**Revised MINGUS Integration Prompts Based on Current Status**

**🎯 Key Changes Based on Your Application Status**

Your MINGUS application already has:

* ✅ **Complete user profile system** (6 tables, 25+ fields)
* ✅ **Working authentication** (just needs production fixes)
* ✅ **Subscription management** (3 tiers: $10, $20, $50)
* ✅ **React frontend** with profile management
* ✅ **6 API endpoints** for user operations
* ✅ **SQLite database** with comprehensive schema

**Therefore, the prompts focus on INTEGRATION rather than creation from scratch.**

**Phase 1: Assessment System Integration (Not Creation)**

**Prompt 1: Add Assessment Tables to Existing MINGUS Database**

Add lead magnet assessment functionality to the existing MINGUS SQLite database that already has 6 tables (users, user\_profiles, onboarding\_progress, subscription\_plans, subscriptions, feature\_usage).

Current Database Status:

- ✅ users table with comprehensive profile (25+ fields including first\_name, last\_name, zip\_code, industry, job\_title, monthly\_income, etc.)

- ✅ user\_profiles table with detailed information

- ✅ subscription\_plans table with 3 tiers ($10, $20, $50)

- ✅ Working authentication system

NEW Tables to Add:

1. assessments (id, type, title, questions\_json, scoring\_config, active, created\_at)

2. user\_assessments (id, user\_id, assessment\_id, responses\_json, score, segment, completed\_at)

3. assessment\_results (id, user\_assessment\_id, insights\_json, cost\_projections, immediate\_action, created\_at)

Integration Requirements:

- Use existing user.id as foreign key (don't create new user system)

- Follow existing table naming conventions from your 6-table structure

- Add migration script that works with your existing apply\_user\_profile\_migrations.py

- Ensure compatibility with existing API endpoints (/api/user-profile/\*)

Database Migration Script:

Create assessment\_migration.py that adds these 3 tables to your existing mingus.db without affecting the working user profile system.

Seed Data:

- Create 3 assessment records (relationship, career, tax) with the 11-question structure from your documents

- Add sample assessment responses that integrate with your existing test user (test@mingus.com)

**Prompt 2: Extend Existing API Endpoints for Assessments**

Extend the existing MINGUS API system that already has 6 working endpoints:

- ✅ GET /api/user-profile/get

- ✅ POST /api/user-profile/update

- ✅ GET /api/user-profile/onboarding-progress

- ✅ POST /api/user-profile/onboarding-progress

- ✅ GET /api/user-profile/subscription

- ✅ GET /api/user-profile/feature-usage

ADD new assessment endpoints that integrate with existing patterns:

File: backend/routes/assessment\_routes.py (new file, but follow existing patterns from user\_profile.py)

New Endpoints:

1. GET /api/assessments/available

- Return 3 assessments (relationship, career, tax) with popularity stats

- Use existing user authentication patterns

- Integrate with existing subscription system to show available vs. premium assessments

2. POST /api/assessments/<type>/submit

- Accept assessment responses for anonymous or logged-in users

- If anonymous, create user record using existing user creation patterns

- Calculate score using the scoring logic from your documents ($27 conversion strategy)

- Return immediate results (free content) + conversion offer

- Integrate with existing subscription\_plans table for pricing

3. GET /api/user-profile/assessment-insights

- Add to existing user-profile routes

- Return assessment results for logged-in users

- Integrate with existing profile completion system

Integration Requirements:

- Use existing authentication decorators and error handling patterns

- Connect to existing subscription management for conversion offers

- Follow existing data validation and response formatting

- Integrate with existing user profile data for personalized insights

The scoring algorithm should implement your documented strategy:

- 11-18 points: "Balanced Builder" → Budget tier ($10)

- 19-30 points: "Relationship Spender/Income Accelerator/Family Planner" → Mid-tier ($20)

- 31+ points: "Crisis Manager" → Professional tier ($50)

**Prompt 3: Assessment Scoring Service Integration**

Create assessment scoring service that integrates with your existing MINGUS backend architecture and documented conversion strategy.

File: backend/services/assessment\_service.py

Integration with Existing System:

- Use existing user data from user\_profiles table (industry, job\_title, monthly\_income, zip\_code, etc.)

- Connect to existing subscription\_plans table for tier recommendations

- Follow existing service patterns from your backend architecture

Scoring Implementation from Your Documents:

1. Relationship Assessment (11 questions, max 44 points):

- Questions 1-7: Standard emotional/social spending patterns

- Questions 8-10: Partner dynamics (your new questions about spending imbalance, future alignment, partner stress)

- Question 11: Investment mindset

2. Career Assessment (11 questions, max 44 points):

- Use existing user profile data (industry, job\_title, monthly\_income) for salary gap calculations

- Your documented income gap strategy: show specific dollar amounts like "$12,000-$18,000 leaving on table annually"

3. Tax Assessment (11 questions, max 44 points):

- Use existing user profile data (zip\_code, dependents, marital\_status) for tax calculations

- Your documented 2025 tax policy impact strategy

Cost Projection Calculations (from your free vs paywall strategy):

- Relationship: Emotional spending $2,400/year + relationship conflicts $1,800/year + emergency unpreparedness $3,200/year

- Career: Income gap $12,000/year + skills delay $8,000/year + network limitation $5,000/year

- Tax: Tax inefficiency $1,200/year + parent care costs $2,400/year + benefit losses $1,800/year

Free Results Generation (implement your "appetizer approach"):

- Score and segment classification

- Top 3 specific pattern insights

- One immediate action item

- Peer comparison insight

- Annual cost projections (shocking numbers for conversion)

Conversion Offer Integration:

- Connect to existing subscription\_plans table

- Implement your $27 special offer (normally $47 + $20/month = $67)

- 60-minute timer countdown

- Integration with existing billing system

**Phase 2: Frontend Integration with Existing React App**

**Prompt 4: Landing Page Integration with Existing MINGUS Frontend**

Add assessment landing page to the existing MINGUS React application that already has:

- ✅ Working authentication (test@mingus.com / test123)

- ✅ Profile management interface

- ✅ Dashboard at http://localhost:5001

- ✅ Mobile-responsive design

File: frontend/src/components/assessments/AssessmentLanding.tsx

Integration Requirements:

- Use existing React patterns and component structure

- Integrate with existing authentication state management

- Follow existing routing patterns

- Use existing styling and design system

Landing Page Structure (from your strategy document):

```jsx

// Hero Section

"Which Financial Challenge Is Costing You The Most?"

"Discover your biggest money obstacle and get a personalized action plan in just 4 minutes.

Join 1,000+ African American professionals who've taken control of their financial future."

// Three-Option Grid (from your documents):

1. 💕 Love & Money Balance - "Most Popular Choice"

"Is your love life draining your bank account?"

"847 people took this today"

2. 🚀 Career Income Gap - "Highest ROI Potential"

"Is your career paying what you're worth?"

"Average user discovers $18k gap"

3. 📊 2025 Tax Impact - "Most Timely"

"Are you ready for the new tax reality?"

"Policy changes affect 89% of users"

Navigation Integration:

* Add to existing navigation menu
* Integrate with existing user authentication state
* Connect to existing dashboard for logged-in users

Analytics Integration:

* Track assessment\_card\_clicked events
* Integrate with existing user tracking patterns
* Connect to existing performance monitoring

### Prompt 5: Assessment Questionnaire Components for Existing MINGUS App

Create assessment questionnaire components that integrate with your existing MINGUS React application and implement your documented 11-question structure.

Files to Create: frontend/src/components/assessments/ ├── AssessmentFlow.tsx (main component) ├── QuestionCard.tsx (individual questions) ├── ProgressTracker.tsx (completion progress) └── AssessmentTypes.ts (TypeScript interfaces)

Integration with Existing MINGUS:

* Use existing form validation patterns from profile management
* Follow existing component structure and styling
* Integrate with existing user authentication
* Connect to existing API calling patterns

Question Implementation (your documented questions):

Relationship Assessment (11 questions):

* Q1: Financial stress frequency (1-4 points)
* Q2: Relationship spending triggers - checkboxes (0-3 points each)
* Q3: Money communication comfort - rating scales (1-5 converted to points)
* Q4: Financial planning style (0-3 points)
* Q5: Relationship vs money goals conflict (1-4 points)
* Q6: Emergency readiness (1-4 points)
* Q7: Spending motivation (1-3 points)
* Q8: Partner spending balance (your new question, 0-4 points)
* Q9: Future financial alignment (your new question, 0-4 points)
* Q10: Partner spending stress (your new question, 0-4 points)
* Q11: Tool investment mindset (0-3 points)

Similar structure for Career and Tax assessments following your documented questions.

Progress Tracking:

* Visual progress bar showing completion percentage
* Estimated time remaining (target: 4 minutes total)
* Auto-save to localStorage using existing patterns
* Integration with existing user session management

Real-time Validation:

* Use existing MINGUS form validation patterns
* Mobile-optimized touch targets (44px minimum)
* Smooth transitions between questions
* Integration with existing error handling

### Prompt 6: Results Display and Immediate Conversion Modal

Implement assessment results display and immediate conversion system based on your documented $27 conversion strategy.

Files to Create: frontend/src/components/assessments/ ├── ResultsDisplay.tsx ├── ConversionModal.tsx  
├── ScoreVisualization.tsx └── ImmediateOffer.tsx

Integration with Existing MINGUS:

* Use existing modal/overlay patterns if available
* Integrate with existing subscription management UI
* Connect to existing payment/billing interface
* Follow existing design system and styling

Results Display Implementation (your "appetizer approach"):

Free Results Section:

// Score & Segment Classification

🎯 YOUR RELATIONSHIP-MONEY BALANCE SCORE: {score}/44

You're a "{segmentName}" - Join {percentage}% of our users who share this profile.

// Top 3 Specific Pattern Insights

🔍 YOUR BIGGEST CHALLENGES (Based on your responses):

1. STRESS-SPENDING TRIGGER: You selected "stress-shopping after relationship conflicts"

- this pattern alone costs the average person $2,400 annually.

// One Immediate Action Item

🚀 DO THIS TODAY: Based on your spending triggers, implement the "24-Hour Rule"

// Peer Comparison Insight

📊 HOW YOU COMPARE: People with your score typically:

• Spend 23% more on "relationship maintenance" than financially stable couples

Immediate Conversion Modal (your $27 strategy):

// Conversion Offer (appears after 5 seconds)

🚀 SKIP THE GUESSWORK - GET THE COMPLETE SOLUTION NOW

⏰ SPECIAL ASSESSMENT TAKER OFFER (Valid for 60 minutes only):

┌─ NORMALLY $47 E-BOOK + $20/MONTH APP ─┐

│ GET BOTH FOR JUST $27 │

│ (First month of app included FREE) │

└─────────────────────────────────────────┘

[GET INSTANT ACCESS - $27 BUTTON]

Integration with Existing Subscription System:

* Connect to existing subscription\_plans table ($10, $20, $50 tiers)
* Use existing billing/payment infrastructure
* Integrate with existing user account creation
* Connect to existing dashboard for post-purchase experience

Timer Implementation:

* 60-minute countdown using existing JavaScript patterns
* Exit intent detection with $19 emergency offer
* Real-time social proof notifications ("James just purchased...")

## Phase 3: Enhanced Integration Features

### Prompt 7: Dashboard Integration with Existing User Interface

Integrate assessment insights into the existing MINGUS dashboard that already has:

* ✅ User profile management interface
* ✅ Profile completion tracking
* ✅ Subscription management
* ✅ Real-time updates

Files to Update:

* Extend existing dashboard components with assessment widgets
* Add assessment insights to existing user profile display
* Integrate with existing onboarding progress system

Dashboard Enhancements:

Assessment Insight Widgets:

// Add to existing dashboard

<AssessmentInsightWidget>

<h3>Your Financial Health Score: {score}/44</h3>

<SegmentBadge>{segmentName}</SegmentBadge>

<ActionItem>{immediateAction}</ActionItem>

<ProgressTracker>

<span>Challenges Addressed: {completedActions}/3</span>

<span>Estimated Annual Savings: ${projectedSavings}</span>

</ProgressTracker>

</AssessmentInsightWidget>

Integration with Existing Profile System:

* Use assessment data to enhance existing 25+ profile fields
* Connect assessment insights to existing goal tracking
* Integrate with existing feature usage tracking
* Enhance existing subscription recommendations

Onboarding Flow Enhancement:

* Modify existing onboarding\_progress system
* Pre-populate relevant fields from assessment responses
* Customize feature introductions based on assessment results
* Connect to existing subscription upgrade prompts

### Prompt 8: Email Integration with Existing Communication System

Integrate assessment-based email automation with existing MINGUS communication infrastructure.

Note: Based on your tech stack document, MINGUS already has Resend integration and intelligent communication routing.

Files to Update/Create: backend/services/ ├── assessment\_email\_service.py (new) └── communication\_router\_service.py (extend existing)

Integration with Existing MINGUS Communication:

* Use existing Resend email service integration
* Connect to existing intelligent message routing
* Leverage existing cultural personalization for African American professionals
* Integrate with existing communication preferences system

Email Sequence Implementation (your documented strategy):

Immediate Results Delivery:

* Generate personalized PDF with assessment insights
* Use existing email template system
* Dynamic content based on score and segment from your strategy
* Connect to existing user profile data for personalization

5-Email Nurturing Sequence (your documented approach): Day 0: Results delivery + quick wins Day 2: Educational content addressing primary challenge  
Day 4: Success stories from similar profiles (use existing testimonial system) Day 6: Feature showcase connecting to existing MINGUS features Day 7: Final conversion offer with existing subscription integration

Integration with Existing User System:

* Connect to existing user\_profiles table for personalization
* Use existing subscription status for targeted messaging
* Integrate with existing feature\_usage table for relevant recommendations
* Connect to existing notification\_preferences for delivery optimization

## Phase 4: Production Integration and Analytics

### Prompt 9: Analytics Integration with Existing Monitoring

Integrate assessment analytics with existing MINGUS monitoring and analytics infrastructure.

Based on your tech stack document, MINGUS already has comprehensive monitoring. Extend this system for assessments.

Files to Update/Create: backend/services/ ├── assessment\_analytics\_service.py (new) └── existing\_analytics\_service.py (extend)

Integration with Existing MINGUS Analytics:

* Use existing comprehensive monitoring patterns
* Connect to existing business intelligence features
* Integrate with existing user behavior tracking
* Leverage existing performance monitoring tools

Assessment-Specific Analytics (your conversion funnel):

Landing Page Analytics:

* assessment\_card\_clicked (integrate with existing user tracking)
* assessment\_started (connect to existing session management)
* question\_answered (use existing interaction tracking patterns)

Conversion Analytics:

* results\_viewed (connect to existing page view tracking)
* conversion\_offer\_shown (integrate with existing feature usage analytics)
* purchase\_initiated (connect to existing subscription analytics)
* email\_captured (integrate with existing lead tracking)

Revenue Attribution:

* Assessment type → subscription conversion tracking
* Customer lifetime value by assessment entry point
* Integration with existing subscription\_plans analytics
* Connect to existing feature\_usage tracking for retention analysis

Dashboard Integration:

* Add assessment metrics to existing admin/analytics dashboard
* Real-time assessment performance monitoring
* Integration with existing health checks and alerts
* Connect to existing user engagement metrics

### Prompt 10: Production Deployment Integration

Deploy assessment system to production as part of existing MINGUS infrastructure.

Current Production Status:

* ✅ User profile system production-ready
* ✅ Authentication system functional (minor fixes needed)
* ✅ Subscription system operational
* ⚠️ Minor authentication decorators need production configuration

Assessment Production Integration:

Database Migration:

* Add assessment tables to existing production mingus.db
* Ensure backward compatibility with existing 6-table system
* Maintain existing user data integrity
* Use existing backup and monitoring procedures

API Endpoint Deployment:

* Add assessment routes to existing production API
* Use existing authentication and security patterns
* Integrate with existing rate limiting and error handling
* Connect to existing health check system (/api/health)

Frontend Deployment:

* Add assessment components to existing React production build
* Integrate with existing mobile-responsive design
* Use existing CDN and performance optimization
* Connect to existing user authentication flow

Environment Configuration:

# Add to existing production environment variables

ASSESSMENT\_CACHE\_TTL=3600

CONVERSION\_OFFER\_DURATION=3600

ASSESSMENT\_RATE\_LIMIT=10\_per\_hour

ENABLE\_AB\_TESTING=true

Production Testing:

* Extend existing test suite (10+ scenarios) to include assessments
* Use existing API endpoint testing patterns
* Integrate with existing database verification procedures
* Connect to existing web interface testing framework

Monitoring Integration:

* Use existing comprehensive monitoring for assessment endpoints
* Connect to existing error tracking and alerting
* Integrate assessment metrics with existing performance monitoring
* Add assessment health checks to existing monitoring dashboard

---

## 🎯 \*\*Key Changes Summary\*\*

### \*\*Original Prompts → Revised Prompts:\*\*

1. \*\*Database Creation\*\* → \*\*Database Extension\*\*: Add 3 tables to existing 6-table system

2. \*\*User System Creation\*\* → \*\*User System Integration\*\*: Use existing 25+ profile fields

3. \*\*Authentication Setup\*\* → \*\*Authentication Extension\*\*: Use existing working auth system

4. \*\*Subscription Creation\*\* → \*\*Subscription Integration\*\*: Use existing 3-tier system ($10/$20/$50)

5. \*\*Frontend Creation\*\* → \*\*Component Addition\*\*: Add to existing React app

6. \*\*API Creation\*\* → \*\*API Extension\*\*: Add to existing 6 working endpoints

### \*\*Immediate Advantages:\*\*

- \*\*Faster Implementation\*\*: Building on existing working system

- \*\*Data Consistency\*\*: Using existing user profiles and subscription management

- \*\*User Experience\*\*: Seamless integration with existing dashboard and authentication

- \*\*Production Ready\*\*: Leveraging existing production infrastructure

### \*\*Implementation Timeline:\*\*

- \*\*Week 1\*\*: Database extension + API integration (Prompts 1-3)

- \*\*Week 2\*\*: Frontend component addition (Prompts 4-6)

- \*\*Week 3\*\*: Dashboard and email integration (Prompts 7-8)

- \*\*Week 4\*\*: Analytics and production deployment (Prompts 9-10)

Your existing MINGUS infrastructure significantly accelerates the lead magnet implementation since you already have the core user management, subscription, and authentication systems working.