Day 1 — Setting up base files

- Set up watcher.py, clipboard_monitor.py and index_generator.py

Day 2 — OCR Improvements & Metadata Extraction

- Enhance watcher.py to extract:
    - Timestamps (already done)
    - File source (screenshot or clipboard)
    - Optional topic/tags from filename or user input
- Update clipboard_monitor.py to include similar metadata
- Start testing generate_index.py to include **metadata display**

Day 3 — Topic-based Folder Structure

- Inside data/notes/, create subfolders by topic (e.g., Python/, Math/)
- Allow watcher to **assign notes to topics automatically** (based on tags or initial prompts)
- Update generate_index.py to recognize folders as **sections on the website**
- Optional: Add basic CSS for sections

Day 4 — Automatic Index Updates

- Modify generate_index.py to:
    - Detect new notes automatically
    - Sort notes **chronologically**
    - Include **topic sections** and **metadata** (source, timestamp)
- Test local generation → site/index.html

Day 5 — GitHub Pages Integration

- Configure repository for GitHub Pages (site/ as root)
- Test manual push: commit site/ → push → site available online
- Update README with live site URL

Day 6 — Improved Note Display

- Add:
    - Titles (from filename or first line of note)
    - Snippets preview in the index page
    - "Read more" links to full note markdown rendered in HTML

Day 7 — Markdown to HTML Converter

- Integrate Python markdown library (markdown2 or markdown)
- Convert .md notes to .html pages automatically
- Ensure links from index point to generated HTML pages

Day 8 — CSS Styling & Theme

- Add custom **CSS template** for a clean, readable layout
- Responsive design for mobile access
- Optional: Light / dark mode toggle

Day 9 — Automatic Git Commit & Push

- Improve git_ops.py:
    - Auto-stage new notes and generated HTML files
    - Auto-commit with timestamped message
    - Auto-push to GitHub Pages branch
- Optional: CLI flag to run all steps (watcher + generate_index + push)

Day 10 — Custom Sections & Filtering

- Allow **user-defined sections** inside data/notes/
  e.g., Math, Python, Psychology
- Index page shows separate **cards or lists per section**
- Implement basic filtering by section/topic

Day 11 — Search Functionality

- Add **client-side search** using JavaScript
- Search by note title, content snippet, or tags
- Optional: Highlight search terms in results

Day 12 — Note Metadata Display

- Include metadata on each note page:
  - Timestamp
  - Source (screenshot or clipboard)
  - Topic / Section
- Optional: Word count, character count, or reading time

Day 13 — Multi-Device Optimization

- Ensure responsive layout for mobile, tablet, and desktop
- Test on multiple browsers
- Fix CSS glitches or overflow issues

Day 14 — Tagging System

- Implement **tags** inside notes (e.g., #Python, #Mechanics)
- Show tags on index and note pages
- Filter by tags on website

Day 15 — Analytics & Note Tracking

- Optional: Track:
  - Number of notes per topic
  - Most active topics
  - Daily/weekly note generation
- Display **analytics dashboard** on website

Day 16 — Automated Index + Notes Deployment

- Finalize a **single command / script**:
  - Watcher running
  - Notes generated
  - HTML generated
  - Git commit & push
  - Live site updated automatically

Day 17 — Backup & Versioning

- Keep historical versions of notes
- Optional: Git tags for major updates
- Implement local backup of old notes

Day 18 — Advanced Search / Sorting

- Client-side search by:
    - Date range
    - Section / topic
    - Keyword / tag
- Sort notes in index:
    - Newest first
    - Oldest first
    - Alphabetical

Day 19 — UX Improvements

- Navigation bar
- Breadcrumbs per section
- Dark / light mode toggle
- Optional: Collapsible sections or accordion-style topics

Day 20 — Full Production-Ready Deployment

- Test **all scripts together**
- Ensure:
    - Watcher + Clipboard monitor running
    - Notes converted to HTML
    - Index page updated with sections, tags, dates
    - Automatic commit & push to GitHub Pages
- Final polish: CSS, responsiveness, and usability
- **ENVISAGE fully functional and live online**