

Email Deliverability Tool - Corrected Starter Code (Representative Files)

--- backend/package.json ---

```
{  "name": "email-deliverability-backend",  "version": "1.0.0",  "main": "server.js",  "scripts": {    "start": "node server.js",    "dev": "nodemon server.js"  },  "dependencies": {    "axios": "^1.4.0",    "body-parser": "^1.20.2",    "express": "^4.18.2",    "mongoose": "^7.0.0",    "nodemailer": "^6.9.3",    "uuid": "^9.0.0",    "dotenv": "^16.0.0",    "googleapis": "^126.0.0"  },  "devDependencies": {    "nodemon": "^2.0.0"  } }
```

--- backend/src/routes/tests.js ---

```
const express = require('express'); const Test = require('../models/Test'); const generateCode = require('../utils/generateCode'); const poller = require('../services/poller'); const router = express.Router(); // Create a new test (generate code) router.post('/', async (req, res) => {  try {    const { userEmail } = req.body;    if (!userEmail) return res.status(400).json({ error: 'userEmail required' });    const testCode = generateCode();    const inboxes = [      { provider: 'gmail', address: 'test1+inbox@gmail.com' },      { provider: 'gmail', address: 'test2+inbox@gmail.com' },      { provider: 'outlook', address: 'test3@outlook.com' },      { provider: 'yahoo', address: 'test4@yahoo.com' },      { provider: 'custom', address: 'test5@yourdomain.com' }    ];    const test = new Test({ testCode, userEmail, inboxes });    await test.save();    res.json({ testId: test._id, testCode, inboxes });  } catch (err) {    console.error('Create test error:', err);    res.status(500).json({ error: 'server error' });  } } });
```

--- frontend/src/pages/Home.js ---

```
import React, { useEffect, useState } from 'react'; import API from '../api'; export default function Home() {  const [inboxes, setInboxes] = useState([]);  const [email, setEmail] = useState('');  const [test, setTest] = useState(null);  const [status, setStatus] = useState(null);  useEffect(() => {    API.get('/inboxes').then(res => setInboxes(res.data)).catch(err => console.error(err));  }, []);  async function createTest() {    if (!email) return alert('Enter your email to receive report');    try {      const res = await API.post('/tests', { userEmail: email });      setTest(res.data);    } catch (err) {      console.error(err);      alert('Error creating test');    }  }  async function startTest() {    if (!test) return;    try {      await API.post(`/tests/${test.testId}/start`);      setStatus('running');      pollResults();    } catch (err) {      console.error(err);      alert('Error starting test');    }  }  async function pollResults() {    const interval = setInterval(async () => {      try {        const res = await API.get(`/tests/${test.testId}`);        if (res.data.status === 'completed' || res.data.status === 'failed') {          clearInterval(interval);          setStatus(res.data.status);          setTest(res.data);        } else {          setTest(res.data);        }      } catch (err) {        console.error(err);      }    }, 3000);  }  return (    <div style={{ padding: 24 }}>      <h1>Email Deliverability Tester</h1>      <h3>Test Inboxes</h3>      <ul>        {inboxes.map((i) => <li key={i.address}>{i.provider} - {i.address}</li>)}      </ul>      {!test && (        <div>          <label>Your email (for report): </label>          <input value={email} onChange={e=>setEmail(e.target.value)} />          <button onClick={createTest}>Generate Test Code</button>        </div>      )}      {test && (        <div style={{ marginTop: 20 }}>          <h3>Test Code</h3>          <code style={{ fontSize: 20, padding: 8, display: 'inline-block' }}>{test.testCode}</code>          <p>Send an email from your email client to the addresses above. Include the test code in the subject or body.</p>          <button onClick={startTest} disabled={status==='running'}>Start Test</button>        </div>      )}      {test && (        <div style={{ marginTop: 20 }}>          <h3>Results</h3>          <pre>{JSON.stringify(test, null, 2)}</pre>          {status === 'completed' && <a href={`~/report/${test.testId}`}>View Report</a>        </div>      )}    </div>  ); }
```