# Google Photos Navigation POC - Proof of Concept

## Project Purpose

Build a **simple test application** to validate that we can reliably automate navigation in Google Photos by clicking the "next photo" arrow button.

## Why This POC?

Before investing time in building a full tagging assistant application, we need to verify the core assumption: **Can we programmatically click the > button in Google Photos to navigate between photos?**

This POC will answer that question in 1-2 hours instead of discovering problems after days of development.

---

## POC Scope (Minimal Viable Test)

### What It Does:

1. Opens Chrome/Edge browser to photos.google.com
2. Pauses for user to manually:
   - Sign into Google Photos
   - Navigate to any photo
   - Click the (i) button to open info panel (optional, just for visual reference)
3. Displays simple UI window with "NEXT" button
4. When user clicks "NEXT":
   - App locates the **>** (next arrow) button in Google Photos
   - Clicks it
   - Waits 2-3 seconds for photo to load
   - Ready for next click
5. Counter shows how many photos navigated

### What It Does NOT Do:

- ❌No description reading or writing
- ❌No name management
- ❌No data persistence
- ❌No progress tracking
- ❌No error recovery
- ❌No configuration or settings

**This is purely a navigation test.**

---

## Technical Requirements

### Platform

- Python 3.8+
- Selenium or Playwright for browser automation
- Tkinter for simple UI (built into Python)
- Chrome or Edge browser

### Two-Window System

**Window 1: Test UI (User Control)**

- Simple window with one button
- Shows click counter
- User clicks "NEXT" to test navigation

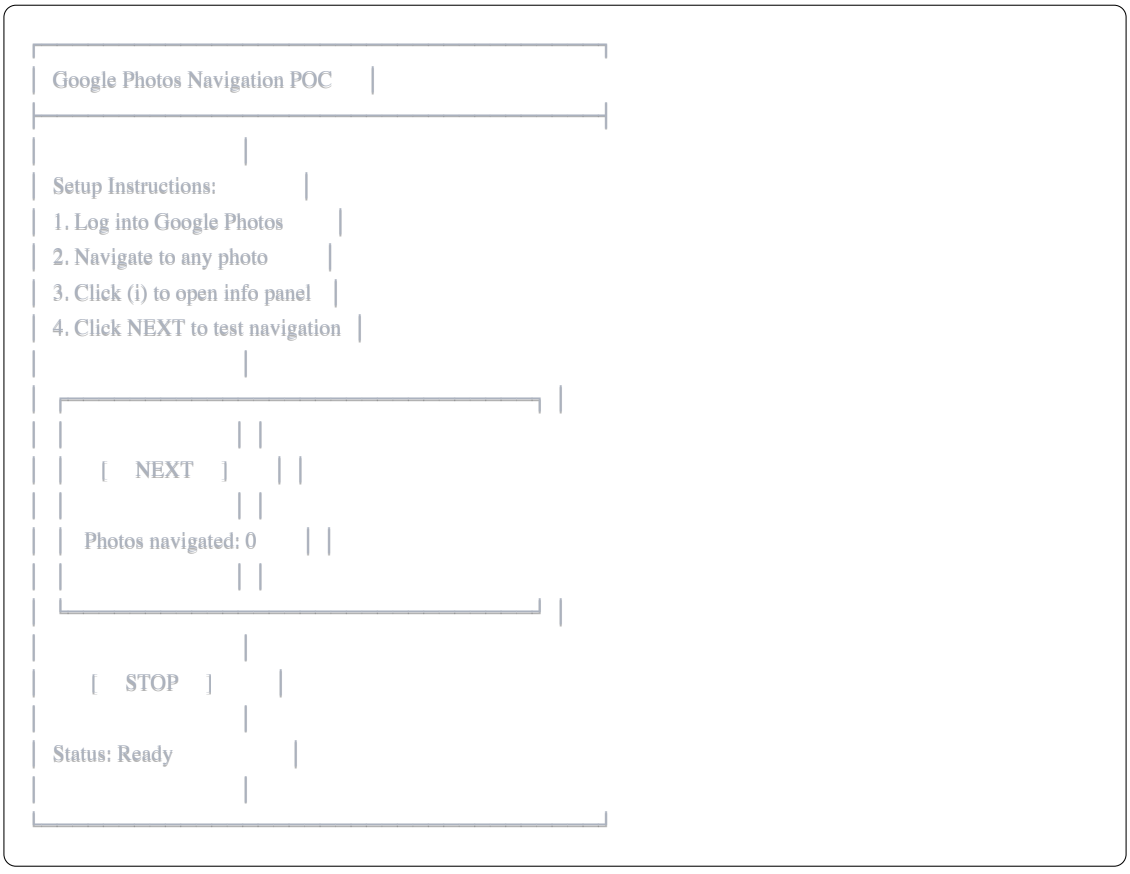
**Window 2: Browser (Google Photos)**

- Automated Chrome/Edge window
- User logs in manually
- User navigates to any photo manually
- App will click the > button when user clicks "NEXT"

---

## User Interface

### Test UI Window (Simple)




---

## How To Use (Test Process)

### Step 1: Start the App

- Run the Python script
- Browser window opens to photos.google.com
- Test UI window appears

### Step 2: Manual Setup

- In browser window: Log into Google Photos
- Navigate to any photo (any date, any album)
- Optionally click (i) to open info panel
- Return to Test UI window

### Step 3: Test Navigation

- Click "NEXT" button in UI
- Observe: Does photo advance in browser?
- Click "NEXT" again
- Repeat 20-30 times

### Step 4: Validation

**Success criteria:**

- ✅> button is found and clicked each time
- ✅Photo advances to next photo reliably
- ✅Works consistently across 20-30+ clicks
- ✅No errors or failures
- ✅Info panel stays open (if opened)

**Failure indicators:**

- ❌Can't find > button
- ❌Clicks wrong element
- ❌Photos don't advance
- ❌Errors after several clicks
- ❌Unreliable behavior

### Step 5: Stop

- Click "STOP" button to close browser and exit

---

## Implementation Details

### Browser Automation

- Launch Chrome/Edge in visible mode (headful)
- Navigate to photos.google.com
- Wait for user to complete manual setup
- Locate > button using:
  - CSS selector
  - ARIA label
  - XPath (whatever works most reliably)
- Click button when user clicks "NEXT"
- Simple 2-3 second wait after click

### Error Handling (Minimal)

- If > button not found: Display error in status
- If click fails: Display error in status
- User must manually troubleshoot (POC only)

### No Data Storage

- Counter resets each run
- No persistence needed
- No configuration files

---

## Success Criteria

### POC Passes If:

1. Can navigate through 30+ photos without failure
2. button is found reliably every time
3. Navigation feels smooth and consistent
4. No manual intervention needed during test
5. Works on both Windows and macOS (if testing both)

### POC Fails If:

- Can't locate > button consistently
- Clicks wrong elements
- Navigation is unreliable
- Requires constant manual fixes
- Google Photos structure prevents automation

---

## Next Steps Based On Results

### If POC Succeeds:

✅Core assumption validated
✅Proceed with full application development
✅Use same navigation logic in final app
✅Confidence that project is viable

### If POC Fails:

❌Investigate why navigation failed
❌Try alternative selectors or approaches
❌May need to reconsider project approach
❌Saves time before building complete app

---

## Deliverables

### What I Need:

1. **Python script** (poc_navigation_test.py)
   - Simple, well-commented code
   - ~100-150 lines maximum
2. **Setup instructions**
   - How to install Python dependencies
   - How to install ChromeDriver/WebDriver
   - Step-by-step for Windows and macOS
3. **Quick start guide**
   - How to run the script
   - What to expect
   - How to interpret results

### What I'll Provide:

- Feedback on whether navigation works reliably
- Any error messages or issues encountered
- Decision on whether to proceed with full app

---

## Timeline

- **POC Development**: 1-2 hours
- **Testing**: 15-30 minutes
- **Decision**: Immediate (proceed or pivot)

---

## Technical Notes

### Browser Element to Find

The **>** (next photo) button in Google Photos:

- Located on right middle side of photo
- May have ARIA label like "Next photo" or similar
- May be in a button or div element
- Should be visible when viewing a single photo

### Simple Wait Strategy

After clicking >:

```python
time.sleep(2.5)  # Simple fixed wait
```

No need for complex waits in POC.

### UI Button Handler

```python
def on_next_click():
    # Find and click > button in browser
    # Increment counter
    # Update status
```

---

## Questions This POC Answers

1. ✅Can we find the > button programmatically?
2. ✅Can we click it reliably?
3. ✅Does Google Photos respond to automated clicks?
4. ✅Is navigation consistent across multiple photos?
5. ✅Are there any rate limits or bot detection for navigation?

**After POC:** We'll know if the full project is viable or needs a different approach.

---

**Ready to build this POC first before proceeding to the full application.**