🎉 Stripe Integration - Final Verification Report

✅ All Code Changes Are Live on GitHub!

Branch: `feature/stripe-checkout-integration`
GitHub URL: https://github.com/Jgabbard61/roblox-lander/tree/feature/stripe-checkout-integration
Status: Ready for deployment to Vercel

📋 Verification Checklist

✅ Code Changes

File	Status	Description
`app/credits/page.tsx`	✅ Pushed	Placeholder removed, full Stripe integration added
`app/api/checkout/route.ts`	✅ Pushed	New checkout API endpoint created
`app/credits/success/page.tsx`	✅ Pushed	Success page created
`package.json`	✅ Pushed	Stripe dependencies added
`.gitignore`	✅ Pushed	.env protection added
`STRIPE_INTEGRATION.md`	✅ Pushed	Complete integration docs

✅ Key Features Implemented

- [x] Removed placeholder message: “Stripe integration coming soon!”
- [x] Integrated Stripe Checkout with `loadStripe()` and session creation
- [x] Added loading states on purchase buttons
- [x] Implemented cancel flow with toast notifications
- [x] Created success page with session ID display
- [x] Added comprehensive error handling
- [x] Included webhook metadata for credit allocation
- [x] Added comprehensive documentation

✅ Git Status

```
Current Branch: feature/stripe-checkout-integration
Latest Commit: 9bfa968 - fix: Update Stripe API version
Remote Status: ✅ All code changes synced with GitHub
Total Commits on this branch: 24 commits ahead of main
```

🚀 Next Steps for Deployment

1. Deploy to Vercel (5 minutes)

Go to your Vercel dashboard and either:
- **Option A**: Change production branch to `feature/stripe-checkout-integration`
- **Option B**: Create a preview deployment for testing first

2. Add Environment Variables (2 minutes)

In Vercel Settings → Environment Variables, add:

```
NEXT_PUBLIC_STRIPE_PUBLISHABLE_KEY=pk_test_...
STRIPE_SECRET_KEY=sk_test_...
```

Get these from: https://dashboard.stripe.com/test/apikeys

3. Configure Stripe Webhook (3 minutes)

After deployment:
1. Go to: https://dashboard.stripe.com/test/webhooks
2. Add endpoint: `https://www.verifylens.com/api/credits/webhook`
3. Select events: `checkout.session.completed`, `payment_intent.succeeded`
4. Copy webhook secret and add to main app’s environment variables

4. Test the Integration (5 minutes)

1. Visit: `https://site.verifylens.com/credits`
2. Click “Purchase Package”
3. Use test card: `4242 4242 4242 4242`
4. Complete checkout
5. Verify redirect to success page
6. Check webhook received event in main app

🔍 Code Verification

Credits Page - No Placeholder Message ✅

```
grep -r "Stripe integration coming soon" app/credits/page.tsx
# Result: No matches found (placeholder successfully removed!)
```

Purchase Handler Exists ✅

```
const handlePurchase = async (packageIndex: number) => {
  // Full Stripe checkout implementation
  // Creates checkout session
  // Redirects to Stripe hosted page
  // Handles errors and loading states
}
```

Checkout API Endpoint ✅

```
// POST /api/checkout
// Creates Stripe Checkout Session
// Returns session URL for redirect
// Includes metadata for webhook
```

Success Page ✅

```
// /credits/success?session_id={CHECKOUT_SESSION_ID}
// Displays confirmation
// Shows session ID
// Links to main app
```

📊 Purchase Flow

```
┌────────────────────────────────────────────────────────────────┐
│  1. User visits: site.verifylens.com/credits                   │
│  2. Clicks "Purchase Package" button                           │
│  3. Frontend calls: POST /api/checkout                         │
│  4. Backend creates Stripe Checkout Session                    │
│  5. User redirected to: checkout.stripe.com                    │
│  6. User enters payment info (secure on Stripe)                │
│  7. Payment processed by Stripe                                │
│  8. User redirected to: /credits/success?session_id=...        │
│  9. Stripe sends webhook to: www.verifylens.com/api/webhook   │
│ 10. Main app adds credits to customer account                  │
│ 11. Customer can immediately use credits                       │
└────────────────────────────────────────────────────────────────┘
```

🎯 What Was Fixed

Before (Current Production)

```
// Old placeholder code
alert("Stripe integration coming soon! For now, please contact us to purchase credits.")
// No actual purchase functionality
```

After (This Branch)

```
// New working implementation
const handlePurchase = async (packageIndex: number) => {
  setIsLoading(true)
  const response = await fetch('/api/checkout', {
    method: 'POST',
    body: JSON.stringify({ packageName, credits, price })
  })
  const { url } = await response.json()
  window.location.href = url // Redirect to Stripe
}
// Full Stripe Checkout integration!
```

📦 Packages Added

```
{
  "@stripe/stripe-js": "^4.9.0",
  "stripe": "^17.5.0",
  "react-hot-toast": "^2.4.1"
}
```

⚙️ Environment Variables Reference

Landing Page (Vercel - roblox-lander)

```
NEXT_PUBLIC_STRIPE_PUBLISHABLE_KEY=pk_test_51QKH2qCUW...
STRIPE_SECRET_KEY=sk_test_51QKH2qCUW...
```

Main App (www.verifylens.com)

```
STRIPE_WEBHOOK_SECRET=whsec_...
```

🧪 Testing Guide

Test Cards (Stripe Test Mode)

Card Number	Result
4242 4242 4242 4242	Success
4000 0000 0000 0002	Card declined
4000 0025 0000 3155	Requires authentication

Note: Use any future expiration date and any 3-digit CVC

Testing Checklist

- [ ] Visit credits page
- [ ] Click purchase button
- [ ] Redirected to Stripe checkout
- [ ] Enter test card info
- [ ] Complete payment
- [ ] Redirected to success page
- [ ] Session ID displayed
- [ ] Webhook event received in main app
- [ ] Credits added to account
- [ ] Test cancel flow (back button)
- [ ] Test on mobile devices

🔐 Security Notes

- ✅ All payment data handled by Stripe (PCI compliant)
- ✅ API keys stored in environment variables
- ✅ .env file excluded from git
- ✅ Webhook signature verification required
- ✅ HTTPS required for production webhooks
- ✅ No sensitive data in client-side code

📞 Support & Troubleshooting

If purchase button doesn’t work:

1. Check browser console for errors
2. Verify `NEXT_PUBLIC_STRIPE_PUBLISHABLE_KEY` is set in Vercel
3. Check that key starts with `pk_test_` or `pk_live_`

If redirect fails:

1. Check Vercel function logs
2. Verify `STRIPE_SECRET_KEY` is set correctly
3. Check Stripe Dashboard logs

If webhook doesn’t receive events:

1. Verify webhook URL: `https://www.verifylens.com/api/credits/webhook`
2. Check webhook secret matches Stripe Dashboard
3. Ensure main app webhook endpoint is accessible
4. Check Stripe webhook delivery logs

If credits aren’t added:

1. Check webhook is configured in Stripe Dashboard

2. Verify metadata is being passed correctly
3. Check main app logs for webhook processing
4. Verify `customer_stats` table update

📈 Monitoring After Deployment

Day 1-3 (Test Mode)

- [ ] Monitor Vercel deployment logs
- [ ] Check Stripe Dashboard for test transactions
- [ ] Verify webhook delivery success rate
- [ ] Test all package tiers
- [ ] Test cancel flow
- [ ] Test on different devices/browsers

Week 1 (Production)

- [ ] Switch to live Stripe keys
- [ ] Monitor real transactions
- [ ] Track webhook success rate
- [ ] Monitor customer support tickets
- [ ] Gather user feedback

🎊 Summary

✅ **All code changes are on GitHub**
✅ **Placeholder message removed**
✅ **Full Stripe Checkout integrated**
✅ **Success page created**
✅ **Documentation complete**
✅ **Ready for deployment**

Next Action: Deploy to Vercel and add environment variables!

Report Generated: October 28, 2025
Branch: feature/stripe-checkout-integration
Commits: 24 ahead of main
Status: ✅ Ready for Production