📧 Campaign Workflow & Email Integration Guide

🎯 Current Setup Status

☑ **Your email is already integrated!**

- Email: elaziziabdeldjalil@gmail.com
- Status: Working and ready to send campaigns
- Integration: Automatic - all campaigns use this email

🚀 How to Create & Send Email Campaigns

Method 1: Using Admin Panel (Easiest)

Step 1: Create Email Campaign

1. **Go to Admin Panel**

   ```
   http://127.0.0.1:8000/admin/
   ```

2. **Login** with your admin credentials

3. **Navigate to Email Campaigns**

   - Click "Email campaigns" in the sidebar
   - Click "Add Email Campaign" button (top right)

4. **Fill Campaign Details**

   ```
   Name: "January Newsletter"
   Subject: "Exciting Updates from MakePlus!"

   Body HTML: (Use HTML editor)
   <html>
     <body>
       <h1>Hello!</h1>
       <p>Welcome to our newsletter.</p>
       <p><a href="https://example.com">Visit our website</a></p>
     </body>
   </html>

   From Email: elaziziabdeldjalil@gmail.com (auto-filled)
   From Name: MakePlus Team

   Track Opens: ✓ (checked)
   Track Clicks: ✓ (checked)
   ```

```
Status: Draft
```

5. **Save the Campaign**

Step 2: Add Recipients

1. **Go to "Email recipients"** in admin
2. **Click "Add Email Recipient"**
3. **Fill Details**

```
Campaign: Select "January Newsletter" (your campaign)
Email: customer@example.com
Name: Customer Name
```

4. **Save** - Tracking token is auto-generated
5. **Repeat** for all recipients

Or Add Multiple Recipients via Python:

```python
from dashboard.models_email import EmailCampaign, EmailRecipient

campaign = EmailCampaign.objects.get(name="January Newsletter")

# Add multiple recipients
recipients = [
    {"email": "user1@example.com", "name": "User One"},
    {"email": "user2@example.com", "name": "User Two"},
    {"email": "user3@example.com", "name": "User Three"},
]

for recipient_data in recipients:
    EmailRecipient.objects.create(
        campaign=campaign,
        email=recipient_data["email"],
        name=recipient_data["name"]
    )

print(f"Added {len(recipients)} recipients to campaign")
```

Step 3: Send Campaign

Option A: Send via Django Shell

```
cd E:\makeplus\makeplus_backend\makeplus_api
..\venv\Scripts\python.exe manage.py shell
```

```
from dashboard.utils_campaign import send_campaign

# Send to all recipients in campaign
result = send_campaign(campaign_id="your-campaign-id-here")

print(f"✅ Sent: {result['sent']}")
print(f"❌ Failed: {result['failed']}")
```

Option B: Send to Single Recipient

```
from dashboard.utils_campaign import send_campaign_email
from dashboard.models_email import EmailRecipient

# Get specific recipient
recipient = EmailRecipient.objects.get(email="user@example.com")

# Send email
send_campaign_email(recipient)

print(f"✅ Email sent to {recipient.email}")
```

Option C: Create a Send Button (Future Enhancement)
You can add a "Send Campaign" button in the admin panel for easier sending.

Method 2: Using Dashboard Interface

1. **Go to Email Templates Page**

   ```
   http://127.0.0.1:8000/dashboard/email-templates/
   ```

2. **Create Email Template**

   - Click "Create New Template"
   - Fill in template details
   - Save template

3. **Create Campaign from Template**

   - Go to Admin Panel
   - Create EmailCampaign

    - Reference your template
    - Add recipients
    - Send via Django shell

📊 Tracking & Statistics

Automatic Tracking (Already Integrated!)

When you send a campaign, tracking happens **automatically**:

1. **Email Opens** - Tracked via invisible 1x1 pixel

   ```
   Tracking URL: http://127.0.0.1:8000/track/email/open/TOKEN/
   ```

2. **Link Clicks** - All links are automatically wrapped with tracking

   ```
   Original: https://example.com
   Tracked:
   http://127.0.0.1:8000/track/email/click/LINK_TOKEN/RECIPIENT_TOKEN/
   ```

3. **Recipient Status** - Auto-updated

    - `sent` → Email successfully sent
    - `delivered` → Email delivered (based on no bounce)
    - `bounced` → Email bounced
    - `unsubscribed` → User unsubscribed

View Statistics

Option 1: Admin Panel

```
http://127.0.0.1:8000/admin/
- Email campaigns → See open rate, click rate
- Email recipients → See individual opens/clicks
- Email opens → See all open events
- Email clicks → See all click events
```

Option 2: Dashboard Stats (Detailed)

```
http://127.0.0.1:8000/dashboard/campaigns/
- See all campaigns with stats
- Click campaign → View detailed analytics
- 4 tabs: All Recipients | Who Opened | Who Clicked | Not Opened
```

Option 3: Individual Recipient Detail

```
http://127.0.0.1:8000/dashboard/campaigns/<campaign_id>/recipients/<recipient_i
d>/
- See which links they clicked
- How many times they clicked each link
- Complete timeline of opens and clicks
```

🔄 Complete Workflow Example

Real-World Example: Send Newsletter to 100 Customers

```
# 1. Create Campaign
from dashboard.models_email import EmailCampaign, EmailRecipient
from dashboard.utils_campaign import send_campaign

campaign = EmailCampaign.objects.create(
    name="Monthly Newsletter - January 2026",
    subject="🎉 New Features Released!",
    body_html="""
    <html>
      <body style="font-family: Arial, sans-serif; padding: 20px;">
        <h1>Hello {{name}}!</h1>
        <p>We're excited to announce new features:</p>
        <ul>
          <li>Feature 1</li>
          <li>Feature 2</li>
          <li>Feature 3</li>
        </ul>
        <p><a href="https://makeplus.com/features">Learn More</a></p>
        <p><a href="https://makeplus.com/pricing">View Pricing</a></p>
        <hr>
        <small><a href="{{unsubscribe_url}}">Unsubscribe</a></small>
      </body>
    </html>
    """,
    from_email="elaziziabdeldjalil@gmail.com",
    from_name="MakePlus Team",
    track_opens=True,
    track_clicks=True,
    status='draft'
)

# 2. Add Recipients (example: from your database)
customers = [
    {"email": "customer1@example.com", "name": "John Doe"},
    {"email": "customer2@example.com", "name": "Jane Smith"},
    # ... 98 more customers
]

for customer in customers:
    EmailRecipient.objects.create(
        campaign=campaign,
        email=customer["email"],
        name=customer["name"]
    )

print(f"✅ Campaign created with {campaign.recipients.count()} recipients")

# 3. Send Campaign
result = send_campaign(campaign_id=campaign.id)

print(f"""
Campaign Sent!
✅ Sent: {result['sent']}
❌ Failed: {result['failed']}

View stats at:
http://127.0.0.1:8000/dashboard/campaigns/{campaign.id}/stats/
""")

# 4. Check Results Later
campaign.refresh_from_db()
print(f"""
Statistics:
- Total Sent: {campaign.total_sent}
- Total Opened: {campaign.total_opened} ({campaign.get_open_rate()}%)
- Total Clicked: {campaign.total_clicked} ({campaign.get_click_rate()}%)
- Unique Opens: {campaign.unique_opens}
- Unique Clicks: {campaign.unique_clicks}
""")
```

🗑 How to Remove Email Before Deployment

Step 1: Revoke Gmail App Password

1. **Go to Google Account Security**

   ```
   https://myaccount.google.com/security
   ```

2. **Click "App passwords"**

3. **Find "MakePlus Test"** in the list

4. **Click Remove** (X button)

5. **Confirm removal**

☑ **Your Gmail account is now disconnected** - Django can no longer access it

Step 2: Remove from .env File

Option A: Comment Out (Recommended for keeping reference)

Edit `E:\makeplus\makeplus_backend\makeplus_api\.env`:

```
# Gmail Configuration - REMOVED BEFORE DEPLOYMENT
# EMAIL_HOST=smtp.gmail.com
# EMAIL_PORT=587
# EMAIL_USE_TLS=True
# EMAIL_HOST_USER=elaziziabdeldjalil@gmail.com
# EMAIL_HOST_PASSWORD=cjrpnhanwilgruln
# DEFAULT_FROM_EMAIL=elaziziabdeldjalil@gmail.com

# Use console backend for development (prints to terminal)
EMAIL_BACKEND=django.core.mail.backends.console.EmailBackend
SITE_URL=http://127.0.0.1:8000
```

Option B: Delete Completely

Remove these lines from `.env`:

```
EMAIL_HOST=smtp.gmail.com
EMAIL_PORT=587
EMAIL_USE_TLS=True
EMAIL_HOST_USER=elaziziabdeldjalil@gmail.com
EMAIL_HOST_PASSWORD=cjrpnhanwilgruln
DEFAULT_FROM_EMAIL=elaziziabdeldjalil@gmail.com
```

Step 3: Switch to Console Backend (Testing Without Sending)

Add to `.env`:

```
EMAIL_BACKEND=django.core.mail.backends.console.EmailBackend
```

Now emails will **print to terminal** instead of actually sending. Great for testing!

Step 4: Clear Test Data (Optional)

```python
# In Django shell
from dashboard.models_email import EmailCampaign
```

```
# Delete test campaigns
EmailCampaign.objects.filter(name__icontains="test").delete()

# Or delete all campaigns
# EmailCampaign.objects.all().delete()
```

Step 5: Verify Removal

```
cd E:\makeplus\makeplus_backend
.\venv\Scripts\python.exe test_email_setup.py
```

Should show: **"EMAIL_HOST_USER is not set"** ☑ (This means it's removed)

🌐 Before Production Deployment

Switch to Professional Email Service

Recommended: Brevo (Sendinblue)

- 300 emails/day FREE forever
- Professional deliverability
- Better inbox rates

Setup for Production:

1. **Sign up at Brevo**

   ```
   https://www.brevo.com/
   ```

2. **Get SMTP credentials**
   - Dashboard → Settings → SMTP & API
   - Generate SMTP key

3. **Update .env for production**

   ```
   EMAIL_HOST=smtp-relay.brevo.com
   EMAIL_PORT=587
   EMAIL_USE_TLS=True
   EMAIL_HOST_USER=your-production-email@example.com
   EMAIL_HOST_PASSWORD=your-brevo-smtp-key
   DEFAULT_FROM_EMAIL=noreply@makeplus.com
   SITE_URL=https://yourdomain.com
   ```

4. **Deploy with new credentials**

See EMAIL_SETUP_GUIDE.md for full production setup

🔐 Security Best Practices

For Development (Now)

- ☑ Using App Password (not real Gmail password)
- ☑ Only on local machine
- ☑ .env file in .gitignore
- ☑ Easy to revoke

For Production (Later)

- ☑ Use professional email service (Brevo/SendGrid)
- ☑ Use environment variables (not .env file)
- ☑ Verify sender domain (SPF, DKIM)
- ☑ Monitor bounce rates
- ☑ Include unsubscribe links
- ☑ Comply with GDPR/CAN-SPAM

📋 Quick Commands Reference

Start Server

```
cd E:\makeplus\makeplus_backend\makeplus_api
..\venv\Scripts\python.exe manage.py runserver
```

Django Shell

```
cd E:\makeplus\makeplus_backend\makeplus_api
..\venv\Scripts\python.exe manage.py shell
```

Send Campaign

```python
from dashboard.utils_campaign import send_campaign
result = send_campaign(campaign_id="your-campaign-id")
```

Check Statistics

```python
from dashboard.models_email import EmailCampaign

campaign = EmailCampaign.objects.get(id="campaign-id")
print(f"Open Rate: {campaign.get_open_rate()}%")
print(f"Click Rate: {campaign.get_click_rate()}%")
```

View All Campaigns

```
http://127.0.0.1:8000/admin/dashboard/emailcampaign/
```

View Campaign Stats

```
http://127.0.0.1:8000/dashboard/campaigns/<campaign_id>/stats/
```

✅ Summary

Current State

- ✅ Email integrated and working
- ✅ Gmail App Password configured
- ✅ Automatic tracking enabled
- ✅ Statistics dashboard ready
- ✅ Can send campaigns now

To Use Now

1. Create campaign in admin panel
2. Add recipients
3. Send via Django shell: `send_campaign(campaign_id=...)`
4. View stats in dashboard
5. Track opens/clicks automatically

Before Deployment

1. Revoke Gmail App Password
2. Remove credentials from .env
3. Sign up for Brevo/SendGrid
4. Add production SMTP credentials
5. Update SITE_URL to your domain
6. Test with production service
7. Deploy!

🎯 Next Steps

1. **Test Now**: Create a real campaign with multiple recipients
2. **Check Stats**: View detailed analytics in dashboard
3. **Before Production**: Follow removal steps above
4. **For Production**: Set up Brevo/SendGrid

Your email system is fully integrated and ready to use! 🚀