Assignment 1

Aashray Gupta (2021ucs0081)
Aarav Jain (2021ucs0080)

(If the portal is unreachable, please reach out to any one of us. Most probably, shutting down of VM to save cost is causing that)

Email Tracker Project (Github Link)

Overview

This project implements an **Email Tracking System** designed to send emails, track when they are opened, and record key metrics related to user interactions. The system provides detailed logs on email activities, including whether the email was opened, the IP address, user agent, and geographic location of the recipient, along with the time taken to open the email after it was sent.

The project is designed with a modern user interface for sending emails and viewing email activity in a dashboard, making it a complete and user-friendly solution for email tracking.

Technologies Used

Backend:

- **Python (Flask):** The core backend framework used for creating RESTful APIs, managing email tracking, and serving web pages.
- **SQLite:** A lightweight relational database used to store email metadata and tracking information.
- **GeoIP API:** A third-party service used to map IP addresses to geographical locations for tracking.

Frontend:

- **HTML & CSS:** For designing interactive pages, including a form for sending emails and a dashboard for visualizing email activity.
- **JavaScript (Fetch API):** To handle asynchronous communication with the backend for sending emails and processing tracking events.

Email Sending Pipeline:

1. **SMTP Protocol:** Emails are sent through an SMTP server with customized headers and embedded tracking pixels.
2. **Dynamic Tracking Links:** Embedded tracking links in the email detect user interactions, such as email opening.

Email Tracking Pipeline:

1. **Tracking Links:** Each email contains unique tracking links with an embedded email ID and sequence ID.
2. **Flask Route:** A specific Flask route logs the opening event when the recipient interacts with the email.
3. **GeoIP Integration:** The IP address from the tracking request is processed via the GeoIP API to obtain location data.
4. **Database Logging:** All email activity, including timestamps, location, and time-to-open, is stored in the SQLite database.

Deployment:

- **Azure VM:** The application is deployed on an Ubuntu-based virtual machine hosted on Microsoft Azure.

Accomplishments

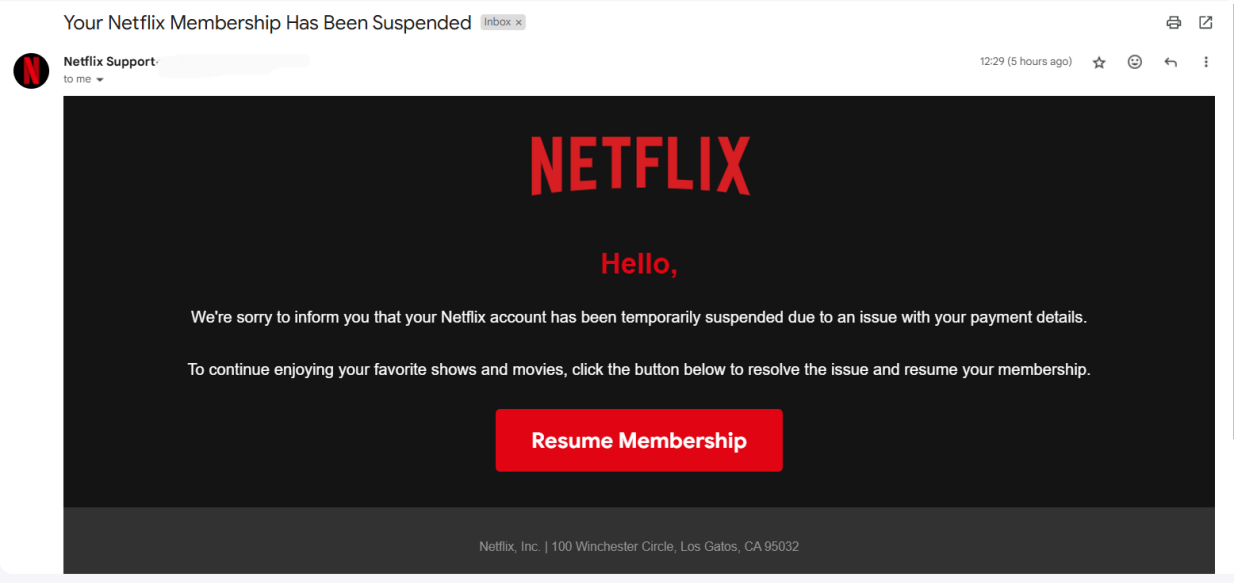
1. **Comprehensive Email Tracking:** Logs key metrics such as email opens, recipient location, user agent, and time-to-open.
2. **Dynamic Dashboard:** Displays real-time email data with a user-friendly interface.

3. **Asynchronous Interaction:** Modern front-end communication with the backend ensures smooth operations without full-page reloads.
4. **Ease of Deployment:** Simple pipeline for local or cloud-based deployment.

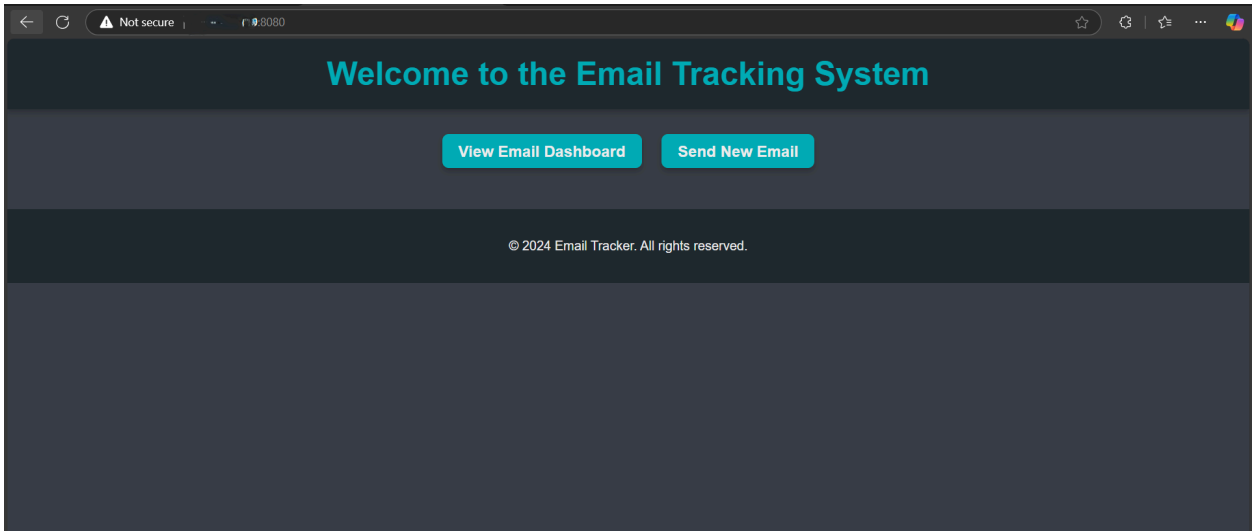
This project serves as an excellent example of integrating modern web technologies to solve practical use cases in email communication and analytics.

Some Screenshots:

1. The Received Email:

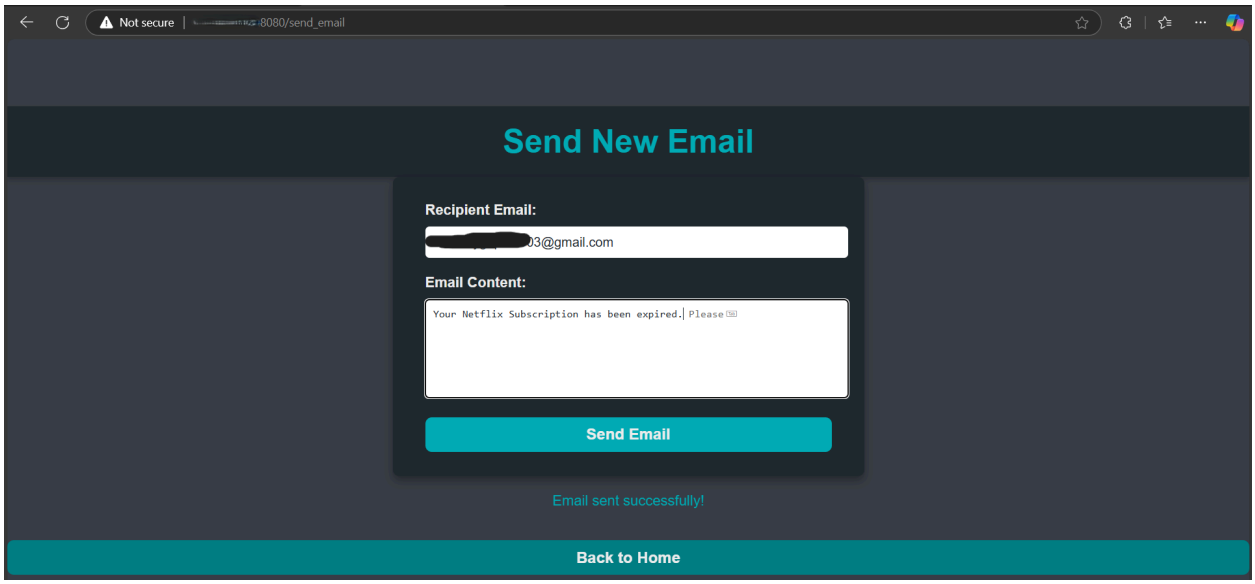


2. The Landing page of portal:



3. Email Tracking Dashboard:

4. Sending Mail portal:



Control Flow:

1. The sender composes mail using the sending mail portal.
2. The mail is received by the victim, which when clicks the flashy button, generates a POST request on the server.
3. The server uses the headers present in HTML to reverse engineer the details.
4. It is even able to detect the correct location (Indian Institute of Technology, Jammu, Jagti) in screenshot 3, line 6.