is also the name of the version that has been released. It is programmed for deep natural language processing and checks for spam signals in email parts (address, body, footer). If we keep the temperature at 0.5, then it is more or less a combination of creativity and accuracy. This is great for scanning for spam, such as urgency, capitalization, and word play.

LangChain plugs Gemini LLM directly into the tool, so that the model is guided through the analysis of each email block with structured prompts. The model output is an explanation of whether or not the email is spam or not, which translates into more transparency and trust. Gemini Performance: This ensures real-time processing and thus a fast user experience.

## **3.4 Prompt Engineering**

It is prompt engineering that helps Gemini LLM guide email component analysis. The callouts are designed to target the biggest spam triggers like offensive words, style of greeting and footer copy. A template, for instance, asks the model to crawl the email address, body, and footer and look for words or formatting conventions that imply spam.

The request also invites a full-blown spam risk analysis, unifying the model output into a concise score. The prompt was revised during development to make it more explicit and precise. Using a steady temperature of 0.5 guarantees stability in the model responses and not to have an excess of creativity that can corrupt the analysis.

## **3.5 Geolocation with IPinfo.io API**

IPinfo.io API integration — This API gathers geolocation information by analyzing IP address. This contextualizes the spam analysis and allows users to calculate risk depending on the location. IPinfo.io API returns city, region, country etc as a parameter that gets included in the result.

IPinfo.io – Geolocation providers who offer quality data and are safe to use. The app makes use of IPinfo’s free plan for simple geolocation requests, so it is relatively cheap to integrate. If the platform grows, IPinfo’s higher-level packages come with more advanced options, like ASN data and privacy detection, which help extend the system. The geolocation service contextualizes the origin of the threats during the spam analysis.

## **3.6 Functional and Non-Functional Requirements**

It categorizes the needs into functional and non-functional, for what the system will do and a good user experience.

### **3.6.1 Functional Requirements:**

Functional requirements are what you need the spam detection platform to provide in order to function properly and be useful for users. These requirements are for the most basic functions the system must run such as:

* **Authorization and Registration for Users**: You have to enable user registration, login, and JWT token authentication for users. Users need to be able to register, log in, and have protected routes to manage their spam filters.
* **Email Filtering**: The main purpose of the platform is to receive email submissions (address, body, footer), analyze it based on Gemini LLM model, and provide a spam label (Spam or Not Spam) and analysis.
* **Geolocation API integration**: The system needs to be integrated with the IPinfo.io API to fetch user’s geographical location by IP address and present it along with the spam analysis result.
* **Responsive User Interface**: Web app must have a simple user-friendly interface with step-by-step instructions for entering email addresses, seeing results, and accessing other pages such as About and Contact.
* **Rendering of Content**: The system must convert the model’s markdown output to HTML so the end user can see it in a structured way on the frontend.
* **Handling of Errors**: The app should be able to gracefully handle an error and give proper notifications when inputs are invalid or external APIs don’t work.

### **3.6.2 Non-Functional Requirements:**

Non-functional requirements are about quality, performance, and usability of the system. These include:

* **Performance**: Gemini LLM model should be analyzed and returned from the platform in a reasonable time, making it easy for the user to use with minimal analysis and geolocation wait times.
* **Scalability**: System must be scalable to support the growing user load and scale to more API requests as the number of users increases (especially for spam analysis and IP geolocation).
* **Reliability**: You want an application that is uptime & minimal interruptions. It should be fault tolerant, especially with third party API calls (e.g., Gemini LLM, IPinfo).
* **Safety**: Platform should be secured while handling users’ data especially during authentication and when engaging with external APIs. Security related information such as passwords needs to be encrypted, and JWT tokens need to be safe stored and sent.
* **Usability**: User interface must be nudgeable, and users must get clear directions and notifications. It should be simple for technical and non-technical users to use, and spam detection should be a piece of cake.
* **Maintainability**: You should create the system so it can be updated, patched, and upgraded. Modularity and documentation should be maintained in case of future fixes.
* **Data Security**: The platform should be compliant with the privacy and data security policies and procedures, user data must be protected and protected as per the law.

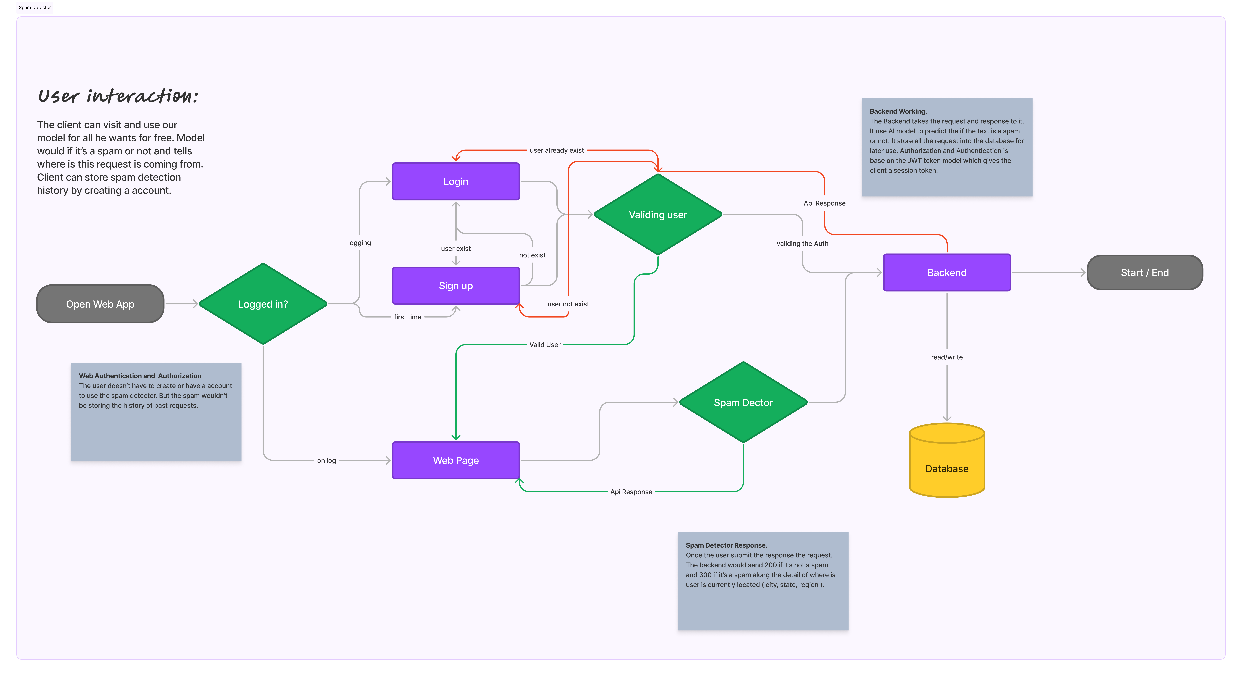
These functional and non-functional features are essential to making sure the spam detection system is good at its job and will work as expected.

## **3.7 Diagrams and Framework Components**

The following flowcharts show you how the core operations and data are moved through the system:

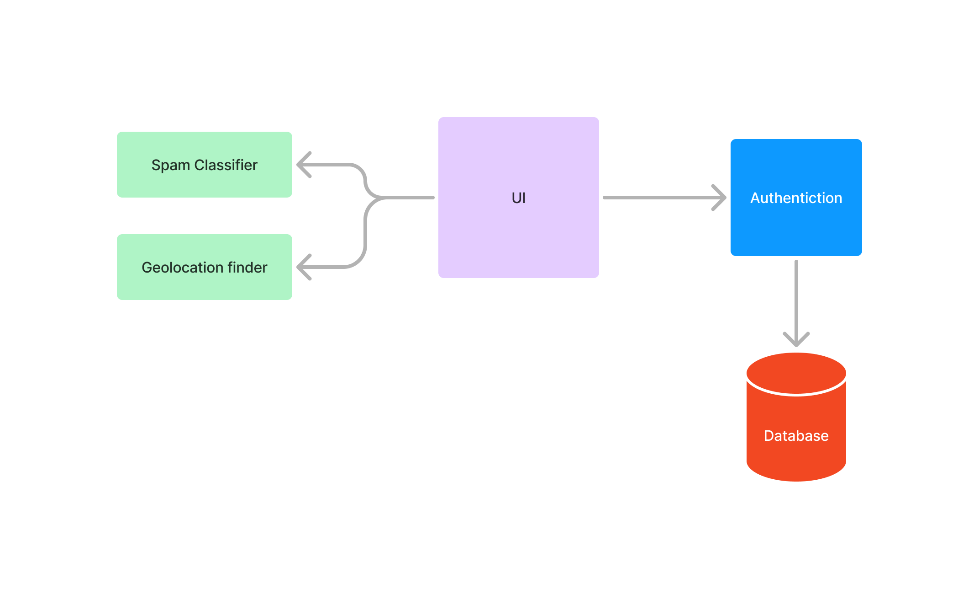
### **3.7.1 Activity Diagram**

Activity diagram of the spam detection system The flow of interactions between the user, the platform, the components of the system and the database. User — the user being the main player, either has an existing account or creates a new one. Once logged in, they can open the web application and paste in a message or website URL. This input is passed to the spam detector, and it goes through the data and tests in several ways whether or not it is spam. The spam detector, after digesting the input, passes the result to the validator. The validator in turn checks the output and the authenticity of the source of the request before returning the final result (spam or no spam) back to the web app. When the verdict comes, the web app shows the result to the user. When the user is logged in, the spam detection history is saved to the database for later. The activity diagram here shows clearly how the system implements secure and effective spam detection through integration between users, processes and storage.



### **3.7.2 Block Diagram**

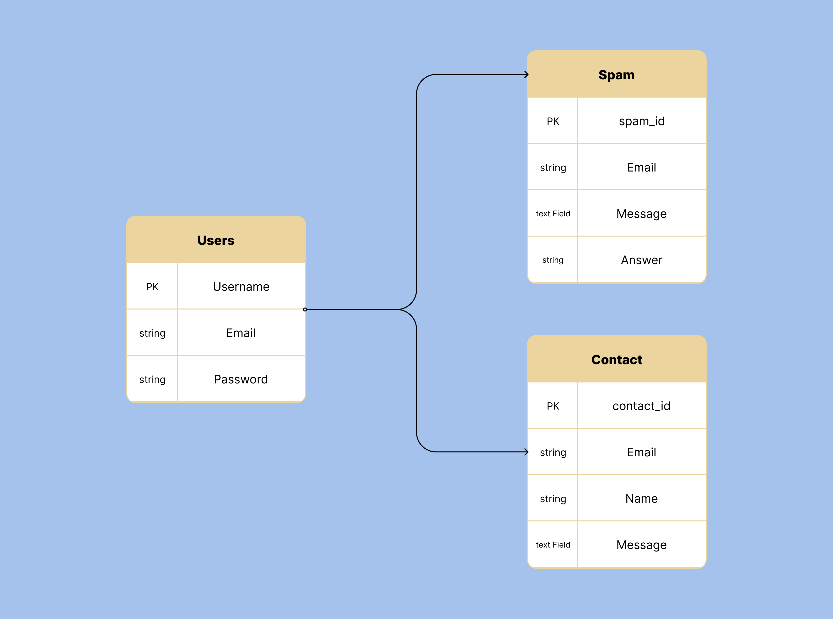
The block diagram of the spam scanner shows you the various components of the platform and how they interact to make the platform run smoothly. The UI is the front-end, and you will fill in text messages or URLs for analysis. Authentication module: It validates who has access to the system with usernames, passwords or two-factor authentication. Authenticated data then sent to Spam Classifier that uses machine learning, natural language processing or statistical analysis to determine if it’s spam or not. We can also use the Geolocation Finder to determine the user’s geographic location or origin for the content, which gives a nice bit of background for classification if spam is originated in specific places. Classification’s output and geolocation data if any are also returned to the user via UI. For added convenience and later reference, data such as user data, input data, classification results are maintained in a Database. This block diagram depicts visually the way all of the components of the system interact to deliver spam prevention and secure user experience.



### **3.7.3 Entity Diagram**

Entity-Relationship (ER) model of the spam detector model describes entities in the system and relations among entities. One of them is Users, which holds the data about the users of the platform like their username, email, and password. Spam has records about the spam messages reported by users, such as the spam ID, the email address where the message came from, the message content, and the response/vote the system derived about the message. The Contact object keeps the contacts data for the users like their ID, email address, name, and any messages.

These interactions are of the essence to making sense of data movement. The Users to Spam is a one-to-many connection and you can report several spam messages to it. Users to Contact is also one-to-many meaning a user will be able to handle more than one contact. We have also a Contact to Spam many-to-many relationship (where the same contact might receive multiple spam messages and one spam message could be belonging to multiple contacts). This ER diagram visualizes how the system logs users, their spam, and contacts in the bigger spam detection process.

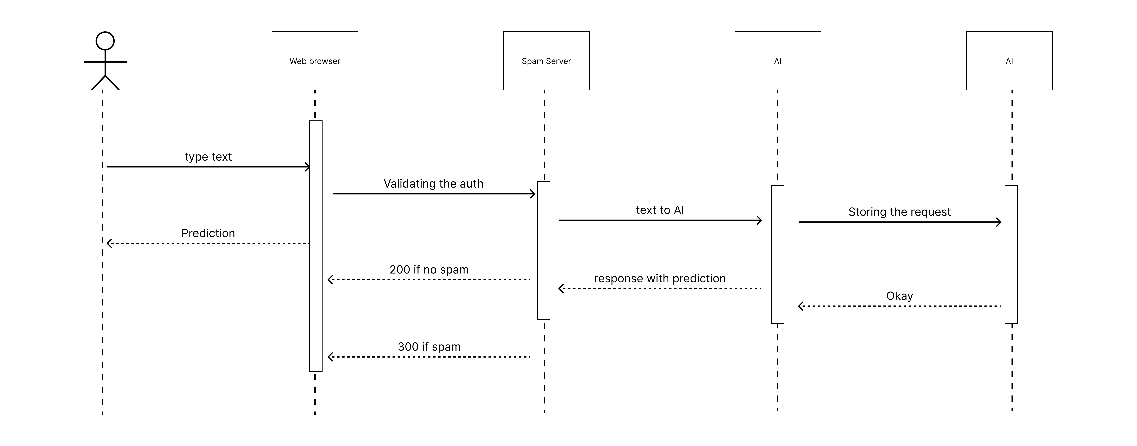


### **3.7.4 Sequence Diagram**

Spam detection system sequence diagram: the interaction pattern of the various modules when the user typed text to be detected. This is started by the User, typing something in the Web Browser, who is the User’s interface. Once a user enters the text, the Web Browser submits it to the Spam Server with any authentication information.

The Spam Server then passes the text to the AI part, which reads the content. The AI takes in the text, applies its already trained algorithms and models and gives a prediction, "No spam" or "Spam". The AI relays the prediction to the Spam Server and the Spam Server relays the result to the Web Browser. The return code includes a status: "200 OK" if no spam is found or "300 Spam" if text was found to be spam.

And finally, the Web Browser shows the result to the User, it displays the text as spam or not, with relevant system response to the User. It is this chain of events that makes sure that the user is able to receive spam detection results on time and correctly.

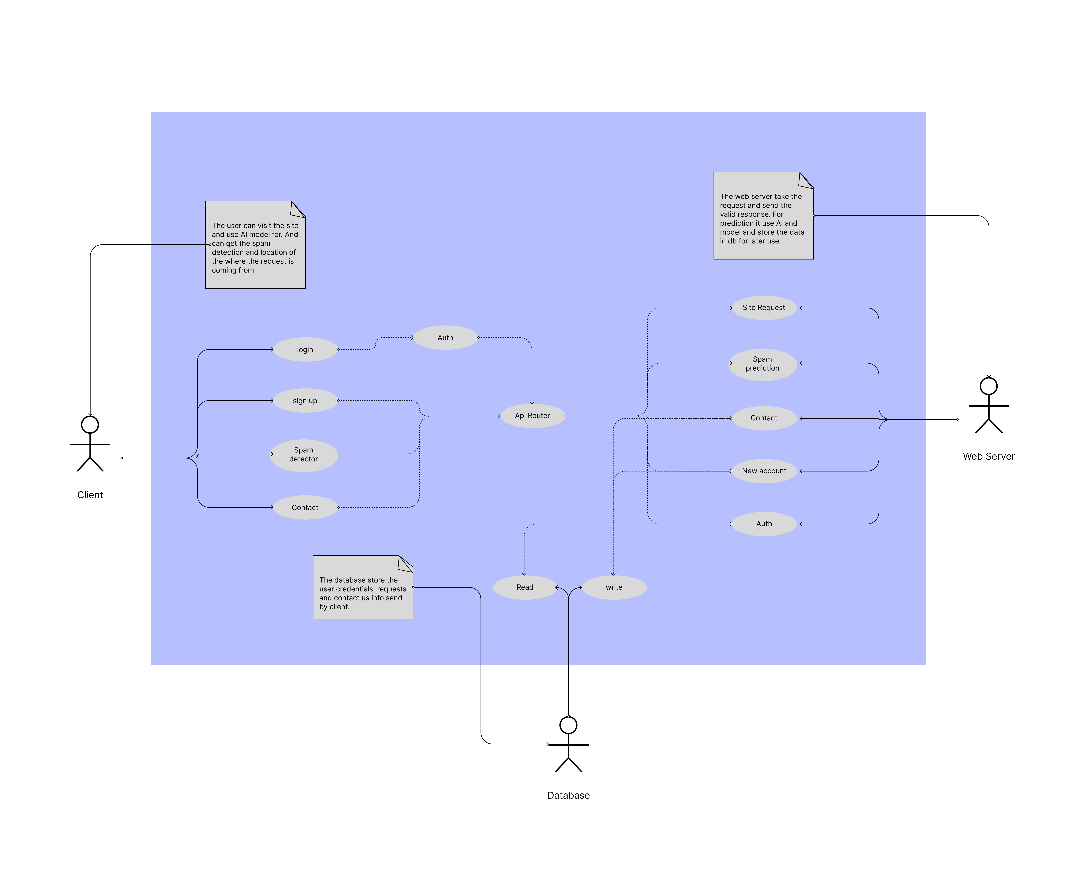


### **3.7.5 Use Case Diagram**

A spam detection system use case diagram shows how the system and its users interact with each other, in terms of what actors interact with the system. The Client which represents the end-users, connects to the Web Server which is the fundamental part that accepts the requests from the user and handles services. Database : This is the place where all the data needed by the system are stored: User name, Spam Detection result, Logs etc.

The system use cases are Login, the client login to access the account, Sign Up, new clients login to the system and Spam Detection, the client can fill text or other content for the spam detection. Users can also View Account for accessing profile or setting, Contact Us to interact with the system team.

The system can also issue Push Notifications for clients to get updates or critical notifications. The Database is the core of this ecosystem that keeps the login credentials, request data, and so on which the Web Server can read to offer you seamless services. Client-web server communication, web server-database communication all support the functionality of the spam detection system.

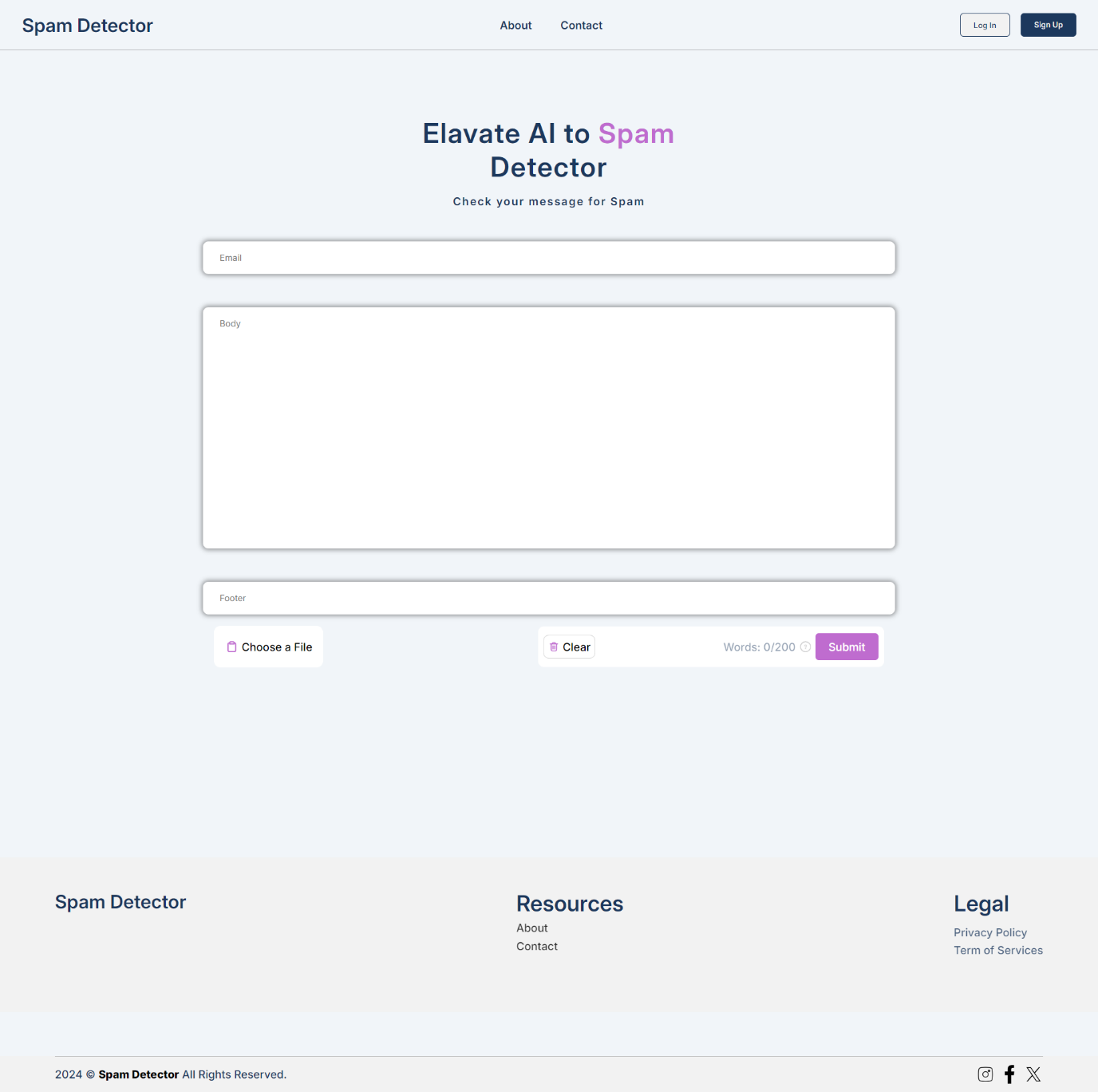


## **3.8 Web Application**

The web app for Spam Detector platform is a neatly designed app to give users an intuitive and practical way to review emails and flag spam messages. Backend developed in Flask and frontend created using HTML, CSS, and JavaScript with the use of a responsive frontend. It reads the user inputs, pings the Gemini LLM model for spam detection, and makes IP geolocation usages available from the IPinfo.io API to contextualize the results. Here is a brief of the app’s main pages.

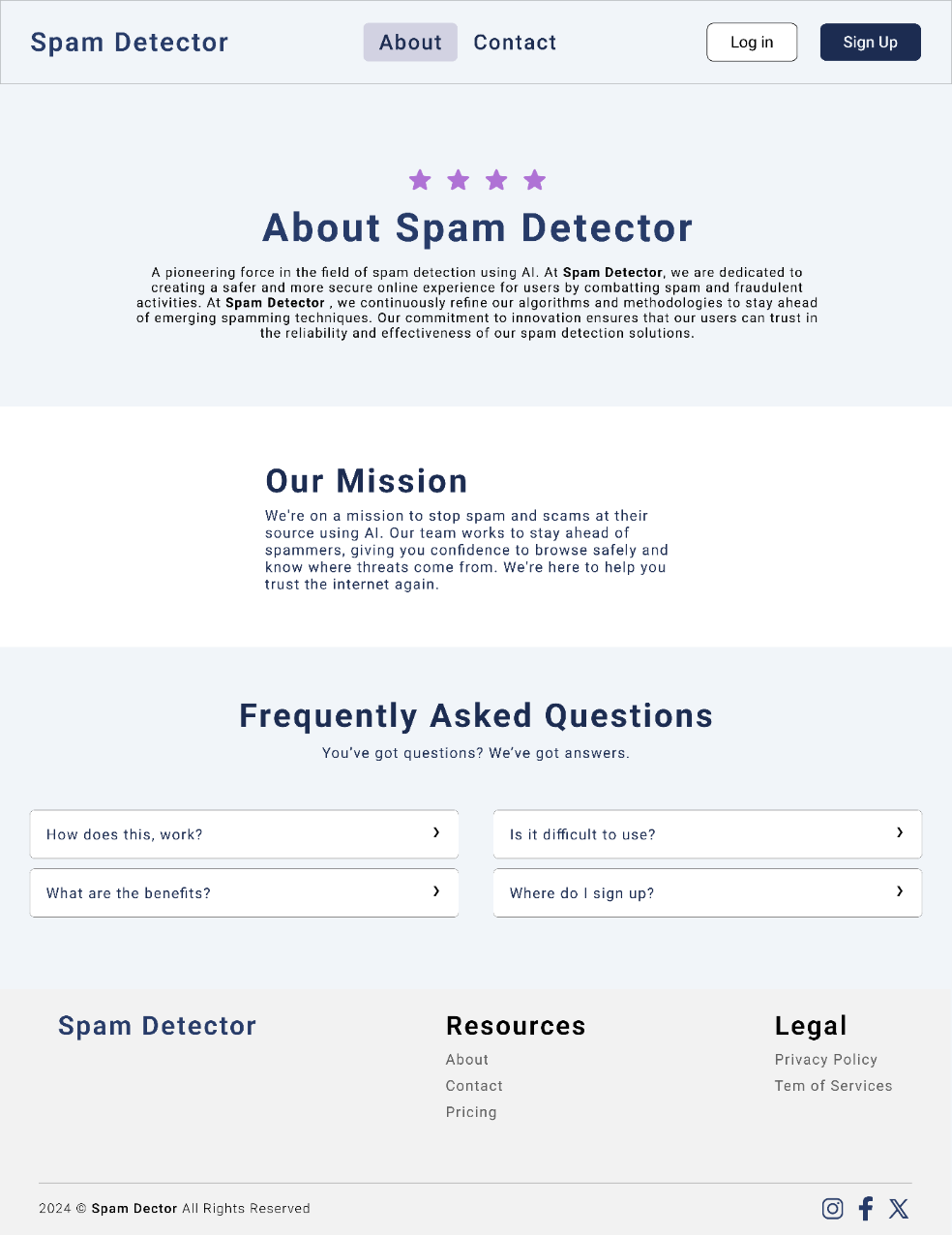
### **3.8.1 Landing Page**

Landing Page: This is the main screen where people are able to grasp the fundamental function of the platform and get started with the spam detector. It has a clean and simple look with the core function of the platform — email spam detection. The users are asked to type three parts of an email, email address, body, and footer. The platform reads these inputs, applies Gemini LLM model and gives you a spam classification along with a descriptive report. This page is the hub of all user actions and the analysis output is located directly below the form to be able to see right away what outcome a user sent. It also has input fields for email, body and footer, a submit button to initiate the spam analysis and a display box to present the spam score (Spam or Not Spam) and model-based comprehensive analysis. The IP geolocation information is also included to get more context about where the user is.



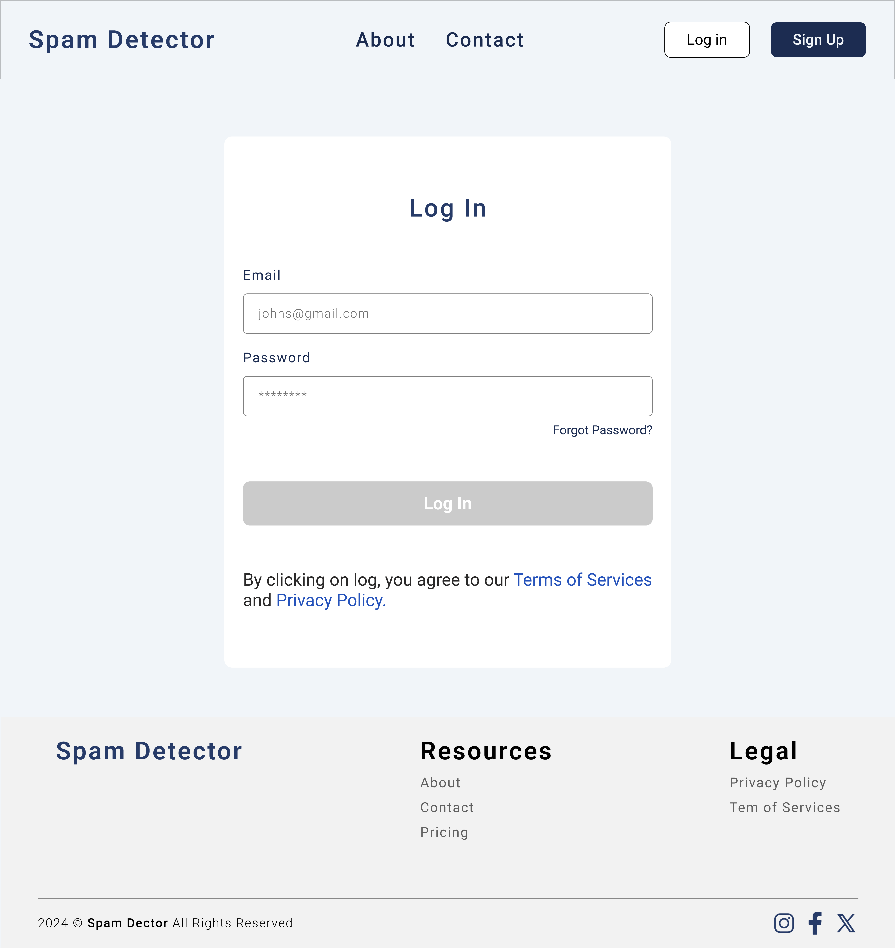
### **3.8.2 About Page**

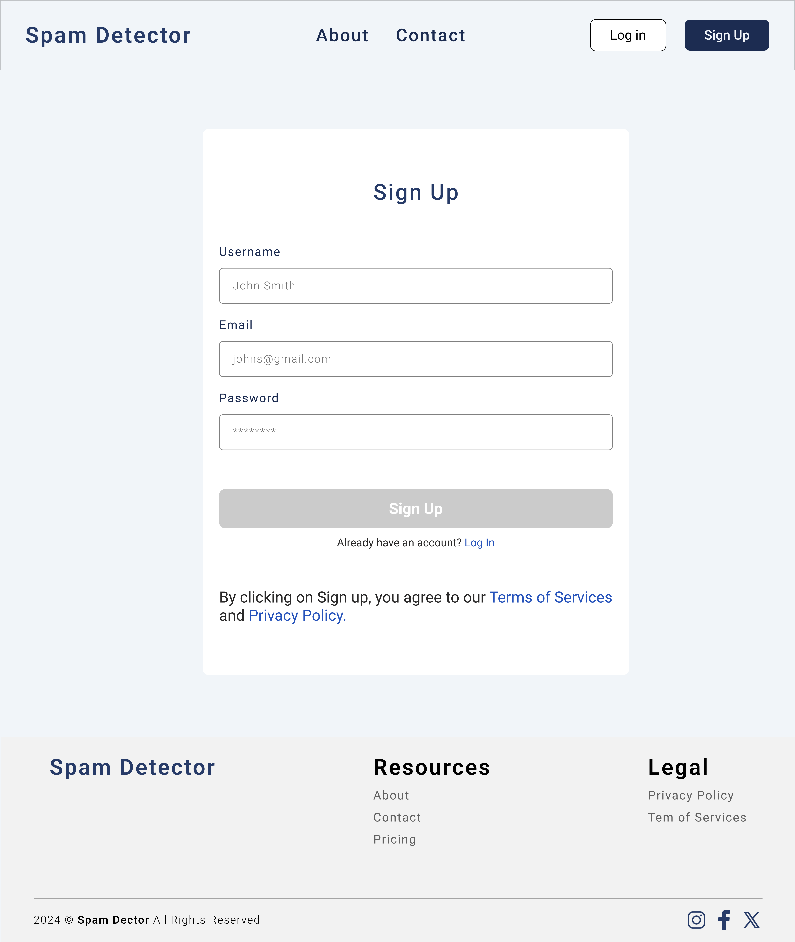
About Page provides you with the basics on what the Spam Detector platform is and why it’s here. This page will inform visitors on the platform’s purpose, technology and team that’s helping with spam detection. There are a couple of chapters in it like "About Spam Detector" it introduces the platform as a leading AI-powered spam detection and tries to eliminate spam and fraudulent content for safer web space. In another part, the "Our Mission," you will see that platform is about stopping spam and scams at their source with AI. This website tries to empower people and get them to trust the internet again by keeping spammers at bay. There is also a "Frequently Asked Questions (FAQs)" page with questions you might have about the platform, its features and benefits, user-friendliness, and registration.



### **3.8.3 Sign-Up / Sign-In Page**

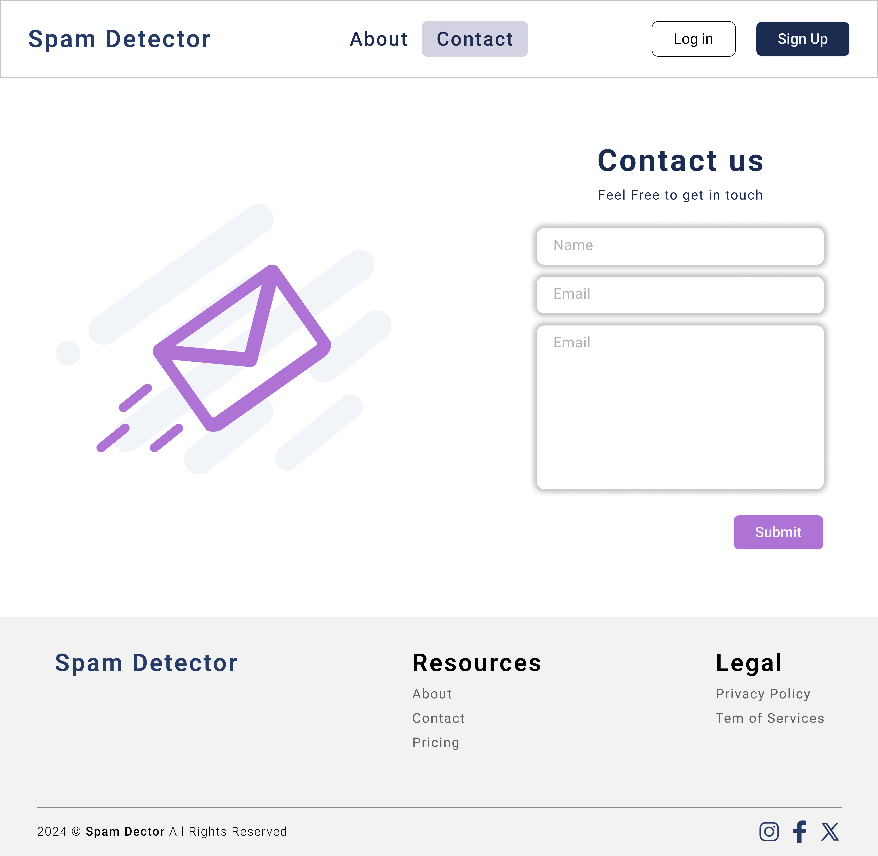
Sign-Up / Sign-In Page You can sign up or sign in to your customized spam detection services. You can sign up with a username, email and password for new users. It encrypts the passwords and saves the information of users in a database for future reference. Users who are already registered with us can log in with their email and password. After successful authentication, the user gets a JWT token which opens the route protection in the platform.





### **3.8.4 Contact Page**

Contact Page — This is a page where the user can communicate with support team to ask questions, feedback, or questions about the platform. Here there is a contact form with name, email address and message content. The form is meant for easy communication and user feedback collection.



The web application is built from the functional and UI point of view. It gives the tools you need to enter email information, get spam data, and other contextual information, like geolocation. The web-browser is easy, which allows the platform to be navigated and engaged by technical as well as non-technical users. With pages such as Landing, About, Sign-Up/Sign-In and Contact, the system provides a single integrated user experience that combines spam-detection and usability. All pages have a mission and direct the users through spam detection and platform activity.

# CHAPTER 4: Implementation and results

## **4.1 Implementation**

The implementation of Spam Detection platform started with designing the base architecture, which combines the NLP (natural language processing) of Gemini Pro and LangChain with the IPinfo.io geolocation data. Backend developed with Flask a lightweight python library for writing simple web apps. The app was made simple and intuitive with authentication using JWT tokens for security. The interface was responsive and user-friendly, you could enter the email address, body, footer of your email and have real-time spam reports.

During development, we worked on the integration of these tools and technologies to form a working system that could rank spam in emails based on patterns in language, greeting, footer, and geolocalized information. Each email message initiates multiple analyses: Gemini LLM analyzes the message, IPinfo.io API gets the location of the user to contextualize the data. The output is then presented on a simple, nifty screen, with a classification of whether the email is spam and in-depth response.

## **4.2 Tools Used**

### **4.2.1 Gemini LLM**

Central to the Spam Detection system is Gemini Pro, Google DeepMind’s powerful large language model (LLM) to parse and comprehend textual information. Gemini Pro works best when applied to spam detection since it’s capable of parsing and deciphering complex linguistic patterns, finding out espionage signals, and providing in-depth insights into text contents like email addresses, bodies, and footers.

Gemini Pro is multimodal in that it can process text, but also image, audio and other data types, so it is good for things other than spam detection. But in this use, it takes advantage of its extensive knowledge of NLP to pick up a red flag: urgency, vague words, illegible formatting, and other usual indicators of spam or phishing emails.

Google's Tensor Processing Units (TPUs) run the model and support the computation power for real-time calculations. With Gemini Pro’s fine-tuning based on human feedback, it is more accurate and reliable than ever, being able to detect patterns even in emails that attempt to disguise malicious intent.

### **4.2.2 LangChain Integration**

LangChain is the linker between the Gemini Pro model and the rest of the application. It organises the cues for Gemini so the analysis of each email body, address, and footer is comprehensive and context-aware. We can use LangChain library to create organized calls to the model, which will process the text as per the spam detection goals, and output is easily understandable.

### **4.2.3 IPinfo.io API for Geolocation**

IPinfo.io API — Third-party API allows the platform to fetch geolocation based on IP address. The service offers context data about the user’s geographic location which can be added to the spam analysis. We incorporate geolocation into the data to bring another dimension of intelligence as it will be useful for identifying regions specific spam campaigns or abnormal activity from certain geographical regions.

### **4.2.4 Flask Framework**

The application backend is in Flask framework which allows us to create a lightweight and scalable web server. We route and respond to requests, and integrate services (Gemini LLM, IPinfo.io API) in Flask. It’s ideal for low overhead, fast development projects. Flask was also super easy to work with, which was ideal for this project since the aim was to provide an easy to use web interface for implementing large services.

### **4.2.5 Frontend Technologies (HTML, CSS, JavaScript)**

The app frontend is HTML, CSS, and JavaScript coded so that the platform is responsive and easy to use. The content is structured with HTML and the styling and layout done with CSS so the platform looks good. JavaScript for dynamic functions (form filling and live spam detection results). These technologies make it seamless for the user.

### **4.2.6 SQLite for Data Storage**

SQLite: Database for holding user credentials, email requests, and spam reports. This thin relational database is built into the app to allow you to persist your spam check history and get past results. : SQLite is a very easy and easy to integrate server that was picked for this project.

## **4.3 Gemini Pro: LLM Model for Text Analysis**

Gemini Pro is the AI engine which is most relevant to the spam detection work. It can use its advanced machine-learning abilities in natural language processing to do granular analyses of email messages. The model is retraining from huge data volumes and therefore, can detect subtle patterns in the language, anomalies and risk factors of emails. Spam Symptoms: From the email address, body, and footer, Gemini Pro can identify urgency, fraudulent offers, capitalization, and other spam behavior.

Because of its multimodal architecture, Gemini Pro isn’t restricted to text processing but it is best used for textual analysis, and hence, is ideal for a project such as spam detection. It is able to learn about both simple and complex grammatical forms of speech, which means that it’s able to flag emails that could otherwise go undetected by simpler models.

The live processing of Gemini Pro will notify users right away whether or not their email data is spam. And also, the reinforcement learning used for training makes sure that it gets better over time and is accurate and consistent in its tests.



## **4.4 Testing**

This testing process took a series of steps to make sure that Spam Detection platform worked. To test the backend, we first exposed API endpoints with tools such as Postman. That included trying GET and POST requests to see if the server could handle inbound data, like email inputs to be analysed. The intent was to see if the server could be configured to route requests, handle them with the Gemini Pro model and return correct results.

On these tests we also looked to test that the platform handled edge cases, like malformed input or an API call gracefully. We made sure that it was able to handle errors correctly by posting friendly error messages when they happen. We also made sure the JWT authentication was secure and users’ data was not being compromised during the test phase.

That’s also the heart of the testing, where you send a bunch of email samples – legitimate and bogus – and then let the system sort them out. The system went through every submission thoroughly and returned an evaluation of the email language, greeting, and footer. The model spotted any anomalous activity and returned a total "Spam" vs. "Not Spam" rating. The address, body and footer all got scored individually for odd or deceptive language, helping dissect the logic behind the final category.

These tests showed that the platform consistently identified suspicious emails and gave users specific actionable alerts from its text and context analyses.

# CHAPTER 5: Future Work and Recommendations

## **5.1 Future Work**

Currently existing Spam Detection platform has already mastered the use of latest technologies like Gemini Pro and LangChain to scan emails but there is always something more to improve on. There are a number of possible ways of building on top of the system:

**Expansion to Multi-Language Support**

As of now, it only offers anglophone email. But the future work might be to extend this to multi-languages. With the power of Gemini Pro’s multilingual features, the system would be able to analyse emails in multiple languages, and thus could be applicable to a larger user base. It would mean that the LLM needs to be trained again on various data sets to be able to detect spam correctly on all linguistic variations.

**Improved Spam Detection Algorithms**

The model is currently spam detection on the basis of words, but there are some constraints. Future upgrades might be in the form of incorporating more advanced machine learning algorithms (eg, deep learning or hybrids using supervised and unsupervised learning). This would allow the system to catch even more advanced spammers (including artificially created content) and decrease false positives.

**Enhanced Geolocation Analysis**

Despite the IPinfo.io integration offering useful location information, more geolocation-specific data could have been added, such as the IP addresses’ reputation, or if the email is from a region where spam or phishing attacks are common. This would give much better detail to spam detection and improve the process.

**Real-Time Spam Reporting and Learning**

An improvement would be to be able to feed in user feedback to the system. : Users could be able to set spam or not on emails, this data being fed back into the model for continuous learning. That would let the system adjust in real-time to new spam strategies so that the detection algorithm could respond to emerging threats more dynamically.

**Integration with Email Services**

If you’re going to use the spam detection system on a larger scale, you can integrate it with email systems (e.g., Gmail, Outlook) and have it automatically check emails for new ones in real time. The system could be used as a plugin or extension in these sites, to warn users about spam messages without them having to manually submit them. That would streamline the process and offer users a true real-time, seamless spam filtering.

**Mobile App Development**

Since smartphones have been the main way people access emails, there may be an application version of the spam filter that people can scan their emails from their smartphones. The app might be the same functionality as the web application but with push notifications, real-time alerts and native integration with email apps.

## **5.2 Recommendations**

Now that we have an implementation in place and some testing has gone ahead, here are some suggestions to improve the spam filter:

**Continuous Model Training and Updates**

Spam techniques change all the time so you have to train the model repeatedly with new datasets. Updates of the model over time can enable the model to detect new kinds of spam and phishing attacks so the system doesn’t fall short in the long run. We might even be able to implement active learning techniques where model learns from new examples generated by users.

**User-Centric Interface Enhancements**

The UI itself is good enough but there is room for improvement. : for instance, more detailed comments about flagged emails (eg, the word or phrase that caused the spam flag) can be offered to make users know why it was categorized as spam. Also, using visualizations like heatmaps or confidence scores could help make the analysis more clearer for users to understand.

**Robustness and Scalability**

If the users number is to expand, the platform’s robustness and scalability will be key. The backend must be optimized to handle a higher and higher number of requests with low latency. There are tools such as load balancing, caching, and horizontal scaling which can keep your performance high as the traffic goes up.

**Data Privacy and Compliance**

Since email information is very sensitive, the platform should comply with data privacy laws like the General Data Protection Regulation (GDPR) and California Consumer Privacy Act (CCPA). Providing security mechanisms such as anonymization of data, secure storage and clear processes of consent will establish trust and make the platform legitimate and trustworthy.

**Collaboration with Cybersecurity Experts**

For a better spam detection solution, partnering with cybersecurity experts and threat intelligence feeds can provide in real time information on the latest spam practices and trends. These experts can help the platform to keep up with new threats and adding threat intelligence from outside of the platform might improve detection rates even further.

**User Education and Awareness**

Its platform will filter spam emails but users must be taught to identify phishing and malicious emails as well. There should be a user education section that has tutorials, advice and best practices on how to identify a suspicious email to empower users to take an active role in protecting their digital identity.

# References

Al-Dhaqm, A. R.-K.-R. (2021). Digital forensics subdomains: the state of the art and future directions. *IEEE Access, 9*. doi:152476-152502

Broadhurst, R. &. (2020). Malware in spam email: Risks and trends in the Australian Spam Intelligence Database. *Trends and Issues in Crime and Criminal Justice,*, (603), 1-18.

Bujang, Y. R. (2013). Should we be concerned with spam emails? A look at its impacts and implications. *In 2013 5th International Conference on Information and Communication Technology for the Muslim World (ICT4M).* (pp. pp. 1-6). IEEE.

Chhabra, G. S. (2015). Review of e-mail system, security protocols and email forensics. *International Journal of Computer Science & Communication Networks,, 5(3),*, 201-211.

Crawford, M. K. ( 2015). Survey of review spam detection using machine learning techniques. *Journal of Big Data,* , 2, 1-24.

Devendran, V. K. (2015). A comparative study of email forensic tools. ,. *Journal of Information Security, 6(2),* , 111.

Ghafarian, A. .. (2020). *An empirical analysis of email forensics tools.* From Available at SSRN 3624617.

Graafland, J. H. (2018). *New technologies and 21st century children: Recent trends and outcomes.*

Mohammad, R. M. (2024). . A lifelong spam emails classification model. . *Applied Computing and Informatics,.*, 20(1/2), 35-54.

Patel, M. B. (2021). Latest Technology and Future Trends. . *In Applications of Artificial Neural Networks for Nonlinear Data. IGI Global.*, pp. 270-273.

Siddique, Z. B. ( 2021). Machine Learning‐Based Detection of Spam Emails. *Scientific Programming*, 2021(1), 6508784.

Yaseen, Q. (2021). Spam email detection using deep learning techniques. *Procedia Computer Science,, 184,*, 853-858.

Al-Dhaqm, Arafat, Richard Adeyemi Ikuesan, Victor R. Kebande, Shukor Abd Razak, George Grispos, Kim Kwang Raymond Choo, Bander Ali Saleh Al-Rimy, and Abdulrahman A. Alsewari. 2021. “Digital Forensics Subdomains: The State of the Art and Future Directions.” *IEEE Access* 9: 152476–502. doi:10.1109/ACCESS.2021.3124262.

Devendran, Vamshee Krishna, Hossain Shahriar, and Victor Clincy. 2015. “A Comparative Study of Email Forensic Tools.” *Journal of Information Security* 06(02): 111–17. doi:10.4236/jis.2015.62012.

Ghafarian, Ahmad, Ash Mady, and Kyung Park. 2020. “An Empirical Analysis of Email Forensics Tools.” *International Journal of Network Security & Its Applications* 12(3): 39–57. doi:10.5121/ijnsa.2020.12303.

Singh Chhabra, Gurpal, Gurpal Singh Chhabra Asst Professor, and Dilpreet Singh Bajwa Asst Professor. *Review of E-Mail System, Security Protocols and Email Forensics*. https://www.researchgate.net/publication/286053691.