

# SMTP Setup Guide for CareLinkAI Email Verification

---

## Overview

This guide will help you configure production email delivery for CareLinkAI's email verification system. Currently, the system uses Ethereal (a test email service) which doesn't deliver emails to real users. You need to configure a production SMTP service to enable actual email delivery.

## Quick Start - Gmail SMTP (Recommended for Quick Setup)

---

### Prerequisites

- Gmail account (profyt7@gmail.com)
- 2-Factor Authentication enabled on Gmail

### Step 1: Enable 2-Factor Authentication

If you haven't already:

1. Go to <https://myaccount.google.com/security>
2. Under "Signing in to Google", click on "2-Step Verification"
3. Follow the prompts to enable 2FA

### Step 2: Create Gmail App Password

1. Go to <https://myaccount.google.com/apppasswords>
2. You may need to sign in again
3. Under "Select app", choose "Mail"
4. Under "Select device", choose "Other (Custom name)"
5. Enter "CareLinkAI Production" as the name
6. Click "Generate"
7. **IMPORTANT:** Copy the 16-character password (format: xxxx xxxx xxxx xxxx)
  - Save this password securely - you won't be able to see it again
  - Remove spaces when using it: xxxxxxxxxxxxxxxx

### Step 3: Configure Environment Variables on Render

1. Go to your Render dashboard: <https://dashboard.render.com>
2. Select your CareLinkAI web service
3. Click on "Environment" in the left sidebar
4. Add the following environment variables:

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

**Important Notes:**

- `SMTP_SECURE=false` is correct for port 587 (uses STARTTLS)
- Do NOT use your regular Gmail password - use the App Password
- Remove spaces from the app password

**Step 4: Deploy Changes**

1. After adding environment variables, click “Save Changes”
2. Render will automatically redeploy your service
3. Wait for deployment to complete (usually 2-5 minutes)

**Step 5: Test Email Delivery**

1. Try registering a new account with a real email address
2. Check that you receive the verification email
3. Check spam folder if not in inbox
4. Verify the email by clicking the link

**Alternative Option - SendGrid (Recommended for Production Scale)**

SendGrid offers better deliverability and analytics for production use.

**Step 1: Create SendGrid Account**

1. Go to <https://sendgrid.com/>
2. Sign up for free account (100 emails/day free tier)
3. Complete account verification

**Step 2: Create API Key**

1. Go to Settings > API Keys
2. Click “Create API Key”
3. Name it “CareLinkAI Production”
4. Select “Restricted Access”
5. Grant “Mail Send” permission
6. Click “Create & View”
7. Copy the API key (starts with SG.)

**Step 3: Configure Environment Variables on Render**

```
SMTP_HOST=smtp.sendgrid.net
SMTP_PORT=587
SMTP_SECURE=false
SMTP_USER=apikey
SMTP_PASSWORD=<your-sendgrid-api-key>
```

**Important Note:** The `SMTP_USER` is literally the word “apikey”, not your email.

# Testing Email Configuration

---

## Test in Development (Local)

1. Set environment variables in `.env.local` :

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

1. Restart your development server:

```
npm run dev
```

1. Test registration with your email address
2. Verify you receive the email

## Test in Production (Render)

1. After setting environment variables and deploying
2. Go to <https://getcarelinkai.com/auth/register>
3. Register with a test email
4. Check that email is delivered
5. Verify email by clicking link

---

# Troubleshooting

---

## Emails Not Being Sent

### Check Render Logs:

1. Go to Render dashboard
2. Click on your service
3. Click “Logs” tab
4. Look for lines containing `[sendVerificationEmail]`
5. Check for error messages

### Common Issues:

#### “Invalid login” Error

- **Cause:** Using regular Gmail password instead of App Password
- **Fix:** Generate and use a Gmail App Password (see Step 2 above)

#### “Connection timeout” Error

- **Cause:** Port blocked or incorrect
- **Fix:** Ensure `SMTP_PORT=587` and `SMTP_SECURE=false`

#### “Authentication failed” Error

- **Cause:** Wrong credentials or spaces in app password
- **Fix:**

- Verify email address is correct
- Remove all spaces from app password
- Regenerate app password if needed

## Still Using Ethereal (Preview URLs in logs)

- **Cause:** Environment variables not set correctly
- **Fix:** Verify all 5 SMTP variables are set in Render
- Check logs for `[sendVerificationEmail]` Using production SMTP

## Emails Going to Spam

### For Gmail SMTP:

1. Add a custom domain and configure SPF/DKIM records
2. Or accept that some emails may go to spam with Gmail SMTP
3. Tell users to check spam folder

### For SendGrid:

1. Configure Domain Authentication:
    - Go to SendGrid Settings > Sender Authentication
    - Click “Authenticate Your Domain”
    - Follow DNS configuration steps
  2. This significantly improves deliverability
- 

## Security Best Practices

### 1. Protect Your SMTP Credentials

- Never commit SMTP credentials to Git
- Use environment variables only
- Rotate App Passwords periodically

### 2. Monitor Email Sending

- Check Render logs regularly
- Set up alerts for failed email delivery
- Monitor Gmail “Less secure app access” warnings

### 3. Use SendGrid for Production

- Better security than Gmail SMTP
  - More reliable delivery
  - Detailed analytics and logging
  - Easier to debug issues
- 

## Current System Status

### Email Flow

1. **Registration:** User creates account → Status set to PENDING
2. **Email Sent:** Verification email sent with token

**3. User Verifies:** Clicks link in email → Status changed to ACTIVE

**4. Login Allowed:** User can now log in

## Features Implemented

- Email verification required before login
  - Clear error messages showing verification status
  - Verification help section with instructions
  - Resend verification email functionality
  - Rate limiting (3 attempts per 15 minutes)
  - Support for production SMTP
  - Fallback to Ethereal in development
- 

## Next Steps After SMTP Configuration

### 1. Test the Complete Flow:

- Register new account
- Receive verification email
- Click verification link
- Log in successfully

### 2. Update User Communication:

- Inform existing PENDING users to check email
- Provide support email for verification issues

### 3. Monitor System:

- Watch Render logs for email errors
- Track verification success rate
- Monitor spam complaints

### 4. Consider Enhancements:

- Add email verification status page
  - Implement verification reminder emails
  - Add SMS verification as backup
- 

## Support

If you encounter issues during setup:

1. Check Render logs for error messages
2. Verify all environment variables are set correctly
3. Test with a different email provider if issues persist
4. Consider using SendGrid for more reliable delivery

## Quick Reference - Environment Variables

```
# Gmail SMTP
SMTP_HOST=smtp.gmail.com
SMTP_PORT=587
SMTP_SECURE=false
SMTP_USER=profyt7@gmail.com
SMTP_PASSWORD=<16-char-app-password>

# SendGrid SMTP
SMTP_HOST=smtp.sendgrid.net
SMTP_PORT=587
SMTP_SECURE=false
SMTP_USER=apikey
SMTP_PASSWORD=<SG-api-key>
```

---

**Last Updated:** January 6, 2026

**System Version:** CareLinkAI v1.0

**Email System:** Nodemailer with SMTP transport