

Resend Email Migration Guide

Date: January 6, 2026

Status:  Completed - Deployed to GitHub

Commit: `3f67084` - feat: Switch to Resend API for email delivery

Summary

Successfully migrated the email system from SMTP (nodemailer with Gmail) to **Resend API** for simpler, more reliable transactional email delivery.

Benefits of Resend

-  **No SMTP configuration** - Simple API integration
-  **Higher deliverability** - Modern email infrastructure
-  **Professional templates** - Clean HTML email design
-  **Easy setup** - Just one API key needed
-  **Better monitoring** - Dashboard to track email delivery

Changes Made

1. Created New Email Utility (`src/lib/email.ts`)

Purpose: Centralized email service using Resend API

Key Features:

- `sendVerificationEmail()` - Send verification emails to new users
- `sendPasswordResetEmail()` - Send password reset emails (future use)
- Professional HTML email templates with modern design
- Automatic fallback for missing API key (logs error)
- Plain text fallback for email clients that don't support HTML

Template Design:

- Gradient blue header with CareLinkAI branding
- Clean, responsive layout (mobile-friendly)
- Professional "Verify Email Address" button
- 24-hour expiry notice
- Footer with manual link option

2. Updated Registration Route (`src/app/api/auth/register/route.ts`)

Changes:

- Removed `nodemailer` import
- Added `import { sendVerificationEmail } from "@lib/email"`
- Replaced SMTP email function with `sendVerificationEmailToUser()`
- Simplified email sending logic (no transporter configuration)

Before:

```
import * as nodemailer from "nodemailer";
// 150+ lines of SMTP configuration and email sending
```

After:

```
import { sendVerificationEmail } from "@/lib/email";
// Simple function call with 3 parameters
```

3. Updated Resend Verification Route (`src/app/api/auth/resend-verification/route.ts`)**Changes:**

- Removed `nodemailer` import
- Added `import { sendVerificationEmail } from "@/lib/email"`
- Removed 120+ lines of SMTP email function
- Now uses centralized email utility

Result: Cleaner, more maintainable code

4. Updated Environment Configuration**Files Modified:**

- `.env.example` - Added RESEND_API_KEY documentation
- `.env` - Added placeholder for local development

New Environment Variable:

```
RESEND_API_KEY=rexxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

**Deployment Steps****Step 1: Set Up Resend API Key on Render**

The application has been pushed to GitHub and will auto-deploy to Render. However, **you need to add the RESEND_API_KEY environment variable** for emails to work.

Option A: Using Render Dashboard (Recommended)**1. Go to Render Dashboard:**

- Visit: <https://dashboard.render.com/>
- Log in with your account

2. Select Your Service:

- Find “carelinkai” web service
- Click on it to open service details

3. Navigate to Environment:

- Click on “Environment” in the left sidebar

4. Add New Environment Variable:

- Click “Add Environment Variable”
- **Key:** RESEND_API_KEY
- **Value:** Your Resend API key (starts with re_)
- Click “Save Changes”

5. Trigger Manual Deploy:

- Click “Manual Deploy” → “Deploy latest commit”
- Or wait for automatic deployment from GitHub push

Option B: Get Resend API Key

If you don't have a Resend API key yet:

1. Visit Resend:

- Go to: <https://resend.com/>

2. Sign Up / Log In:

- Create account with profyt7@gmail.com
- Or log in if account exists

3. Create API Key:

- Go to: <https://resend.com/api-keys>
- Click “Create API Key”
- Name: “CareLinkAI Production”
- Copy the API key (starts with re_)

4. Add to Render:

- Follow “Option A” steps above
- Paste your API key

Step 2: Verify Domain (Optional but Recommended)

For production use, verify your domain to send from noreply@getcarelinkai.com :

1. In Resend Dashboard:

- Go to: <https://resend.com/domains>
- Click “Add Domain”
- Enter: getcarelinkai.com

2. Add DNS Records:

- Copy the TXT, MX, and CNAME records
- Add them to your domain’s DNS settings
- Wait for verification (can take up to 48 hours)

3. Update Email Utility:

- Edit src/lib/email.ts
- Change FROM_EMAIL from profyt7@gmail.com to noreply@getcarelinkai.com
- Commit and push changes

Until domain is verified: Emails will be sent from profyt7@gmail.com

Step 3: Test Email Functionality

After deployment with RESEND_API_KEY set:

1. Test Registration:

- Go to: <https://getcarelinkai.com/auth/register>
- Register a new test account
- Check email inbox for verification email

2. Test Resend Verification:

- Go to: <https://getcarelinkai.com/auth/login>
- Click “Resend verification email”
- Check inbox for new verification email

3. Check Resend Dashboard:

- Go to: <https://resend.com/emails>
- Verify emails are being sent successfully
- Check delivery status and logs



Implementation Details

Email Template Features

HTML Email Includes:

- Responsive design (works on all devices)
- Gradient blue header with branding
- Clear call-to-action button
- Information box with expiry notice
- Footer with manual link option
- Professional typography and spacing

Plain Text Email Includes:

- All important information
- Clickable verification link
- Expiry time notice
- Professional formatting

Error Handling

Missing API Key:

```
if (!process.env.RESEND_API_KEY) {
  console.error('[Resend] RESEND_API_KEY is not configured');
  console.error('[Resend] Please set RESEND_API_KEY on Render dashboard');
  return false;
}
```

Email Sending Errors:

- Logs detailed error information
- Returns `false` to caller
- Registration still succeeds (user can resend verification)

Code Quality

Before (SMTP):

- 150+ lines per email function
- Complex transporter configuration
- Separate dev/prod logic
- Harder to maintain

After (Resend):

- 10-15 lines per email call
 - Single utility file
 - Centralized templates
 - Easy to maintain
-

Testing Checklist

Local Testing (Development)

- [x] Build succeeds without errors
- [x] No TypeScript errors
- [x] Code compiles successfully
- [x] Changes committed to Git
- [x] Pushed to GitHub

Production Testing (After Deployment)

- [] Set RESEND_API_KEY on Render
 - [] Deploy succeeds on Render
 - [] Registration sends verification email
 - [] Email arrives in inbox (check spam folder too)
 - [] Verification link works
 - [] Resend verification works
 - [] Check Resend dashboard for logs
-

File Summary

Files Created (1)

1. `src/lib/email.ts` (335 lines)
 - Resend email utility with templates
 - Verification and password reset functions

Files Modified (4)

1. `src/app/api/auth/register/route.ts`
 - Replaced nodemailer with Resend
 - Simplified email sending logic
 - Reduced code complexity

2. `src/app/api/auth/resend-verification/route.ts`
 - Replaced nodemailer with Resend
 - Removed duplicate email function
 - Uses centralized utility
3. `.env.example`
 - Added RESEND_API_KEY documentation
 - Includes link to get API key
4. `.env`
 - Added placeholder for local development

Total Changes

- **Lines Added:** ~400
 - **Lines Removed:** ~270
 - **Net Change:** +130 lines (more features, cleaner code)
-

Next Steps

Immediate (Required)

1. **Set RESEND_API_KEY on Render**
 - Go to Render dashboard
 - Add environment variable
 - Deploy latest commit
2. **Test Registration Flow**
 - Register test account
 - Verify email arrives
 - Confirm link works

Short-term (Recommended)

1. **Verify Domain**
 - Set up `getcarelinkai.com` in Resend
 - Add DNS records
 - Update FROM_EMAIL in code
2. **Monitor Email Delivery**
 - Check Resend dashboard
 - Review delivery rates
 - Investigate any bounces

Long-term (Optional)

1. **Implement Password Reset**
 - Use `sendPasswordResetEmail()` function
 - Create reset password page
 - Test flow end-to-end
2. **Add Email Analytics**
 - Track open rates

- Monitor click rates
- Optimize templates

3. Create Additional Templates

- Welcome emails
 - Tour confirmations
 - Inquiry notifications
-

Troubleshooting

Issue: Emails Not Sending

Symptoms:

- No emails arrive in inbox
- Console logs show "RESEND_API_KEY is not configured"

Solution:

1. Check Render environment variables
2. Ensure RESEND_API_KEY is set correctly
3. Verify API key starts with `re_`
4. Restart Render service

Issue: Emails in Spam Folder

Symptoms:

- Emails arrive but marked as spam
- Low deliverability

Solution:

1. Verify domain in Resend
2. Add SPF, DKIM, DMARC records
3. Use custom domain instead of gmail.com
4. Monitor Resend dashboard for issues

Issue: Build Errors

Symptoms:

- TypeScript errors during build
- Import errors

Solution:

1. Ensure `resend` package is installed: `npm install resend`
2. Check imports: `import { sendVerificationEmail } from '@/lib/email"`
3. Rebuild: `npm run build`

Issue: Links Not Working

Symptoms:

- Verification link returns 404
- Token not recognized

Solution:

1. Check NEXTAUTH_URL environment variable

2. Verify token generation logic
 3. Check token expiry (24 hours)
 4. Test resend verification
-

Support Resources

Resend Documentation

- **API Reference:** <https://resend.com/docs/api-reference/introduction>
- **Domain Setup:** <https://resend.com/docs/dashboard/domains/introduction>
- **Email Best Practices:** <https://resend.com/docs/knowledge-base/deliverability>

CareLinkAI Resources

- **GitHub Repository:** <https://github.com/profy7/carelinkai>
- **Render Dashboard:** <https://dashboard.render.com/>
- **Resend Dashboard:** <https://resend.com/>

Contact

- **Email:** profyt7@gmail.com
 - **GitHub:** @profyt7
-

Verification

Build Status:  Successful

Commit Hash: 3f67084

Branch: main

Deployment: Pushed to GitHub (auto-deploy to Render)

Key Achievement: Simplified email system from 270+ lines of SMTP configuration to a clean, maintainable API integration with better deliverability and monitoring.

Last Updated: January 6, 2026

Author: CareLinkAI Development Team