




# Major Update: Duplicate Search Prevention & Transaction History Pagination

---

## Overview

---

This update adds three major features to improve user experience and prevent unnecessary credit charges:

1.  **Transaction History Pagination** - View all past transactions with full navigation
  2.  **Duplicate Search Prevention** - No charges for searching the same term multiple times
  3.  **“No Results” Transaction Logging** - Complete audit trail including free searches
- 

## Feature 1: Transaction History Pagination

---

### What Changed

- Transaction history now shows **10 items per page** with full pagination controls
- Users can navigate through **all** their transaction history (not just the latest 10)
- Clean, responsive pagination UI with page numbers
- Shows “Showing X to Y of Z transactions”
- Loading states during page transitions
- Auto-refreshes after successful purchases

### User Impact

- **Before:** Could only see the latest 10 transactions
- **After:** Can browse through entire transaction history with Previous/Next buttons and page numbers

### Technical Details

- Added pagination state management to dashboard
  - Updated `/api/credits/transactions` endpoint (already supported limit/offset)
  - Smooth UX with loading indicators
- 

## Feature 2: Duplicate Search Prevention

---

### What Changed

- New **search result cache** system stores past searches per customer
- When a user searches for the same term again:
- Returns cached result **instantly** (0ms response time)
- **Does NOT charge any credits**
- Records a “free search” transaction with 0 credits
- Works for both **Smart Search** and **Exact Search**

- Cache is per-customer, so different customers maintain separate caches

## User Impact

- **Before:** Searching “john\_doe” 5 times = 5 credits charged
- **After:** Searching “john\_doe” 5 times = 1 credit charged (first search only)

### Example Scenario:

User searches “john\_doe” (Smart Search):

- 1st time: 1 credit charged ✓
- 2nd time: 0 credits (cached) ✓
- 3rd time: 0 credits (cached) ✓
- 100th time: 0 credits (cached) ✓

## Technical Details

- New table: `search_result_cache`
- Stores: search term (normalized), search type, results, timestamps
- Unique constraint: `(customer_id, search_term, search_type)`
- Tracks access count and last accessed time
- Auto-updates on cache hits via database trigger

## Cache Management

- Results are cached indefinitely (until manually cleared)
- Optional maintenance: Clear cache entries older than X days
- Access count tracking helps identify frequently searched terms



## Feature 3: “No Results” Transaction Logging

### What Changed

- Searches that return **no results** now properly log transactions
- These transactions show:
  - **Amount: 0 credits**
  - **Description:** Clear reason (e.g., “Exact search for ‘xyz’ (no results, no charge)”)
  - **Balance:** Unchanged
- Provides complete audit trail of all search activity

### User Impact

- **Before:** No record when searches returned no results
- **After:** Complete transaction history showing all searches, including free ones

### Technical Details

- New function: `recordFreeSearch()` in credits service
- Creates transaction record with `amount: 0`
- Balance before = Balance after (no change)
- Linked to `search_history` for full traceability

## Database Migration Required

---

### IMPORTANT: Run this migration on production

**File:** `RUN_MIGRATION_007.sql`

#### What it does:

- Creates `search_result_cache` table
- Adds indexes for fast lookups
- Creates trigger for auto-updating access tracking
- Adds comments for documentation

#### How to run:

```
# Using psql
psql -h your-db-host -U your-db-user -d your-db-name -f RUN_MIGRATION_007.sql

# Or copy/paste the SQL into your database admin panel
```

#### Migration is safe:

- Uses `CREATE TABLE IF NOT EXISTS`
  - No data loss
  - No downtime required
  - Can be run multiple times safely
- 

## Files Changed

---

### New Files

1. `database/migrations/007_add_search_result_cache.sql` - Migration file
2. `RUN_MIGRATION_007.sql` - Production migration script
3. `src/app/lib/search-cache/index.ts` - Search cache module

### Modified Files

1. `src/app/api/search/route.tsx` - Added duplicate detection and caching
  2. `src/app/dashboard/page.tsx` - Added pagination UI and state
  3. `src/app/lib/credits/index.ts` - Added `recordFreeSearch()` function
  4. `src/app/lib/utils/cache.ts` - Added `DUPLICATE_SEARCH` TTL constant
- 

## Testing Checklist

---

### Test Duplicate Search Prevention

1. ☒ Search for “testuser123” (Smart Search) - Should charge 1 credit
2. ☒ Search for “testuser123” again - Should show 0 credits charged
3. ☒ Check transaction history - Should see duplicate entry with description
4. ☒ Switch to Exact Search for “testuser123” - Should charge 1 credit (different cache)
5. ☒ Search for “testuser123” (Exact) again - Should show 0 credits

## Test Transaction History Pagination

1. ☒ Ensure you have more than 10 transactions
2. ☒ Go to dashboard - Should see pagination controls
3. ☒ Click "Next" - Should load page 2
4. ☒ Click "Previous" - Should go back to page 1
5. ☒ Click page number directly - Should jump to that page
6. ☒ Verify "Showing X to Y of Z" is accurate

## Test "No Results" Logging

1. ☒ Search for "definitelynotarealuser99999" (Exact Search)
2. ☒ Should return no results
3. ☒ Check transaction history - Should see entry with 0 credits
4. ☒ Description should say "no results, no charge"
5. ☒ Balance should remain unchanged



## Expected Behavior Summary

Search Scenario	Credits Charged	Transaction Recorded
First Smart Search (results found)	1	Yes (1 credit)
Duplicate Smart Search	0	Yes (0 credits, "cached result")
First Exact Search (results found)	1	Yes (1 credit)
Duplicate Exact Search	0	Yes (0 credits, "cached result")
Exact Search (no results)	0	Yes (0 credits, "no results")
Duplicate Exact Search (no results)	0	Yes (0 credits, "cached result")



## UI/UX Improvements

### Transaction History

- Clean pagination with 5 visible page numbers
- Previous/Next buttons for easy navigation
- Disabled state for buttons at boundaries
- Loading state shows opacity during fetch

- Transaction count summary at bottom

## Duplicate Search Indicators

- Console logs show “[Duplicate Search Detected]”
  - Transaction description clearly states “cached result, no charge”
  - Instant response (0ms) for cached results
- 

## Security & Performance

---

### Security

- Cache is customer-scoped (customers can’t access each other’s cache)
- Search terms are normalized (lowercase, trimmed) consistently
- No sensitive data exposed in cache

### Performance

- **Duplicate searches:** Instant (0ms) from database cache
  - **Database indexes:** Optimized for fast lookups
  - **Pagination:** Efficient with LIMIT/OFFSET
  - **Auto-refresh:** Only refetches current page, not entire history
- 

## Deployment Instructions

---

1. **Deploy code** (already done via git push)
2. **Run migration** on production database:

```
sql
-- Copy contents of RUN_MIGRATION_007.sql and execute
```

3. **Verify migration:**

```
sql
SELECT * FROM search_result_cache LIMIT 1;
-- Should return empty result (new table) with no errors
```

4. **Test on production:**

- Perform a search
  - Perform the same search again
  - Check transaction history
- 

## Future Enhancements

---

### Potential Additions

1. **Cache expiration:** Add optional TTL for cache entries
2. **Cache statistics:** Show users how many credits they’ve saved via cache hits
3. **Admin dashboard:** View cache hit rates across all customers
4. **Manual cache clear:** Allow users to clear their search cache
5. **Export transactions:** Download transaction history as CSV

---

## Support

---





If you encounter any issues:

1. Check that migration 007 was run successfully
  2. Verify database connection is working
  3. Check browser console for any errors
  4. Review server logs for error messages
- 





## Summary

---

### What users get:

- Never pay for the same search twice 
- View complete transaction history 
- Better transparency on credit usage 
- Faster duplicate searches (instant) 

### What you get:

- Complete audit trail of all searches 
- Better customer satisfaction 
- Reduced support tickets about duplicate charges 
- Professional, modern UI 

### Migration Required:

- Yes - Run `RUN_MIGRATION_007.sql` on production database 
- 

**Date:** October 30, 2025

**Commit:** d24d305

**Status:**  Deployed to GitHub, Ready for Production