React Email Verification Setup Guide - **Only for Local Projects**

This document outlines the steps to implement email verification in a React application using Mailpit or other services. Email verification is essential to ensure the authenticity of user-provided email addresses during registration.

1. Prerequisites

- Node.js and npm installed
- React app already created (e.g., using Create React App)
- Backend server (Node.js, Express, or any other)
- Mailpit or Mailtrap installed for local email testing

2. What is Mailpit?

Mailpit is a lightweight email testing tool that acts like a fake SMTP server. It catches all outgoing emails and shows them via a web interface instead of sending them to real users.

3. Setting up Mailpit

1. Download Mailpit from https://github.com/axllent/mailpit/releases.
2. Run Mailpit with: `./mailpit` or `mailpit.exe`
3. Access the web interface at http://localhost:8025
4. Set your backend to use SMTP host: `localhost`, port: `1025`

4. Example Backend Email Sending Code (Node.js)

Use Nodemailer to send emails through Mailpit:

```
const nodemailer = require('nodemailer');

const transporter = nodemailer.createTransport({
  host: 'localhost',
  port: 1025,
  secure: false,
  tls: { rejectUnauthorized: false }
});

const sendVerificationEmail = async (to, token) => {
  const link = `http://localhost:3000/verify?token=${token}`;

  await transporter.sendMail({
    from: '"Homemade Heaven" <no-reply@homemade.com>',
    to,
    subject: 'Email Verification',
    html: `<p>Click the link to verify your email: <a
href="${link}">Verify Email</a></p>`
  });
};
```

---

5. Alternatives to Mailpit

- [Mailtrap (https://mailtrap.io)](https://mailtrap.io) – Reliable GUI with 500 emails/month free.
- [Ethereal Email (https://ethereal.email)](https://ethereal.email) – Fake SMTP service for test/dev usage.

6. Frontend Workflow (React)

1. User fills registration form and submits.
2. Backend stores user as unverified and sends email.
3. User receives email with verification link.
4. Clicking link verifies the account by calling backend endpoint.

7. Sample Verification Page in React

Create a page like `VerifyPage.jsx` to read token and call backend to verify.

```
useEffect(() => {
  const token = new
URLSearchParams(window.location.search).get('token');
  axios.post('/api/verify', { token }).then(res => setStatus('Verified!'));
}, []);
```

8. Final Notes - Important

- Do not use Mailpit in production.
- For real deployments, consider services like SendGrid or Amazon SES.
- Always encrypt user data and secure email links with expiry.

-- Mailpit is not only the plate form where you can do this in upcoming Docs you will find some other paths also…