GISTer Development Session Summary

Date: October 14, 2025
Checkpoint: “Dev helper: Use Sample button”
Session Type: Development tooling improvement

Problem Statement

User requested a faster way to test features during development without going through the full camera capture and AI analysis flow every time. This was slowing down iteration speed when testing other parts of the application.

Solution Implemented

Added a **development-only “Use Sample” button** that instantly creates a test listing with mock data.

Key Features:

1. **One-Click Sample Listing Creation**
   - Button visible only in development mode (`NODE_ENV !== 'production'`)
   - Randomly selects from 5 pre-configured sample items
   - Creates listing in <1 second without AI processing
   - Redirects directly to listing detail page

2. **Sample Data Library**
   - 5 diverse sample items covering different categories:
     - Vintage Sony Walkman (Electronics, Special Item)
     - Nike Air Jordan 1 (Footwear, Special Item)
     - Vintage Pyrex Set (Home & Kitchen, Special Item)
     - MacBook Pro M1 (Computers, Regular Item)
     - Patagonia Fleece (Clothing, Regular Item)
     - Includes both premium and regular items for testing
     - Realistic pricing and descriptions
     - Mock platform recommendations

3. **UI Integration**
   - Orange “Use Sample” button in top-right of camera viewfinder
   - Visible alongside user name and status indicators
   - Loading state with spinner when creating
   - Non-intrusive placement

Technical Implementation

New API Endpoint

File: `app/api/dev/sample-listing/route.ts`

Features:
- Production safety: Returns 403 in production environment
- Authentication: Requires valid session
- Random selection: Picks from sample items array
- Database creation: Uses Prisma to create listing with correct schema fields

Schema Fields Used:
- `title`, `description`, `category`, `condition`, `price`
- `theGist` (prefixed with "DEV SAMPLE:")
- `confidence` (hardcoded to 0.95)
- `isPremiumItem`, `specialClass` (for special items testing)
- `recommendedPlatforms`, `qualifiedPlatforms`
- `marketInsights`, `premiumFacts`
- `tokensUsed: 0`, `apiCost: 0` (no AI processing)

UI Changes

File: `app/camera/_components/camera-screen.tsx`

Additions:
- `isDev` constant checking `process.env.NODE_ENV`
- `isCreatingSample` state for loading indicator
- `createSampleListing()` async function
- Button component in camera overlay (flex container)

Benefits

For Development:

- ⚡ **Faster Iteration:** Create test listings in <1 second vs 30+ seconds with AI
- 🎯 **Focused Testing:** Test listing detail, editing, posting without waiting
- 📊 **Varied Data:** Test both premium and regular items instantly
- 🔄 **Repeatable:** Consistent test data for debugging

For Production:

- ✅ **Zero Impact:** Button hidden in production
- 🔒 **Safe:** API endpoint rejects production requests
- 🚫 **No Cost:** No AI API calls for dev samples

Usage

Development Mode:

1. Open camera page at `/camera`

2. Click orange “Use Sample” button in top-right
3. Random sample listing created and displayed
4. Test listing features (edit, post, etc.)

Production Mode:

- Button automatically hidden
- API endpoint returns 403 if called

Files Modified

1. `app/api/dev/sample-listing/route.ts` (new file, 126 lines)
   - Development-only API endpoint
   - Sample data library (5 items)
   - Prisma listing creation

2. `app/camera/_components/camera-screen.tsx`
   - Added dev mode detection
   - Added sample listing creation function
   - Added “Use Sample” button to UI
   - Reorganized camera overlay layout

3. `docs/CHANGELOG.md`
   - Added Session 2 entry
   - Documented new dev tools

4. `sessions/2025-10-14_dev-sample-helper.md` (this file)
   - Comprehensive session summary

Schema Fields Reference

During implementation, corrected several field names to match actual Prisma schema:

Incorrect (initial)	Correct (schema)	Type
`estimatedPrice`	`price`	Float?
`isSpecialItem`	`isPremiumItem`	Boolean
`specialItemCategory`	`specialClass`	String?
`specialItemReason`	N/A (removed)	-
`platformRecommendations`	`recommendedPlatforms`	String[]
-	`qualifiedPlatforms`	String[]
`premiumMarketInsight`	`marketInsights`	String?

Testing Results

Build Status:

- ✅ TypeScript compilation: Success (exit_code=0)
- ✅ Next.js build: Success
- ✅ Dev server: Running on localhost:3000
- ✅ No runtime errors
- ✅ Button renders correctly in development

Functionality Verified:

- ✅ Button only shows in dev mode
- ✅ Sample listing created successfully
- ✅ Redirects to listing detail page
- ✅ Listing data displays correctly
- ✅ No conflicts with existing camera flow

Next Steps

Potential Enhancements:

1. Add more sample items (different categories)
2. Allow selecting specific sample item (dropdown?)
3. Add "Clear All Dev Samples" button
4. Generate sample listings with photos (use placeholder images)
5. Add dev mode toggle in UI settings

Priority: Low

These are nice-to-have improvements. Current implementation meets immediate need.

Documentation Updates

✅ **CHANGELOG.md:** Updated with Session 2 changes
✅ **Session Summary:** This file created
⚠️ **API.md:** Should document `/api/dev/sample-listing` endpoint
⚠️ **FEATURES.md:** Could add "Development Tools" section

Git Status

Commit Message: `feat(dev): Add Use Sample button for faster testing`

Files Changed:
- `.abacus.donotdelete` (metadata)

- `app/app/api/dev/sample-listing/route.ts` (new)
- `app/app/camera/_components/camera-screen.tsx` (modified)

Lines Changed: +190, -15

Key Learnings

1. **Schema Verification:** Always check actual Prisma schema fields before writing code
2. **Production Safety:** Use environment checks for dev-only features
3. **Sample Data Quality:** Realistic sample data improves testing effectiveness
4. **Quick Wins:** Small dev tools can dramatically improve workflow

---

Session completed successfully. Development velocity improved! 🚀