Pull Request: Email Management System - Phase 1

📧 Overview

This PR implements **Phase 1** of the Email Management System for Writgo, providing the foundation for email handling with IMAP integration, secure configuration management, and a complete admin interface for viewing and managing emails.

✨ Features Implemented

🔐 Security & Encryption

- **AES-256-GCM Encryption** for password storage
- Environment-based encryption key (`EMAIL_ENCRYPTION_KEY`)
- Masked passwords in API responses
- Secure credential storage in database

📬 IMAP Integration

- Full IMAP client implementation using `imapflow`
- Email parsing with `mailparser` (headers, body, attachments)
- Real-time email fetching from IMAP server
- Connection testing functionality
- Read/unread status management

🎨 Admin Interface

Three new admin pages with dark theme design:

1. **Email Instellingen** (`/admin/email/instellingen`)
   - Email account configuration form
   - IMAP/SMTP settings
   - Test connection functionality
   - Password encryption on save
   - Help section for common providers (Gmail, Outlook)

2. **Email Inbox** (`/admin/email/inbox`)
   - List of emails (last 50)
   - Read/unread indicators
   - Attachment badges
   - Refresh functionality
   - Click to open email

3. **Email Detail** (`/admin/email/inbox/[uid]`)
   - Full email view with headers
   - HTML/Text body toggle
   - Attachments list

- Auto-mark as read
- Back to inbox navigation

🔌 API Routes

Four new API endpoints:

1. **POST/GET/DELETE** `/api/admin/email/accounts` - Account management
2. **POST** `/api/admin/email/test-connection` - Test IMAP connection
3. **GET** `/api/admin/email/inbox` - Fetch emails via IMAP
4. **GET/POST** `/api/admin/email/message` - Single email operations

📦 New Files

Core Libraries

- `nextjs_space/lib/encryption.ts` - AES-256-GCM encryption utilities
- `nextjs_space/lib/email/imap-client.ts` - IMAP client with full functionality

API Routes

- `nextjs_space/app/api/admin/email/accounts/route.ts`
- `nextjs_space/app/api/admin/email/test-connection/route.ts`
- `nextjs_space/app/api/admin/email/inbox/route.ts`
- `nextjs_space/app/api/admin/email/message/route.ts`

Admin Pages

- `nextjs_space/app/admin/email/instellingen/page.tsx`
- `nextjs_space/app/admin/email/inbox/page.tsx`
- `nextjs_space/app/admin/email/inbox/[uid]/page.tsx`

Documentation

- `EMAIL_SYSTEM_PHASE1.md` - Complete implementation documentation
- `EMAIL_SYSTEM_PHASE1.pdf` - PDF version

🔧 Modified Files

- `nextjs_space/lib/admin-navigation-config.ts` - Added Email section with Inbox and Instellingen items
- `nextjs_space/package.json` - Added `imapflow` and `mailparser` dependencies
- `nextjs_space/package-lock.json` - Dependency updates
- `nextjs_space/yarn.lock` - Dependency updates

🗄️ Database

Uses existing tables from migration `20251210_email_inbox_tables.sql`:
- **MailboxConnection** - Email account configuration (IMAP/SMTP settings, encrypted passwords)
- **InboxEmail** - Email storage (prepared for Phase 2)

No new migrations required - schema already exists.

📋 Setup Instructions

1. Generate Encryption Key

```
node -e "console.log(require('crypto').randomBytes(32).toString('hex'))"
```

2. Add Environment Variable

Add to `.env.local` or Render environment variables:

```
EMAIL_ENCRYPTION_KEY=your-64-character-hex-key-here
```

3. Install Dependencies

```
cd nextjs_space
npm install --legacy-peer-deps
```

4. For Gmail Setup

1. Enable 2-Step Verification in Google Account
2. Generate App Password (Security → App Passwords → Mail)
3. Use settings:
   - IMAP: `imap.gmail.com:993`
   - SMTP: `smtp.gmail.com:587`
   - Use app password (not regular password)

🧪 Testing Checklist

- [x] TypeScript compilation successful
- [ ] Email account configuration saves correctly
- [ ] Test connection works for Gmail/Outlook
- [ ] Inbox loads emails from IMAP
- [ ] Email detail page shows full content
- [ ] Passwords are encrypted in database
- [ ] HTML and text email bodies render correctly
- [ ] Attachments metadata is displayed
- [ ] Navigation works between pages

🎯 Known Limitations (Phase 1)

1. **No Email Sending** - SMTP configured but not implemented yet
2. **No Email Persistence** - Real-time IMAP fetch only (not stored in database)
3. **No AI Analysis** - Database fields ready, functionality in Phase 2
4. **No Attachment Downloads** - Metadata only
5. **No Thread Grouping** - Individual emails only
6. **No Search/Filtering** - Basic inbox view only
7. **Single Account** - Uses first active account only

🚀 Roadmap

Phase 2: Email Persistence & AI Analysis

- Store emails in InboxEmail table
- AI categorization, priority, sentiment analysis
- Invoice detection
- Background sync job
- Search & filtering

Phase 3: Email Sending & Replies

- SMTP client implementation
- Email composer UI
- Reply/Forward functionality
- Email templates

Phase 4: Advanced Features

- Thread grouping
- Auto-reply configuration
- Multiple accounts
- Email automation rules

📊 Impact

- **Lines Added:** ~2,935
- **Files Changed:** 15
- **New Dependencies:** 2 (imapflow, mailparser)
- **New API Routes:** 4
- **New Pages:** 3

🔒 Security Considerations

✅ Passwords encrypted with AES-256-GCM before database storage
✅ Encryption key stored in environment variables
✅ Passwords masked in API responses
✅ IMAP connections use TLS
✅ Session-based authentication for API routes

📚 Documentation

Complete documentation available in `EMAIL_SYSTEM_PHASE1.md`:
- Architecture overview
- API documentation
- Setup guide
- Usage instructions
- Troubleshooting
- Security implementation details

🤝 Review Checklist

- [ ] Code review completed
- [ ] Security review passed
- [ ] Documentation reviewed
- [ ] Environment variables documented
- [ ] Testing checklist completed
- [ ] Ready for deployment

🎉 Result

After merging this PR, admin users will be able to:
1. Configure their email account (IMAP/SMTP) in the admin panel
2. View their inbox with real-time IMAP fetching
3. Read emails with full HTML/text content
4. See attachment metadata
5. Test email connections before saving

All with proper encryption, error handling, and a clean dark-themed UI! 🚀