

Phase 1 Implementation Plan - The Mindful Musicpreneur App

Current Status (as of completion)

✓ COMPLETED Components (35-40%)

1. Foundation

- NextJS 14 project setup with TypeScript
- Prisma database schema with all models
- Tailwind CSS with brand colors and fonts
- NextAuth authentication setup
- Stripe and Resend integration configured

2. Core Pages

- Homepage with hero section, product grid, CTAs ✓
- Product pages (Guide, Planner, BOGO) with complete copy ✓
- The Collective public page with full description ✓
- Freebie opt-in page ✓
- Auth pages (signin, signup, error) ✓

3. Components & Layout

- Header with navigation ✓
- Footer with links ✓
- UI component library (shadcn/ui) ✓
- Brand styling throughout ✓

4. Email Templates

- All 6 email templates created in /lib/email.ts ✓
- Resend integration configured ✓

🚧 IN PROGRESS / TO COMPLETE (60-65%)

CRITICAL PATH (Must Build)

1. Checkout Flow (Priority: CRITICAL)

- [] /app/checkout/page.tsx - Cart review, payment selection
- [] /app/api/checkout/route.ts - Create Stripe session
- [] /app/checkout/success/page.tsx - Order confirmation
- [] /app/api/webhooks/stripe/route.ts - Handle payments, send emails

2. User Dashboard (Priority: CRITICAL)

- [] /app/dashboard/page.tsx - Show purchases, downloads, profile
- [] /app/api/downloads/[productId]/route.ts - Secure PDF downloads

3. The Collective Application & Portal (Priority: HIGH)

- [] /app/collective/apply/page.tsx - Application form
- [] /app/collective/portal/page.tsx - Member-only portal
- [] /app/api/collective/apply/route.ts - Save applications

4. Admin Dashboard (Priority: HIGH)

- [] /app/admin/page.tsx - Dashboard home with stats
- [] /app/admin/orders/page.tsx - View/manage orders
- [] /app/admin/users/page.tsx - Manage users
- [] /app/admin/collective/applications/page.tsx - Review applications
- [] /app/admin/collective/members/page.tsx - Manage members
- [] /app/admin/content/page.tsx - Upload PDFs, manage portal
- [] /app/admin/emails/page.tsx - Email list management
- [] /app/admin/analytics/page.tsx - Sales & user stats
- [] Admin API routes for all actions

5. Additional Pages (Priority: MEDIUM)

- [] /app/terms/page.tsx - Terms of Service
- [] /app/privacy/page.tsx - Privacy Policy
- [] /app/unsubscribe/page.tsx - Email preferences

6. Testing & Deployment (Priority: CRITICAL)

- [] Create seed script with test products and user
- [] Test complete user flows
- [] Test BOGO workflow
- [] Test admin workflows
- [] Verify mobile responsiveness
- [] Add error handling throughout
- [] Update README with deployment instructions

File Structure Overview

```

app/
├── page.tsx [✓] (Homepage)
├── layout.tsx [✓] (Root layout)
└── products/
    ├── guide/page.tsx [✓]
    ├── planner/page.tsx [✓]
    └── bogo/page.tsx [✓]
└── collective/
    ├── page.tsx [✓] (Public page)
    ├── apply/page.tsx [✗] (Application form)
    └── portal/page.tsx [✗] (Member portal)
└── checkout/
    ├── page.tsx [✗] (Checkout form)
    └── success/page.tsx [✗] (Confirmation)
└── dashboard/
    ├── page.tsx [✗] (User dashboard)
└── admin/
    ├── page.tsx [✗] (Admin home)
    ├── orders/page.tsx [✗]
    ├── users/page.tsx [✗]
    ├── collective/
    │   ├── applications/page.tsx [✗]
    │   ├── members/page.tsx [✗]
    │   ├── content/page.tsx [✗]
    │   ├── emails/page.tsx [✗]
    │   └── analytics/page.tsx [✗]
    ├── auth/
    │   ├── signin/page.tsx [✓]
    │   ├── signup/page.tsx [✓]
    │   └── error/page.tsx [✓]
    ├── freebie/page.tsx [✓]
    ├── terms/page.tsx [✗]
    ├── privacy/page.tsx [✗]
    └── unsubscribe/page.tsx [✗]
└── api/
    ├── auth/
    │   ├── [...nextauth]/route.ts [✓]
    │   ├── signup/route.ts [✓]
    │   └── checkout/route.ts [✗]
    ├── webhooks/
    │   ├── stripe/route.ts [✗]
    ├── downloads/
    │   ├── [productId]/route.ts [✗]
    ├── collective/
    │   ├── apply/route.ts [✗]
    └── admin/
        ├── users/route.ts [✗]
        ├── applications/
        │   ├── [id]/approve/route.ts [✗]
        │   ├── [id]/deny/route.ts [✗]
        │   ├── content/route.ts [✗]
        │   └── emails/route.ts [✗]
        └── freebie/route.ts [✓]

```

Key Workflows to Implement

1. Product Purchase Flow

```
User → Product Page → Checkout → Stripe Payment → Webhook →
→ Create Order → Grant Access → Send Email → Redirect to Success Page
```

2. BOGO Purchase Flow

```
Buyer → BOGO Page → Checkout (with recipient info) → Stripe Payment →
→ Webhook → Create 2 user accounts → Grant both access →
→ Send 2 emails (buyer + recipient)
```

3. Collective Application Flow

```
User → Buy Guide → Dashboard → Apply to Collective →
→ Admin Reviews → Approve/Deny → Send Email → Grant/Deny Portal Access
```

4. Download Flow

```
Authenticated User → Dashboard → Click Download →
→ API checks entitlement → Returns secure signed URL → Download starts
```

Database Seeds Needed

Create in `/prisma/seed.ts` :

- Products (Guide, Planner, BOGO, Freebie, Collective Monthly/Yearly)
- Test admin user
- Test regular user
- Sample portal content (Zoom URL, welcome message)

Environment Variables Required

```
# Database
DATABASE_URL="postgresql://..."

# NextAuth
NEXTAUTH_SECRET="..."
NEXTAUTH_URL="http://localhost:3000"

# Stripe
NEXT_PUBLIC_STRIPE_PUBLISHABLE_KEY="pk_test_..."
STRIPE_SECRET_KEY="sk_test_..."
STRIPE_WEBHOOK_SECRET="whsec_..."

# Resend
RESEND_API_KEY="re_..."

# App
NEXT_PUBLIC_APP_URL="http://localhost:3000"
ADMIN_EMAIL="admin@example.com"
```

Next Steps

1. **Create remaining pages** (dashboard, checkout, admin)
2. **Build all API routes** for workflows
3. **Create seed script** with test data
4. **Test all user flows** end-to-end
5. **Deploy to production** environment
6. **Update README** with instructions

Estimated Completion

- Remaining pages: ~15-20 files
- API routes: ~10-12 files
- Testing & fixes: ~4-6 hours
- Documentation: ~1-2 hours

Total estimated time to complete Phase 1: 8-12 hours of focused development

Notes:

- All brand assets are in `/public` folder
- Email templates are in `/lib/email.ts`
- Product data is in `/lib/products.ts`
- Database schema is complete in `/prisma/schema.prisma`