Full-Stack Email Aggregator Project - Source Code

```
// Backend (Node.js, Express, IMAP, Elasticsearch, AI Processing)
const express = require("express");
const { simpleParser } = require("mailparser");
const { Client } = require("@elastic/elasticsearch");
const nodemailer = require("nodemailer");
const imap = require("imap-simple");
const OpenAI = require("openai-api");
const { WebClient } = require("@slack/web-api");

const app = express();
const PORT = 3000;

const elasticClient = new Client({ node: "http://localhost:9200" });
const openai = new OpenAI(process.env.OPENAI_API_KEY);
const slackClient = new WebClient(process.env.SLACK_TOKEN);

async function syncEmails() {
  const config = {
    imap: {
      user: process.env.EMAIL_USER,
      password: process.env.EMAIL_PASS,
      host: "imap.gmail.com",
      port: 993,
      tls: true,
      authTimeout: 10000,
    },
  };

  const connection = await imap.connect(config);
  await connection.openBox("INBOX");

  const messages = await connection.search(["ALL"], { bodies: "" });
  messages.forEach(async (msg) => {
    const parsed = await simpleParser(msg.parts[0].body);
    await elasticClient.index({
      index: "emails",
      body: {
        subject: parsed.subject,
        body: parsed.text,
        date: parsed.date,
      },
    });
  });
}

app.post("/categorize", async (req, res) => {
  const { subject, body } = req.body;
  const aiResponse = await openai.complete({ engine: "text-davinci-003", prompt: `Categorize this email: ${body}` });
  res.json({ category: aiResponse.data.choices[0].text.trim() });
});

app.post("/slack", async (req, res) => {
  await slackClient.chat.postMessage({ channel: "#notifications", text: "New Interested Email!" });
  res.json({ success: true });
});

app.listen(PORT, () => console.log(`Server running on port ${PORT}`));



// Frontend (React.js - Basic UI for Email Management)
export default function App() {
  const [emails, setEmails] = useState([]);

  useEffect(() => {
    fetch("/emails").then((res) => res.json()).then(setEmails);
  }, []);

  return (
    <div className="p-6">
      <h1 className="text-xl font-bold">Email Aggregator</h1>
      {emails.map((email) => (
        <div key={email.id} className="border p-2 my-2">
          <h2>{email.subject}</h2>
          <p>{email.body}</p>
        </div>
      ))}
    </div>
  );
}
```