

# VerifyLens Integration - Executive Summary

---

**Date:** October 28, 2025

**Purpose:** Quick reference guide for automated credit/billing system integration

---

## Current Situation

---

### What Works

- **Live App:** verifylens.com with robust search functionality
- **Authentication:** Secure NextAuth.js with JWT tokens
- **Search Types:** Both Exact and Smart search working perfectly
- **Admin Dashboard:** Full CRUD for customers and users
- **Database:** Well-structured PostgreSQL on Supabase
- **Search Logging:** Complete audit trail of all searches

### Critical Gaps

- **No Credit System:** Searches are free and unlimited
  - **No Payment Integration:** No Stripe or billing system
  - **Manual Account Creation:** Admin must manually create accounts after purchase
  - **No Self-Service:** Customers can't sign up or purchase on their own
  - **No Usage Enforcement:** Can't limit searches based on credits purchased
- 

## The Problem

---

### Current Manual Workflow:

Customer Contacts Sales → Quote → Payment (outside system) →  
Admin Notified → Admin Creates Account Manually →  
**Send** Credentials to Customer → Customer Logs In

**Time:** 10-30 minutes per customer

**Scalability:** Poor

**Error Rate:** High (manual credential handling)

**Customer Experience:** Slow and cumbersome

---

## The Solution

---

### Automated Workflow:

Customer Visits Landing Page → Selects Credit Package →  
 Stripe Checkout → Payment Success → Webhook Triggers →  
 Account Auto-Created → Credits Allocated →  
 Welcome Email Sent → Customer Logs In Immediately

**Time:** < 1 minute (fully automated)

**Scalability:** Unlimited

**Error Rate:** Minimal

**Customer Experience:** Fast and seamless

## Key Requirements

### 1. Database Changes

#### New Tables:

- `credit_packages` - Define credit tiers (10, 50, 100 credits)
- `customer_credits` - Track balance per customer
- `credit_transactions` - Log all credit changes (purchases, usage)
- `stripe_payments` - Store payment records

#### New Columns:

- `customers.stripe_customer_id` - Link to Stripe

### 2. Stripe Integration

- Stripe Checkout Sessions for payment
- Webhook handler for `checkout.session.completed`
- Automatic account creation on successful payment
- Credit allocation linked to payment

### 3. Credit Deduction Logic

- Check balance before search
- Deduct 1 credit per successful search
- **Special Rule:** Exact search with no results = FREE
- Show “Out of Credits” modal when balance = 0
- Low credit warning when balance < 5

### 4. Frontend Updates

- Display credit balance in header
- “Buy More Credits” button → landing page
- Credit transaction history page
- Insufficient credits modal
- Low credit warning banner

### 5. Email Service

- Account creation with credentials
- Payment receipts
- Low credit alerts

- Payment confirmations

---

## Pricing Model

---

### Recommended:

- **1 Credit = \$100 per search** (both Exact and Smart)
- **Package Tiers:**
- 10 credits = \$1,000
- 50 credits = \$5,000 (save 5%)
- 100 credits = \$10,000 (save 10%)

### Special Rules:

- Exact search with no results = FREE (don't charge)
  - Smart search always charged (returns multiple matches)
- 

## Implementation Timeline

---

### Phase 1: Database Foundation (1-2 days)

- Create new tables
- Implement credit management functions

### Phase 2: Stripe Integration (2-3 days)

- Set up Stripe account
- Implement checkout flow
- Build webhook handler

### Phase 3: Automated Account Creation (2-3 days)

- Account creation logic in webhook
- Email integration
- Welcome email template

### Phase 4: Credit Deduction (2-3 days)

- Update search APIs
- Implement balance checks
- Add transaction logging

### Phase 5: Frontend UI (3-4 days)

- Credit balance display
- Insufficient credits modal
- Transaction history page

### Phase 6: Landing Page Updates (2-3 days)

- Add "Buy Now" buttons
- Stripe checkout integration
- Success/failure pages

## Phase 7: Testing & QA (3-4 days)

- End-to-end testing
- Edge case handling
- Security validation

## Phase 8: Deployment (1-2 days)

- Production deployment
- Monitoring setup
- Documentation

**Total Estimated Time:** 16-24 days (3-4 weeks)

---

## Technical Stack

---

### Existing:

- Next.js 15.5.4 (App Router)
- NextAuth.js 4.24.11
- PostgreSQL on Supabase
- Vercel hosting

### To Add:

- Stripe (payment processing)
  - Resend (email service)
- 

## Risk Mitigation

---

### Security

- ☒ Webhook signature verification (prevent fake payments)
- ☒ Server-side credit checks (can't be bypassed)
- ☒ Transaction logging (audit trail)
- ☒ Rate limiting (prevent abuse)

### Edge Cases

- ☒ Duplicate payments (check for existing account)
- ☒ Payment failures (retry logic)
- ☒ Webhook delays (idempotency keys)
- ☒ Credit balance race conditions (database transactions)

### Data Integrity

- ☒ Foreign key constraints (data consistency)
  - ☒ Transaction logs (full audit trail)
  - ☒ Backup strategy (daily Supabase backups)
-

## Success Metrics

---

### After Implementation:

- 🕒 Account creation time: **<1 minute** (from 10-30 minutes)
  - 📈 Customer onboarding: **100% automated**
  - 💰 Revenue tracking: **100% accurate**
  - 🎯 Credit enforcement: **100% compliant**
  - 🗉 Customer satisfaction: **Significantly improved**
- 

## Next Steps

---

1. **Review Analysis:** Read full technical document
  2. **Approve Budget:** Finalize pricing and credit packages
  3. **Set Up Services:** Create Stripe and Resend accounts
  4. **Start Phase 1:** Begin database migration
  5. **Test Thoroughly:** Complete QA before launch
  6. **Deploy:** Go live with automated system
  7. **Monitor:** Track usage and payments
- 

## Key Documents

---

- **Full Analysis:** `VERIFYLENS_APP_COMPREHENSIVE_ANALYSIS.md`
  - **Current Schema:** `database/schema.sql`
  - **Admin Dashboard:** `ADMIN_DASHBOARD.md`
  - **Setup Guide:** `SETUP.md`
- 

## Contact

---

For questions about this integration:

- Technical Details: See full analysis document
  - Implementation: Follow 8-phase plan
  - Timeline: 3-4 weeks estimated
- 

**Status:** Ready to implement - all gaps identified, solutions designed ✅