

Welcome Email System Implementation Summary

Overview

A complete welcome email system has been implemented for Mindful Champion that automatically sends a beautiful, branded welcome email to users when they sign up for the platform.

Completed Tasks

1. Environment Variables Configuration

- Updated `.env` file with Gmail credentials
- **Location:** `/home/ubuntu/mindful_champion/nextjs_space/.env`
- **Variables added:**
 - `GMAIL_USER=deansnow59@gmail.com`
 - `GMAIL_APP_PASSWORD=[Your App Password]`

2. Package Installation

- Installed `nodemailer@^6.9.0` (compatible with next-auth)
- Installed `@types/nodemailer` for TypeScript support
- Installed `dotenv` for environment variable loading

3. Email Service Creation

- **Created:** `lib/email.ts`
- **Features:**
 - Nodemailer transporter configuration with Gmail SMTP
 - `sendWelcomeEmail()` function for sending welcome emails
 - `sendTestWelcomeEmail()` function for testing
 - Automatic email verification on startup
 - Error handling and logging
 - Success/failure tracking

4. Welcome Email Template

- **Beautiful HTML template** with:
 -  Mindful Champion branding (emerald/teal gradient colors)
 -  Personalized greeting with user's first name
 -  Engaging, professional yet fun tone
 -  Quick start guide (4-step onboarding)
 -  Feature highlights with icons:
 -  AI-Powered Coaching
 -  Training Plans & Analytics
 -  Pro Avatar Feature (coming soon)

- ⏳ 7-Day Free Trial
- ✅ “Go to Dashboard” CTA button
- ✅ Support contact info (deansnow59@gmail.com)
- ✅ Mobile-responsive design
- ✅ Plain text fallback for email clients that don’t support HTML

5. Database Schema Updates

- **Updated:** prisma/schema.prisma
- **Added fields to User model:**
 - welcomeEmailSent: Boolean @default(false)
 - welcomeEmailSentAt: DateTime?
- **Database synced** with `npx prisma db push`

6. Signup API Integration

- **Updated:** app/api/signup/route.ts
- **Integration features:**
 - Automatically sends welcome email after user creation
 - Updates `welcomeEmailSent` and `welcomeEmailSentAt` fields
 - Graceful error handling (doesn’t block signup if email fails)
 - Comprehensive logging for debugging
 - Email success/failure tracking

7. Test Script Created

- **Created:** test-welcome-email.ts
- **Features:**
 - Environment variable verification
 - Test email sending to deansnow59@gmail.com
 - Detailed debugging output
 - Success/failure reporting



IMPORTANT: Gmail App Password Issue

Current Status: Email Sending Blocked

The welcome email system is **fully implemented** but cannot send emails due to an **invalid Gmail App Password**.

Error: Missing credentials for "PLAIN" (Authentication failure)

How to Fix This:

Step 1: Verify 2-Factor Authentication is Enabled

1. Go to <https://myaccount.google.com/security>
2. Scroll to “How you sign in to Google”
3. Ensure “2-Step Verification” is **ON**
4. If not enabled, enable it now

Step 2: Generate a New Gmail App Password

1. Visit: <https://myaccount.google.com/apppasswords>
2. You may need to sign in again
3. Click “Select app” → Choose “Mail”
4. Click “Select device” → Choose “Other (Custom name)”
5. Enter: **“Mindful Champion”**
6. Click **“Generate”**
7. Google will display a **16-character password** (e.g., `abcd efgh ijkl mnop`)
8. **Copy this password** (without the spaces)

Step 3: Update .env File

Replace the current `GMAIL_APP_PASSWORD` value with your new 16-character App Password (remove spaces):

```
GMAIL_APP_PASSWORD=abcdefhijklmnop
```

Example:

If Google shows: `abcd efgh ijkl mnop`

Enter in .env: `GMAIL_APP_PASSWORD=abcdefhijklmnop`

Step 4: Restart the Application

```
cd /home/ubuntu/mindful_champion/nextjs_space
# Stop the current server (Ctrl+C if running)
npm run dev
```

Step 5: Test the Email

```
cd /home/ubuntu/mindful_champion/nextjs_space
npx tsx test-welcome-email.ts
```

You should see:

```
 TEST PASSED!
 Welcome email successfully sent to deansnow59@gmail.com
```

Testing Instructions

Option 1: Run Test Script

```
cd /home/ubuntu/mindful_champion/nextjs_space
npx tsx test-welcome-email.ts
```

Option 2: Create a Test User via Signup

1. Go to signup page: <https://mindfulchampionpro.abacusai.app/signup>
2. Create a new test account

3. Check email inbox for welcome email
4. Verify email formatting and content

What to Verify:

- Email arrives in inbox (check spam if not)
 - Subject line: “ Welcome to Mindful Champion - Your Journey Begins!”
 - Sender: “Mindful Champion  deansnow59@gmail.com”
 - HTML renders correctly with branding
 - All links work (especially “Go to Dashboard”)
 - Images and icons display properly
 - Mobile responsive design
-

Files Created/Modified

Created Files:

1. `lib/email.ts` - Email service with nodemailer configuration
2. `test-welcome-email.ts` - Test script for email sending
3. `WELCOME_EMAIL_IMPLEMENTATION.md` - This documentation

Modified Files:

1. `.env` - Added Gmail credentials
 2. `prisma/schema.prisma` - Added email tracking fields
 3. `app/api/signup/route.ts` - Integrated welcome email sending
 4. `package.json` - Added nodemailer and dotenv dependencies
-

Email Template Details

Design Elements:

- **Colors:** Emerald/teal gradient (#10b981 → #059669)
- **Tone:** Professional but fun, encouraging and motivating
- **Personalization:** Uses user's first name throughout
- **Call-to-Action:** Green gradient button linking to dashboard
- **Icons:** Emojis used tastefully for visual appeal

Content Sections:

1. **Header:** Gradient background with Mindful Champion logo text
 2. **Welcome Message:** Personalized greeting with trial info
 3. **CTA Button:** “Go to Dashboard” - primary action
 4. **Quick Start Guide:** 4-step onboarding checklist
 5. **Feature Highlights:** 4 key features with icons and descriptions
 6. **Support Section:** Help contact info with email link
 7. **Footer:** Copyright and tagline
-

Security Considerations

- Gmail App Password stored in `.env` (not in code)
- `.env` should be in `.gitignore` (verify!)
- Credentials never exposed in logs (except last 4 chars)
- Error handling prevents credential leakage
- Email sending failure doesn't block user signup

Email Tracking

The system tracks email delivery in the database:

```
// User model fields
welcomeEmailSent: Boolean      // true if email sent successfully
welcomeEmailSentAt: DateTime    // timestamp of email send
```

Query users who received welcome email:

```
SELECT email, firstName, lastName, welcomeEmailSentAt
FROM User
WHERE welcomeEmailSent = true
ORDER BY welcomeEmailSentAt DESC;
```

Troubleshooting

Email Not Sending

Error: “Missing credentials for PLAIN”

- Ensure 2FA is enabled on Gmail account
- Generate new App Password (16 characters)
- Update `.env` file with correct password
- Remove spaces from App Password
- Restart the application

Error: “Invalid login”

- Verify Gmail email address is correct
- Verify App Password is correct (not regular password)
- Check if App Password was revoked
- Try generating a new App Password

Email Goes to Spam

- Ask recipient to mark as “Not Spam”
- Add `deansnow59@gmail.com` to contacts
- This improves delivery reputation over time

SMTP Connection Timeout

- Check internet connectivity
- Verify firewall allows port 587
- Try port 465 with `secure: true`

Database Issues

Prisma Client Out of Sync

```
npx prisma generate
npx prisma db push
```

Next Steps

Immediate Actions:

1.  **Fix Gmail App Password** (see instructions above)
2.  **Test email delivery** with test script
3.  **Verify email in inbox** (check all sections render correctly)

Optional Enhancements:

-  **Email Templates System:** Create reusable email templates
-  **Email Analytics:** Track open rates and click-through rates
-  **Additional Emails:**
 - Trial expiration reminder (Day 10 of 14)
 - Subscription confirmation
 - Payment receipts
 - Feature update announcements
-  **Email Builder:** Admin UI to edit email templates
-  **Internationalization:** Multi-language email support
-  **Rich Notifications:** In-app notifications alongside emails



Usage Examples

Sending Welcome Email Manually

```
import { sendWelcomeEmail } from '@lib/email';

const result = await sendWelcomeEmail({
  to: 'user@example.com',
  name: 'John Doe',
  firstName: 'John',
});

if (result.success) {
  console.log('Email sent:', result.messageId);
} else {
  console.error('Email failed:', result.error);
}
```

Checking Email Status in Database

```
const user = await prisma.user.findUnique({
  where: { email: 'user@example.com' },
  select: { welcomeEmailSent: true, welcomeEmailSentAt: true },
});

console.log('Email sent?', user.welcomeEmailSent);
console.log('Sent at:', user.welcomeEmailSentAt);
```

📞 Support

If you encounter any issues or need assistance:

- **Email:** deansnow59@gmail.com
- **Documentation:** This file (WELCOME_EMAIL_IMPLEMENTATION.md)
- **Test Script:** npx tsx test-welcome-email.ts
- **Logs:** Check server console for email sending logs

✨ Summary

The welcome email system is **fully implemented and ready to use**. The only remaining step is to **configure a valid Gmail App Password**. Once that's done, new users will automatically receive beautiful, personalized welcome emails when they sign up for Mindful Champion!

Total Implementation Time: ~45 minutes

Status: 🚨 95% Complete (waiting on valid Gmail App Password)

Next Action: Generate Gmail App Password and update .env file