



Email Verification Fix - Deployment Guide

✓ What's Been Completed

All code changes have been implemented, tested, and pushed to GitHub:

- ✓ Login page shows actual error messages
- ✓ Verification help section with resend functionality
- ✓ Resend verification email API endpoint
- ✓ Production SMTP configuration support
- ✓ Comprehensive documentation
- ✓ Build tested successfully
- ✓ Code committed to Git
- ✓ Pushed to GitHub (commit: 8c031e7)

GitHub: Changes are live at <https://github.com/profyt7/carelinkai>

Status: Ready for production deployment ✓

🎯 What You Need to Do Next

Step 1: Set Up Gmail App Password (5 minutes)

Why: The system needs this to send real verification emails to users.

Instructions:

1. Go to <https://myaccount.google.com/apppasswords>
2. Sign in with profyt7@gmail.com
3. You may need to enable 2-Factor Authentication first if not already enabled
4. Click “Select app” → Choose “Mail”
5. Click “Select device” → Choose “Other (Custom name)”
6. Enter “CareLinkAI Production”
7. Click “Generate”
8. **COPY THE 16-CHARACTER PASSWORD** (format: xxxx xxxx xxxx xxxx)
9. **REMOVE SPACES** to get: xxxxxxxxxxxxxxxx
10. Save this password securely - you’ll need it in Step 2

Important:

- This is NOT your regular Gmail password
 - This is a special “App Password” for applications
 - You can only see it once, so copy it now
-

Step 2: Configure Render Environment Variables (5 minutes)

Why: Render needs these credentials to send emails in production.

Instructions:

1. Go to <https://dashboard.render.com>

2. Find and click on your “carelinkai” service
3. Click “Environment” in the left sidebar
4. Add these 5 new environment variables:

```
SMTP_HOST=smtp.gmail.com
SMTP_PORT=587
SMTP_SECURE=false
SMTP_USER=profyt7@gmail.com
SMTP_PASSWORD=<paste-your-16-char-app-password-here>
```

Example Screenshot Reference:

Key	Value
SMTP_HOST	smtp.gmail.com
SMTP_PORT	587
SMTP_SECURE	false
SMTP_USER	profyt7@gmail.com
SMTP_PASSWORD	abcdabcdabcdabcd Your actual app password

1. Click “Save Changes”
2. Render will automatically redeploy your application (takes ~3-5 minutes)

Important:

- Do NOT use quotes around values
- Password should be 16 characters with no spaces
- `SMTP_SECURE=false` is correct for port 587

Step 3: Monitor Deployment (5 minutes)

Why: To ensure the deployment completes successfully.

Instructions:

1. Stay on the Render dashboard
2. Click “Events” or “Logs” tab
3. Watch for these messages:
 - “Build succeeded”
 - “Deploy succeeded”
 - “Service is live”
4. Look for this log entry when first email is sent:
`[sendVerificationEmail] Using production SMTP`
`✉ Verification email sent:`
`- To: <email-address>`

Expected Time: 3-5 minutes for complete deployment

If deployment fails:

- Check that all 5 environment variables are set correctly
- Check for typos in variable names
- Verify app password doesn't have spaces

Step 4: Test Email Verification (10 minutes)

Why: To confirm the complete flow works end-to-end.

Test Scenarios:

A. Test Registration & Email Delivery

1. Go to <https://getcarelinkai.com/auth/register>
2. Register with a REAL email address (use your personal email for testing)
3. Submit registration form
4. **Check your email inbox** (should arrive within 1-2 minutes)
5. **Check spam folder** if not in inbox
6. Verify you received an email from "CareLinkAI"

Expected Email Content:

- Subject: "Verify Your CareLinkAI Account"
- From: CareLinkAI noreply@carelinkai.com
- Contains blue "Verify Email Address" button
- Contains verification link

B. Test Email Verification Link

1. Open the verification email
2. Click "Verify Email Address" button
3. Should redirect to login page
4. Should see green message: "Your email has been verified. You can now sign in."

C. Test Login After Verification

1. On login page, enter your email and password
2. Click "Sign in"
3. Should log in successfully
4. Should be redirected to dashboard

D. Test Unverified User Error Message

1. Register another test account
2. **DON'T** verify the email
3. Try to log in with that account
4. Should see RED error: "Please verify your email before logging in"
5. Should see BLUE verification help section
6. Should see "Resend Verification Email" button

E. Test Resend Verification

1. From the verification help section above
 2. Click "Resend Verification Email" button
 3. Should see green success message
 4. Should receive another verification email
 5. Click resend 4 times rapidly - 4th attempt should show rate limit error
-

Step 5: Update Existing Users (Optional - 10 minutes)

Why: If you have existing PENDING users who couldn't log in before.

Option A: Let them use “Resend” button

- Users can just try to log in
- They'll see the verification help section
- They can click “Resend Verification Email”
- They'll receive a new verification email

Option B: Update users manually in database

If you want to allow existing users without verification:

```
-- Run this in Render Postgres shell or Prisma Studio
UPDATE "User"
SET status = 'ACTIVE', "emailVerified" = NOW()
WHERE status = 'PENDING' AND email = 'specific@email.com';
```

Recommendation: Use Option A - let users self-serve with resend button.



How to Verify Everything is Working

Check 1: Render Logs Show Production SMTP

```
# In Render logs, look for:
[sendVerificationEmail] Using production SMTP
✉ Verification email sent:
- To: user@example.com
```

If you see:

```
[sendVerificationEmail] Using Ethereal test account
- Preview URL: https://ethereal.email/...
```

→ Environment variables are NOT configured correctly

Check 2: Users Can Register and Receive Email

- Real emails delivered to inbox (or spam)
- Emails contain working verification links
- No “Preview URL” in logs

Check 3: Verification Links Work

- Clicking link redirects to login page
- Shows success message
- User can now log in

Check 4: Error Messages are Clear

- Unverified users see specific verification error
- NOT generic “Invalid email or password”

- Verification help section displays

Check 5: Resend Functionality Works

- Button is visible and clickable
 - Email is sent when clicked
 - Rate limiting works after 3 attempts
-



Troubleshooting

Issue: Emails Still Not Being Sent

Check Render Environment Variables:

1. Go to Render → Environment tab
2. Verify all 5 variables are present:
 - SMTP_HOST
 - SMTP_PORT
 - SMTP_SECURE
 - SMTP_USER
 - SMTP_PASSWORD
3. Check for typos in variable names
4. Verify password has no spaces

Check Render Logs:

1. Go to Render → Logs tab
2. Look for [sendVerificationEmail] entries
3. Check if using “production SMTP” or “Ethereal”
4. Look for error messages

Common Errors in Logs:

- Invalid login → Wrong app password
- Connection timeout → Wrong port or host
- Authentication failed → Check credentials
- 535 Authentication failed → Password has spaces or is incorrect

Issue: Emails Going to Spam

Short-term fix:

- Tell users to check spam folder
- Mark CareLinkAI emails as “Not Spam”

Long-term fix (see SMTP_SETUP_GUIDE.md):

- Set up SPF/DKIM records for your domain
- Or migrate to SendGrid with domain authentication

Issue: Verification Link Not Working

Check:

1. Is NEXTAUTH_URL set correctly in Render?
2. Should be: <https://getcarelinkai.com>
3. Link should match your production domain

Test:

- Copy verification link from email
- Paste in browser
- Should redirect to login page

Issue: “Resend” Button Not Working**Check browser console:**

1. Open browser DevTools (F12)
2. Go to Console tab
3. Click “Resend Verification Email”
4. Look for error messages

Common issues:

- 429 error → Rate limit hit (wait 15 minutes)
 - 500 error → Check Render logs for server error
 - Network error → Check if API endpoint is deployed
-

**Success Metrics to Monitor****Week 1 After Deployment:**

- **Email Delivery Rate:** Should be >95%
- **Verification Rate:** % of users who verify email
- **Login Success Rate:** After verification
- **Resend Usage:** How many users click resend

How to Monitor:

1. **Render Logs:** Check for email sending errors
 2. **Database Queries** (Prisma Studio or SQL):


```
```sql
 - Count users by status
 SELECT status, COUNT(*) FROM "User" GROUP BY status;

 - Recent registrations
 SELECT email, status, "createdAt", "emailVerified"
 FROM "User"
 ORDER BY "createdAt" DESC
 LIMIT 20;
      ```
```

 1. **User Feedback:** Ask users about email delivery
 2. **Support Tickets:** Monitor verification-related issues
-

**Additional Resources****Documentation Created:**

1. **SMTP_SETUP_GUIDE.md:** Detailed SMTP configuration guide

2. **EMAIL_VERIFICATION_FIX_SUMMARY.md**: Complete technical implementation summary
3. **This file**: Deployment and next steps guide

Key Files Modified:

- `src/app/auth/login/page.tsx` : Login page with verification help
- `src/app/api/auth/register/route.ts` : Registration with SMTP support
- `src/app/api/auth/resend-verification/route.ts` : New resend endpoint

Support:

- **GitHub Issues**: <https://github.com/profy7/carelinkai/issues>
 - **Render Support**: <https://render.com/support>
 - **Gmail Help**: <https://support.google.com/accounts/answer/185833>
-

Completion Checklist

Before marking this as complete, verify:

- [] Gmail App Password created and saved securely
 - [] All 5 SMTP environment variables added to Render
 - [] Render deployment completed successfully
 - [] Render logs show “Using production SMTP”
 - [] Test registration sent email to real address
 - [] Email arrived in inbox (or spam)
 - [] Verification link worked
 - [] Able to login after verification
 - [] Unverified login shows clear error message
 - [] “Resend” button works
 - [] Rate limiting prevents spam (4th attempt fails)
-

Rollback Plan (If Needed)

If something goes wrong:

Quick Fix - Disable Email Verification Temporarily

```
# Option 1: Remove SMTP_HOST from Render environment
# This will make system fall back to Ethereal (test emails)
# No real emails will be sent, but registration will work

# Option 2: Revert to previous deployment on Render
1. Go to Render → Deployments tab
2. Find previous successful deployment
3. Click "Redeploy"
```

Full Rollback

```
# Revert Git commit
cd /home/ubuntu/carelinkai-project
git revert 8c031e7
git push origin main

# Render will auto-deploy the rollback
```

📞 Need Help?

If you encounter issues during deployment:

1. **Check Render Logs First:** Most issues are visible in logs
2. **Verify Environment Variables:** Double-check all 5 variables
3. **Test App Password:** Try logging into Gmail with it directly
4. **Review Documentation:** SMTP_SETUP_GUIDE.md has troubleshooting
5. **Check This Checklist:** Ensure all steps completed

⌚ Expected Timeline

- **Step 1 (Gmail App Password):** 5 minutes
- **Step 2 (Render Environment):** 5 minutes
- **Step 3 (Monitor Deployment):** 5 minutes
- **Step 4 (Testing):** 10 minutes
- **Step 5 (Update Users):** Optional

Total Time: ~25-30 minutes

✨ What Changes for Users

Before This Fix:

- ✗ Users registered but couldn't login
- ✗ Saw misleading "Invalid email or password"
- ✗ No guidance on what to do
- ✗ No way to resend verification email
- ✗ Emails never arrived (using test system)

After This Fix:

- ✓ Users receive real verification emails
- ✓ Clear error message: "Please verify your email"
- ✓ Helpful instructions on what to do
- ✓ Can resend verification email with one click
- ✓ Complete working flow: register → verify → login

17

Deployment Information

- **Commit:** 8c031e7
- **Branch:** main
- **Date:** January 6, 2026
- **Status:**  Ready for Production
- **Auto-Deploy:** Will deploy when you add environment variables

 You're ready to deploy! Start with Step 1 above.

Good luck! 