

Credit System Fixes and Enhancements

Date: October 30, 2025

Issues Fixed

1. Credit Balance Not Showing on Main Verifier Page

Problem:

- The main verifier page only had a “Buy Credits” button
- No credit balance display was visible to users
- Users couldn’t see their available credits while using the tool

Solution:

- Added `CreditHeader` component import to main page (`src/app/page.tsx`)
- Replaced standalone “Buy Credits” button with `CreditHeader` component
- The `CreditHeader` component displays:
 - Current credit balance with icon
 - Refresh button for real-time updates
 - Low balance warning (when credits < 10)
 - “Buy Credits” button
- All in a beautiful, cohesive UI

Files Modified:

- `src/app/page.tsx` - Added `CreditHeader` import and replaced button section
-

2. Credits Not Reflecting After Purchase (Stripe Sandbox Issue)

Problem:

- After completing test purchases in Stripe sandbox, credits weren’t being added
- Dashboard showed 0 balance, no transaction history, and no purchases
- Issue was that Stripe webhooks don’t fire automatically in test/sandbox mode
- Webhooks only work in production or with Stripe CLI webhook forwarding

Root Cause:

The webhook handler (`src/app/api/credits/webhook/route.ts`) was correctly implemented, but:

- Stripe sandbox mode doesn’t automatically send webhooks to deployed applications
- Webhooks only fire in production with proper Stripe webhook endpoint configuration
- In test mode, webhooks require Stripe CLI with `stripe listen --forward-to` command

Solution:

Created a fallback payment verification system that works in both sandbox and production:

1. **New Verification Endpoint** (`src/app/api/credits/verify-payment/route.ts`):
 - Retrieves Stripe checkout session by `session_id`
 - Verifies payment status is “paid”

- Checks if payment was already processed (prevents duplicates)
- Manually adds credits to customer account
- Records transaction in database
- Works identically to webhook handler but is triggered by the client

2. **Dashboard Auto-Verification** (`src/app/dashboard/page.tsx`):

- Detects when user returns from successful Stripe checkout
- Automatically calls verification endpoint with `session_id`
- Processes payment and adds credits immediately
- Shows success message and refreshes dashboard
- Provides user feedback if verification fails

How It Works:

```

User clicks "Purchase" → Redirected to Stripe Checkout
↓
User completes payment with test card
↓
Stripe redirects to: /dashboard?purchase=success&session_id=xxx
↓
Dashboard detects success + session_id
↓
Calls /api/credits/verify-payment?session_id=xxx
↓
Endpoint verifies payment with Stripe API
↓
Manually adds credits to database
↓
Dashboard refreshes and shows new balance

```

Benefits:

- ☒ Works perfectly in Stripe sandbox/test mode
- ☒ Also works in production (as a backup to webhooks)
- ☒ Prevents duplicate credit additions
- ☒ Provides immediate feedback to users
- ☒ No need to configure Stripe CLI for testing

Files Created:

- `src/app/api/credits/verify-payment/route.ts` - Payment verification endpoint

Files Modified:

- `src/app/dashboard/page.tsx` - Added auto-verification on return from Stripe

3. ☒ **Bland/Plain Stripe Checkout Page**

Problem:

- Stripe checkout page was very minimal and plain
- Large white gap with no branding
- No logo or visual identity
- Generic product description
- Looked unprofessional and untrustworthy

Solution:**1. Created Professional VerifyLens Logo:**

- Generated modern, professional brand logo
- Features shield with checkmark symbol
- Blue to purple gradient color scheme
- Clean, tech-oriented design
- Transparent background PNG
- High resolution (1024x1024)

2. Enhanced Stripe Checkout Session:

- Added logo to product images
- Improved product description
- Added customer email pre-fill
- Better branding and trust signals

Changes in `src/app/api/credits/purchase/route.ts` :

```
product_data: {  
  name: creditPackage.name,  
  description: `${creditPackage.credits} credits for VerifyLens - Roblox User  
Verification Tool`, // Enhanced  
  images: [`${process.env.NEXTAUTH_URL}/verifylens-logo.png`], // Added logo  
},  
customer_email: session.user.email || undefined, // Pre-fill email
```

Result:

- Professional-looking checkout page
- Brand logo prominently displayed
- Clear product description
- Enhanced user trust and conversion rate

Files Created:

- `public/verifylens-logo.png` - Professional brand logo

Files Modified:

- `src/app/api/credits/purchase/route.ts` - Enhanced checkout session

Summary of Changes

Files Created (3):

1. `src/app/api/credits/verify-payment/route.ts` - Manual payment verification
2. `public/verifylens-logo.png` - Brand logo
3. `CREDIT_SYSTEM_FIXES.md` - This documentation

Files Modified (3):

1. `src/app/page.tsx` - Added credit balance display
 2. `src/app/dashboard/page.tsx` - Added auto-verification
 3. `src/app/api/credits/purchase/route.ts` - Enhanced Stripe checkout
-

Testing Instructions

Test Credit Purchase in Sandbox Mode:

- 1. Login to Application:**
 - Use demo customer credentials
 - Navigate to dashboard
 - 2. Purchase Credits:**
 - Click "Buy Credits" or navigate to `/dashboard`
 - Select a credit package
 - Click "Purchase"
 - 3. Complete Stripe Checkout:**
 - Use test card: `4242 4242 4242 4242`
 - Any future expiry date (e.g., 12/34)
 - Any CVC (e.g., 123)
 - Any ZIP code
 - Click "Pay"
 - 4. Verify Credits Added:**
 - You'll be redirected to dashboard
 - Success message will appear
 - Credits will be automatically added
 - Dashboard will refresh showing new balance
 - Transaction history will show the purchase
 - 5. Check Main Verifier Page:**
 - Navigate back to main page (/)
 - Credit balance should be visible in header
 - Can refresh balance with refresh button
-

Production Deployment Notes

Stripe Webhook Configuration (Recommended for Production):

While the manual verification system works in production, it's recommended to also configure webhooks:

- 1. Go to Stripe Dashboard:**
 - Navigate to Developers → Webhooks
- 2. Add Endpoint:**
 - URL: `https://your-domain.com/api/credits/webhook`
 - Events: `checkout.session.completed`
- 3. Add Webhook Secret to Environment:**
 - `STRIPE_WEBHOOK_SECRET=whsec_XXXXXXXXXXXX`
- 4. Benefits of Webhooks:**
 - More reliable (server-to-server)
 - Doesn't rely on user's browser

- Better for handling failed/delayed payments
- Industry best practice

Note: The manual verification system will continue to work as a backup even with webhooks enabled.

Environment Variables Required

Make sure these are set in your `.env` file:









```
# Stripe Configuration
STRIPE_SECRET_KEY=sk_test_XXXXXXXXXXXX
STRIPE_PUBLISHABLE_KEY=pk_test_XXXXXXXXXXXX
STRIPE_WEBHOOK_SECRET=whsec_XXXXXXXXXXXX

# Application URLs
NEXTAUTH_URL=https://your-domain.com
APP_URL=https://your-domain.com







# Database
DATABASE_URL=postgresql://user:password@host:5432/database
```

Credit System Features Summary

For Users:

-  View credit balance on main page
-  Real-time balance updates
-  Low balance warnings
-  Easy credit purchase flow
-  Professional Stripe checkout
-  Immediate credit addition
-  Transaction history
-  Purchase confirmations

For Admins:

-  Automatic credit management
-  Stripe payment integration
-  Transaction tracking
-  Customer credit monitoring
-  Works in sandbox for testing
-  Production-ready with webhooks

Future Enhancements (Optional)

1. Credit Usage Analytics:

- Add charts showing credit usage over time

- Most used search modes
- Cost per customer insights

2. **Auto-Recharge:**

- Allow customers to set up automatic recharge
- When balance drops below threshold, auto-purchase

3. **Credit Packages:**

- Volume discounts for large purchases
- Subscription-based credits
- Enterprise unlimited plans

4. **Notifications:**

- Email alerts for low balance
- Purchase confirmations via email
- Monthly usage reports

Support

If you encounter any issues with the credit system:

1. **Check Browser Console:**

- Look for error messages
- Check network requests

2. **Verify Environment Variables:**

- Ensure all Stripe keys are set
- Check database connection

3. **Test Mode vs Production:**

- Use test cards in sandbox
- Use real cards in production
- Verify Stripe mode matches

4. **Database Check:**

- Verify tables exist (customer_credits, credit_transactions, credit_packages)
- Check for any SQL errors in logs

Conclusion

All three issues have been successfully resolved:

1. ☒ Credit balance now visible on main verifier page
2. ☒ Credits reflect immediately after purchase (even in sandbox)
3. ☒ Stripe checkout page enhanced with professional branding

The application is now ready for continued development and deployment.

Commit Hash: b0d3099

Branch: main

Status: All changes pushed to GitHub