Software Development Assignment – Lucid Growth

📌 Overview

Build a system that can automatically identify the receiving chain and the ESP type of any email sent to it using "IMAP".
The system must provide a responsive user interface and a structured backend, showcasing both technical accuracy and creativity.

Your work will be assessed based on:

- Functionality (does it work as expected?)
- Code quality (clean, modular, and documented)
- UI/UX (clarity, creativity, responsiveness)
- Deployment (working live demo)
- Documentation (setup & usage instructions)

📝 Problem Statement

Email communication may look simple to end users, but under the hood, every email travels across multiple servers and passes through different systems before it reaches the inbox. These details are stored inside the **email headers**, which contain valuable information such as:

- The **path the email traveled** (called the *receiving chain*).
- The **service that was used to send it** (called the *ESP – Email Service Provider*).

In this challenge, you will build a **full-stack application using "IMAP"** that simulates how professional email analysis tools work.

Your application will:

1. Receive Incoming Emails

- Your project must **generate and display an email address** where test emails will be sent.
- It should also **display a subject line** that your backend will use to identify the correct email among all incoming emails.

2. Process the Email Automatically (Use IMAP Only)

- When a user sends an email to the displayed email address, your system must detect it.
- Using the subject line, your system should ensure it is analyzing the **correct test email**.
- From this email, your system must extract two key pieces of information:
    - **Receiving Chain**: The sequence of servers the email passed through before reaching your inbox.
    - **ESP Type**: The provider used to send the email (for example: Gmail, Outlook, Amazon SES, Zoho, etc.).
        - *Note: this is the sender's ESP, not your own mailbox provider.*

3. Present the Results in a User-Friendly Way

- Your frontend should show:
    - The **receiving chain**.
    - The **ESP type** of the test email we sent.
- The UI must be **clean, responsive, and intuitive**, so that anyone (even non-technical users) can easily understand the results.

🔍 References

To help you understand email headers and ESP detection better, you may use:

- **Google Message Header Analyzer** → https://toolbox.googleapps.com/apps/messageheader/
- **InboxDoctor Free Tests** → https://inboxdoctor.ai/

These tools will give you hands-on examples of how email headers look and how ESPs can be identified.

💻 Tech Stack (Mandatory)

- **Frontend**: React.js or Next.js
- **Backend**: Node.js with NestJS framework
- **Database**: MongoDB

⚙️ Requirements

Frontend (React/Next.js)

- Responsive, mobile-friendly UI
- Dashboard/page to display processed email details
- Present email receiving chain visually (e.g., timeline, table, or creative visualization)
- Show ESP type in a clear, intuitive way
- Use your creativity to make the UI professional and engaging

Backend (Node.js with NestJS)

- API to process and store email data
- Extract headers to determine the receiving chain
- Logic to detect ESP type
- Follow proper folder structure & best practices
- Write clear comments and documentation in the code

Database (MongoDB)

- Store incoming email logs (raw + processed)
- Store metadata such as ESP type, timestamps, etc.

📦 Deliverables

Code Repository

- Push your complete code to a **public GitHub repository**.
- Demo Video using Loom or any other screen recording tool. (Just a simple screen recording showcasing your work)
- Include a detailed **README file**.

Deployment (Live Demo)

- Deploy the project to a live server (e.g., Vercel/Netlify for frontend, Render/Heroku/AWS for backend).
- Share the **live URL** for both frontend and backend (if separate).

Submission

- GitHub Repository Link
- Live Deployment Link(s)

⏰ Deadline

All submissions must be completed by:
🗓️ **Wednesday, 3rd Sept 2025 – 8:00 PM IST**