**📛 Project Title:**

**SortWise – Smart File Organizer for macOS**

**1. 🎯 Purpose and Goals**

**Objective:**  
SortWise is a macOS-only desktop utility app designed to help everyday users organize large volumes of images and videos into meaningful subfolders using filters such as date, camera model, file type, and more. It will feature a user-friendly GUI with a single-window design and polished UI built using Electron (frontend) with a Python backend handling file operations.

**Primary Use Case:**  
A user selects one or more folders containing disorganized media. The app analyzes metadata and offers filter-based sorting into subfolders, giving the user control over the sorting criteria, file handling behavior, and a visual preview before execution.

**Secondary Goal:**  
Serve as a portfolio-worthy desktop application to showcase full-stack desktop app development skills (Electron + Python + metadata processing).

**2. 🧑‍💻 Target Audience**

* Everyday macOS users with basic computer skills
* People with large photo/video libraries (e.g., from iPhones, DSLRs, WhatsApp backups, etc.)
* Users who prefer offline tools for local file management
* Personal use or demo purposes for recruiters/clients

**3. 💻 Platforms and Environment**

* **Operating System**: macOS (minimum: macOS 12 Monterey)
* **Packaging**: Standalone .app bundle, no Terminal or Python installation required
* **Architecture**:
  + **Frontend**: Electron (HTML/CSS/JavaScript/React)
  + **Backend**: Python (via Node child process or IPC bridge)
* **Virtual Environment**: Python backend will be bundled using pyinstaller or similar to work standalone.

**4. 📂 Input Options**

* Users can select **one or more folders** using a folder picker dialog.
* Drag and drop is **not required** in this version.

**5. 📦 Sorting Features (Phase 1)**

| **Sorting Criterion** | **Description** |
| --- | --- |
| 📅 EXIF Date | Organize into /YYYY/ or /YYYY/MM/ folders based on metadata |
| 📸 Camera Make/Model | Sort into folders like /iPhone\_13/, /Canon\_80D/, etc. |
| 📁 File Type | Organize by Photos, Videos, GIFs, Screenshots |
| 🌍 Location (GPS) | Sort into folders based on city or country (if GPS metadata exists) |
| ↕️ Orientation | Separate Portrait vs Landscape images |
| 🎞️ Live Photos | Group .jpg and .mov with the same timestamp/name into one subfolder |
| 🧹 Deduplication | Flag or skip files that are identical in hash or content |

**6. 🖥️ User Interface & UX**

* **Single-window utility app**
* Simple and clean UI with native macOS styling (via Electron)
* UI Components:
  + Folder picker (multi-folder support)
  + List of enabled sorting filters (toggle switches)
  + Conflict resolution options (rename / overwrite)
  + Move vs Copy option (radio buttons)
  + Preview summary (file count per rule)
  + Sort button (disabled until folders and rules are selected)
  + Undo Sort button (enabled after any sort)

**7. ⚙️ File Handling & Options**

* **Conflict Handling**:
  + Before starting sort, if duplicates exist, prompt:
    - “Rename duplicates”
    - “Overwrite existing files”
* **Move or Copy**:
  + User can choose whether to:
    - Move files (default)
    - Copy files (preserve originals)
* **Log File**:
  + App generates a .log or .csv file with:
    - Original filename
    - Source path
    - Destination folder
    - Status (Moved, Skipped, Renamed)

**8. ✅ Sorting Confirmation & Feedback**

* **Visual progress bar** while sorting
* **Popup at end**:
  + "Sorting Complete! 326 files organized into 5 folders."
* **Success/failure message** after each major operation
* **Undo Sort** button:
  + Reverts the last sort operation using log records

**9. 🔐 Permissions & Safety**

* All actions are local — no internet or cloud dependency
* No deletion of files in Phase 1 (only move or copy)
* Undo history stored temporarily for one session (Phase 1)

**10. 📈 Future Enhancements (Phase 2+)**

* AI-based image tagging and smart albums (e.g., "Food", "People", "Outdoors")
* Face detection and clustering
* Smart cleanup (e.g., detect WhatsApp duplicates, blurry photos)
* Drag-and-drop support
* Cloud backup support (iCloud, Google Drive, Dropbox)
* CLI version for automation

**11. 📦 Development Stack Summary**

| **Layer** | **Tech / Tool** |
| --- | --- |
| Frontend | Electron (JavaScript/React) |
| Backend | Python (via child\_process) |
| Packaging | pyinstaller, electron-builder |
| Python Libs | Pillow, hachoir, exifread, shutil, os, tkinter |
| Metadata | EXIF, GPS, file type analysis |
| Versioning | Git + GitHub |

**12. 🗓️ Milestones**

| **Phase** | **Deliverable** | **ETA** |
| --- | --- | --- |
| 1 | Basic working Electron+Python bridge | Day 1 |
| 2 | Folder selection + filter toggle UI | Day 2 |
| 3 | File sorting logic integration | Day 3–4 |
| 4 | Logging + Undo + Progress bar | Day 5 |
| 5 | Final UI polish + .app build | Day 6–7 |

Would you like me to now scaffold the folder structure for the Electron + Python app and start building out the core functionality (Electron boilerplate + Python IPC)?