- **Assignment No:** 1

- **Problem Statement:** Write a program for Tracking Emails & Investigating Email Crimes. i.e. Write a program to analyze e–mail header

- **Objectives:**
  - To extract key details from an email header, such as sender, recipient, subject, and date.
  - To trace the path of an email through mail servers using Received headers and IP addresses.
  - To generate a clear and readable report for analyzing and investigating emails.

- **Hardware Requirements:** Intel Core i3 or higher Processor, 4 GB RAM, 250 GB Hard Disk

- **Software Requirements:** Windows 10 / Linux / macOS Operating System, PyCharm, VS Code, or any Python IDE

- **Theory:**
  - Tracking Emails & Investigating Email Crimes

    Emails are one of the most widely used forms of communication in the digital world. However, they are also commonly misused for phishing, spamming, hacking, and other cybercrimes. To investigate suspicious emails or track the source of an email, it is essential to analyze the email header, which contains metadata about the email.

    1. Email Structure

       An email consists of two main parts:

    a. Header: Contains metadata about the email, including:
       - From: Sender’s email address.
       - To: Recipient’s email address.
       - Subject: Topic of the email.
       - Date: When the email was sent.
       - Message-ID: Unique identifier for the email.

   - Received Headers: Shows the path the email traveled through various mail servers.
   - Other fields: Like CC, BCC, MIME version, and content type.

b. Body: The actual content of the email.

2. Importance of Email Headers

   The email header provides crucial information for investigating cybercrimes:

   - Traceability: Helps trace the IP addresses of mail servers that handled the email.
   - Detection of Spoofing: Reveals inconsistencies or forged headers.
   - Source Verification: Identifies the originating server or sender.
   - Timeline Analysis: Tracks when the email passed through different servers.

3. Analyzing Email Headers

   Email headers can be analyzed manually or programmatically using tools. Key steps include:

   - Extract the header: Separate the header from the body of the email.
   - Parse important fields: Extract From, To, Subject, Date, Message-ID.
   - Track routing path: Examine all Received headers to list all IP addresses and servers the email passed through.
   - Investigate IP addresses: Optionally, perform geolocation or check if IPs are blacklisted.

4. Applications

   - Cybercrime investigation: Helps law enforcement and cybersecurity experts track spammers, phishers, and hackers.
   - Email authentication: Verifies the legitimacy of an email.
   - Network security: Monitors suspicious emails within an organization.

❖ **Conclusion:**

This practical shows how email headers can be analyzed to track the source of emails and detect suspicious activity. The program extracts key details and routing information, making email investigation faster and easier.

- **Code:**

```
import re
from email import message_from_string

def extract_ips(received_lines):
    """Extract IP addresses from Received headers."""
    ips = []
    ip_pattern = r'\[([0-9]{1,3}(?:\.[0-9]{1,3}){3})\]'
    for line in received_lines:
        found_ips = re.findall(ip_pattern, line)
        if found_ips:
            ips.extend(found_ips)
    return ips

def extract_header(full_email):
    """
    Extracts the header from the full email text.
    Header ends at the first empty line.
    """
    lines = full_email.splitlines()
    header_lines = []
    for line in lines:
        if line.strip() == "":
            break
        header_lines.append(line)
    return "\n".join(header_lines)

def analyze_email_header(header):
    """Analyze the email header and display information."""
    msg = message_from_string(header)

    from_email = msg.get("From")
    to_email = msg.get("To")
    subject = msg.get("Subject")
    date = msg.get("Date")
    message_id = msg.get("Message-ID")

    received_lines = msg.get_all("Received") or []
    email_path = extract_ips(received_lines)

    if not from_email and not to_email:
        print("Error: No valid email header found.")
        return

    print("\n=== Email Analysis Report ===")
    print(f"From       : {from_email}")
    print(f"To         : {to_email}")
    print(f"Subject    : {subject}")
    print(f"Date       : {date}")
    print(f"Message ID : {message_id}")

    print("\nEmail Routing Path (IP addresses):")
    if email_path:
        for i, ip in enumerate(email_path, 1):
            print(f"{i}. {ip}")
    else:
        print("No routing IPs found.")

# Main Program
print("=== Email Header Analyzer ===")
print("Paste the full email (header + body). End input with an empty line:")

user_input_lines = []
while True:
    line = input()
    if line.strip() == "":
        break
    user_input_lines.append(line)
```

```
full_email_text = "\n".join(user_input_lines)
header_text = extract_header(full_email_text)
analyze_email_header(header_text)
```

- **Output:**

```
=== Email Header Analyzer ===
Paste the full email (header + body). End input with an empty line:
From: example@example.com
To: someone@example.net
Subject: Test Email
Date: Mon, 7 Oct 2025 09:59:45 +0530
Received: from mail.example.com ([192.168.1.1]) by smtp.example.net

Hello, this is the email body.



=== Email Analysis Report ===
From       : example@example.com
To         : someone@example.net
Subject    : Test Email
Date       : Mon, 7 Oct 2025 09:59:45 +0530
Message ID : None

Email Routing Path (IP addresses):
1. 192.168.1.1
```