

Frontend Code - Email Deliverability Tool

Generated: 2025-10-16T16:47:55.634935 UTC

--- package.json ---

```
{ "name": "email-deliverability-frontend", "version": "1.0.0", "private": true, "dependencies": { "react": "^18.2.0", "react-dom": "^18.2.0", "axios": "^1.4.0", "react-router-dom": "^6.4.2" }, "scripts": { "start": "react-scripts start", "build": "react-scripts build" } }
```

--- src/index.js ---

```
import React from 'react'; import { createRoot } from 'react-dom/client'; import App from './App'; import './styles.css'; const root = createRoot(document.getElementById('root')); root.render(<App/>);
```

--- src/App.js ---

```
import React from 'react'; import { BrowserRouter, Routes, Route } from 'react-router-dom'; import Home from './pages/Home'; import Report from './pages/Report'; export default function App() { return (<BrowserRouter><Routes><Route path="/" element={<Home/>}><Route path="/report/:id" element={<Report/>}</Routes></BrowserRouter>); }
```

--- src/api.js ---

```
import axios from 'axios'; const API = axios.create({ baseURL: process.env.REACT_APP_API_BASE || 'http://localhost:4000/api' }); export default API;
```

--- 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'); 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/${test.testId}/start`); setStatus('running'); pollResults(); } catch (err) { console.error(err); alert('Error starting test'); } } function pollResults() { const interval = setInterval(async () => { try { const res = await API.get(`/tests/${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 className="container"><h1>Email Deliverability Tester</h1><div className="card"><h3>Test Inboxes</h3><ul>{inboxes.map(i => <li key={i.address}>{i.provider} - {i.address}</li>)}</ul><div><label>Your email: </label><input value={email}> <button onClick={createTest}>Generate Test Code</button></div>{test && <div><h4>Test Code: <code>{test.testCode}</code></h4><p>Include this code in subject or body when sending email to test inboxes.</p><button onClick={startTest} disabled={status === 'running'}>Start Test</button></div></div>{test && <div className="card"><h3>Results</h3><pre>{JSON.stringify(test, null, 2)}</pre>{status === 'completed' && <a href={`report/${test.testId}`}>View Report</a></div>}; }
```

--- src/pages/Report.js ---

```
import React, { useEffect, useState } from 'react'; import { useParams } from 'react-router-dom'; import API from '../api'; export default function Report() { const { id } = useParams(); const [test, setTest] = useState(null); useEffect(() => { API.get(`/tests/${id}`).then(res => setTest(res.data)).catch(err => console.error(err)); }, [id]); if (!test) return <div>Loading...</div>; return (<div className="container"><h1>Deliverability Report</h1><p>Test code: <strong>{test.testCode}</strong></p><p>Status: {test.status}</p><p>Score: {test.score ?? 'N/A'}</p><h3>Inbox Results</h3><ul>{test.inboxes.map(i => <li key={i.address}><strong>{i.address}</strong> - Received: {String(i.received)} - Folder: {i.folder ?? 'Unknown'}</li>)}</ul></div>); }
```

--- src/styles.css ---

```
body { font-family: Arial, Helvetica, sans-serif; background: #f7f7f8; color: #222 } .container { max-width: 900px; margin: 24px auto; padding: 16px } .card { background: #fff; padding: 16px; border-radius: 8px; box-shadow: 0 2px 6px rgba(0, 0, 0, 0.06); margin-top: 12px } input { padding: 8px; margin: 8px } button { padding: 8px 12px; margin: 8px } code { background: #eee; padding: 4px 6px; border-radius: 4px }
```