

Admin Dashboard Documentation

Overview

The Roblox Verifier Tool now includes a comprehensive Super Admin dashboard with full CRUD operations for users, customers, and advanced search history analytics.

Features Implemented

1. Dashboard Overview

- Real-time statistics (customers, users, searches, success rate)
- User breakdown by role
- Search statistics
- Top customers by search volume
- Recent activity feed

2. Customer Management

- ☒ View all customers with stats
- ☒ Create new customers with admin users
- ☒ Edit customer details (name, contact email, max users)
- ☒ Activate/Deactivate customers
- ☒ Delete customers (with cascade warning)
- Search and filter capabilities

3. User Management

- ☒ View all users with pagination
- ☒ Create new users (assign role and customer)
- ☒ Edit user details (role, customer, status)
- ☒ Reset user passwords
- ☒ Delete users
- Advanced filtering (by customer, role, search term)

4. Search History

- ☒ View all search history with pagination
- ☒ Advanced filters:
 - Customer (including Super Admin searches)
 - Search type (username, userId, displayName, etc.)
 - Result status (success, no_results, error)
 - Date range
 - Search term
- Detailed search results with timing and user info

Critical Bug Fixed

Bug: Super Admin searches were not being logged in the `search_history` table.

Root Cause: The database schema required `customer_id NOT NULL`, but `SUPER_ADMIN` users don't have a `customer_id` (it's `NULL` by design).

Solution:

1. Modified database schema to allow `NULL` `customer_id`
2. Updated search logging logic to handle `NULL` `customer_id`
3. Updated middleware to always set customer headers
4. Updated search API route to log all searches regardless of role

Migration Required: See `database/migrations/001_allow_null_customer_id_in_search_history.sql`

New Files Created

API Routes

- `/src/app/api/admin/users/route.ts` - User CRUD operations
- `/src/app/api/admin/users/password/route.ts` - Password reset
- `/src/app/api/admin/search-history/route.ts` - Search history with filters
- `/src/app/api/admin/stats/route.ts` - Dashboard statistics
- Updated `/src/app/api/admin/customers/route.ts` - Added PUT and DELETE methods

Components

- `/src/app/admin/components/DashboardOverview.tsx` - Dashboard overview
- `/src/app/admin/components/CustomerManagement.tsx` - Customer management
- `/src/app/admin/components/UserManagement.tsx` - User management
- `/src/app/admin/components/SearchHistory.tsx` - Search history viewer

Database

- `/database/migrations/001_allow_null_customer_id_in_search_history.sql` - Migration script
- `/database/migrations/APPLY_MIGRATION.md` - Migration instructions

Deployment Steps

Step 1: Apply Database Migration

CRITICAL: You must apply the database migration before using the new dashboard.

Option 1: Using Supabase SQL Editor (Recommended)

1. Go to your Supabase project dashboard
2. Navigate to SQL Editor
3. Copy the contents of `database/migrations/001_allow_null_customer_id_in_search_history.sql`
4. Paste and execute

Option 2: Using `psql`

```
psql "YOUR_SUPABASE_DATABASE_URL" < database/migrations/001_allow_null_customer_id_in_search_history.sql
```

Step 2: Verify Migration

Run this query to verify the migration was successful:

```
SELECT
  column_name,
  data_type,
  is_nullable
FROM information_schema.columns
WHERE table_name = 'search_history'
AND column_name = 'customer_id';
```

You should see `is_nullable = 'YES'`.

Step 3: Deploy to Vercel

The code is ready to deploy. Push to your GitHub repository and Vercel will automatically deploy.

```
git add .
git commit -m "feat: Add comprehensive admin dashboard with user/customer management
and fix super admin search logging"
git push origin main
```

Step 4: Test the Dashboard

1. Log in as a SUPER_ADMIN
2. Navigate to the admin dashboard
3. Test each tab:
 - Dashboard Overview
 - Customers (create, edit, delete)
 - Users (create, edit, reset password, delete)
 - Search History (filter by various criteria)



Security & Access Control

- All admin routes are protected by middleware
- Only SUPER_ADMIN role can access the dashboard
- All API routes verify SUPER_ADMIN role
- Customer cascading deletes properly warn users
- Password resets display credentials (save before closing modal)



Dashboard Features

Dashboard Tab

- Total customers, users, searches, and success rate
- User breakdown by role (Super Admin, Customer Admin, Customer User)
- Search statistics (successful, no results, errors)
- Top 10 customers by search volume (last 30 days)
- Recent activity (last 10 searches)

Customers Tab

- Create customer with admin user
- Edit customer (name, contact email, max users)
- Activate/Deactivate customer
- Delete customer (with cascade warning if users exist)
- View customer stats (users, searches)

Users Tab

- Create user (username, password, role, customer, email, name)
- Edit user (role, customer, status, email, name)
- Reset user password (with credential display)
- Delete user
- Filter by customer, role, and search term
- Pagination support (50 per page)

Search History Tab

- View all searches with full details
- Filter by:
 - Customer (including “Super Admin Searches”)
 - Search type (username, userId, displayName, url, exact, smart)
 - Result status (success, no_results, error)
 - Date range (start and end date)
 - Search term (query or username)
- Pagination support (50 per page)
- Display search results, timing, and user info



UI/UX Features

- Modern gradient design with purple/pink theme
- Responsive layout (mobile, tablet, desktop)
- Loading states for all operations
- Error handling with user-friendly messages
- Success notifications for all actions
- Confirmation dialogs for destructive operations
- Modal forms for create/edit operations
- Hover effects and smooth transitions



Technical Details

Database Schema Changes

- `search_history.customer_id` now allows NULL
- Added index on `search_history.customer_id` (partial index for non-NULL values)
- Added index on `search_history.user_id`

API Endpoints

Stats API

- GET /api/admin/stats?days=30 - Get dashboard statistics

Users API

- GET /api/admin/users?page=1&limit=50&customerId=1&role=SUPER_ADMIN&search=john - List users
- POST /api/admin/users - Create user
- PATCH /api/admin/users - Update user
- DELETE /api/admin/users?userId=1 - Delete user
- POST /api/admin/users/password - Reset password

Customers API (Enhanced)

- GET /api/admin/customers - List customers
- POST /api/admin/customers - Create customer
- PATCH /api/admin/customers - Update customer status
- PUT /api/admin/customers - Update customer details
- DELETE /api/admin/customers?customerId=1&force=true - Delete customer

Search History API

- GET /api/admin/search-history?page=1&limit=50&customerId=1&searchType=username&resultStatus=success&startDate=2025-01-01&endDate=2025-12-31&search=john - Get search history



Notes

1. **Password Display:** When creating users or resetting passwords, credentials are displayed in an alert. Make sure to save them before closing!
2. **Customer Deletion:** Deleting a customer will CASCADE delete all associated users and search history. A warning is shown with user count.
3. **Super Admin Searches:** These now appear in search history with “Super Admin” as the customer name and `customer_id = NULL` in the database.
4. **Pagination:** All tables support pagination. Default is 50 items per page.
5. **Filters:** All filters are preserved when paginating. Use “Clear Filters” to reset.



Known Limitations

- No CSV export functionality for search history (can be added if needed)
- No bulk user/customer operations (can be added if needed)
- No activity logs for admin actions (can be added if needed)



Support

If you encounter any issues:

1. Check browser console for errors
2. Check Vercel deployment logs
3. Verify database migration was applied correctly
4. Ensure SUPER_ADMIN user has correct permissions

Testing Checklist

- ☐ Database migration applied successfully
- ☐ Can access admin dashboard as SUPER_ADMIN
- ☐ Dashboard overview shows correct statistics
- ☐ Can create/edit/delete customers
- ☐ Can create/edit/delete users
- ☐ Can reset user passwords
- ☐ Search history filters work correctly
- ☐ Pagination works on all tables
- ☐ Super Admin searches are now logged
- ☐ All modals open and close correctly
- ☐ Error handling works properly
- ☐ Responsive design works on mobile

Conclusion

The admin dashboard is now fully functional with comprehensive management capabilities. All CRUD operations are implemented with proper error handling, user feedback, and security measures.