Perfect catch — email is **crucial** for a professional SaaS. Let’s now **finalize the full setup** of your product **LinkShield Pro**, with **email alerts, user verification, and security notifications** fully integrated.

**✅ Finalized Feature Stack (Updated with Email)**

| **Layer** | **Feature** | **Tools/Libs** |
| --- | --- | --- |
| **🧠 Core** | URL Shortener + Safety Scan | Express, MongoDB, NanoID, Google Safe Browsing |
| **🧪 Analytics** | Track IP, location, browser, device, timestamp | user-agent, ip-api.com |
| **🔐 Auth** | JWT login/register, cookie sessions | jsonwebtoken, bcryptjs |
| **📩 Email** | ✅ Alert users on flagged links, login, etc. | nodemailer with Gmail / Mailgun / SendGrid |
| **📊 User Panel** | Dashboard with link stats and click logs | EJS + Tailwind |
| **⚙️ Admin Panel** | Moderate links/users, view all data | Custom routes |
| **📱 Anonymous Use** | 3 links max, tracked via localStorage or IP | Cookie parser, DB logic |

**📬 Email Use Cases in LinkShield Pro**

| **Event** | **Email Type** |
| --- | --- |
| ✅ User signup | Welcome + optional verification link |
| 🔐 Login from new IP | Security alert |
| ⚠️ Link flagged unsafe | Alert with link info |
| 📈 Weekly stats summary (optional) | Email with top-performing links |
| 🔁 Forgot password | Reset password link |

**📁 Updated Folder Structure**

linkshield-pro/

├── controllers/

│ ├── authController.js

│ ├── linkController.js

│ ├── analyticsController.js

│ └── adminController.js

├── models/

│ ├── User.js

│ ├── Link.js

│ └── Click.js

├── routes/

│ ├── authRoutes.js

│ ├── linkRoutes.js

│ ├── analyticsRoutes.js

│ └── adminRoutes.js

├── middleware/

│ ├── auth.js

│ ├── limitFreeLinks.js

│ ├── verifySafeLink.js

│ └── verifyJWT.js

├── services/

│ ├── geolocationService.js

│ ├── safeBrowsingService.js

│ ├── tracker.js

│ └── emailService.js ✅

├── utils/

│ └── generateToken.js

├── views/

│ ├── index.ejs

│ ├── dashboard.ejs

│ ├── login.ejs

│ ├── register.ejs

│ ├── emailTemplates/

│ │ ├── welcome.ejs

│ │ ├── linkFlagged.ejs

│ │ └── passwordReset.ejs

│ └── admin.ejs

├── public/

│ ├── js/

│ └── style.css

├── .env

├── app.js

├── Dockerfile

├── README.md

└── .github/workflows/deploy.yml

**🔑 .env Example**

PORT=3000

MONGODB\_URI=mongodb+srv://your-uri

JWT\_SECRET=your-jwt

EMAIL\_USER=your@gmail.com

EMAIL\_PASS=your-email-password-or-app-key

GOOGLE\_API\_KEY=your-safe-browsing-key

**📬 EmailService Example (Nodemailer)**

// services/emailService.js

const nodemailer = require('nodemailer');

const ejs = require('ejs');

const path = require('path');

const transporter = nodemailer.createTransport({

service: 'gmail', // or use SendGrid, Mailgun

auth: {

user: process.env.EMAIL\_USER,

pass: process.env.EMAIL\_PASS

}

});

async function sendEmail(template, subject, to, data = {}) {

const templatePath = path.join(\_\_dirname, '..', 'views', 'emailTemplates', `${template}.ejs`);

const html = await ejs.renderFile(templatePath, data);

return transporter.sendMail({

from: process.env.EMAIL\_USER,

to,

subject,

html

});

}

module.exports = { sendEmail };

**📧 How Emails Are Triggered**

| **File** | **When it triggers** |
| --- | --- |
| authController.js | On register/login |
| linkController.js | When a link is flagged |
| adminController.js | When admin sends warning or link is deleted |