

TASK: Repo Cleanup & Documentation Structuring

v1

Goal:

Organize the repository into a clean, professional structure separating:

- the **three product applications**
- the **swarm / experimental system**
- the **infrastructure / platform**
- **data, exports, and third-party libraries**

and ensure each of the three apps has basic documentation in place.

0. GLOBAL RULES (DO NOT SKIP)

1. Never commit secrets

2. Do **not** commit:

- `.env`, `.env.*`
- API keys, Notion tokens, DB URLs
- `.notion_db_id`

3. If you see obvious secrets in tracked files, **stop and add to a report** for the human.

4. Do not delete anything irreversible

5. Use `git mv` to preserve history.

6. Delete only:

- caches (`__pycache__`, `node_modules/`)
- regenerable temp files

7. Do not modify third-party repos

8. Especially: `react-notion-x-master/`

9. Move it to `third_party/` but do not edit internal code.

10. Explain each non-trivial change

11. Use `platform/cleanup/CHANGE_LOG.md`

12. Assume GitHub Codespaces for paths, environment, and commands.

1. TARGET DIRECTORY STRUCTURE

```
/ (repo root)
|
├── product/
│   ├── backend/
│   ├── frontend/
│   ├── docs/
│   │   ├── researcher-dashboard/
│   │   ├── team-lead-tool/
│   │   └── onboarding-app/
└── tests/
|
├── swarm-lab/
│   ├── agents/
│   ├── scripts/
│   ├── prompts/
│   ├── history/
└── docs/
|
├── platform/
│   ├── devcontainer/
│   ├── ci-cd/
│   ├── cleanup/
└── infra/
|
├── data/
|
├── third_party/
└── react-notion-x-master/
|
└── README.md
```

2. REASONING PROTOCOL (APPLY BEFORE MOVING ANY FILE)

For every file/folder, decide:

1. **Category:** product, swarm, platform, data, third_party, or junk.
2. **Intended owner:** researcher dashboard, PM tool, onboarding app, agents, or infra.
3. **Ambiguity:** If uncertain → put in shared docs or leave in place + log in `OPEN_QUESTIONS.md`.
4. **Check imports:** Don't break Python or React imports silently.
5. **Minimal safe change:** Prefer small, reversible steps.

3. PHASE 1 — SCAN & MAP

1. Generate initial tree:

```
ls -R > platform/cleanup/INITIAL_TREE.txt
```

2. Create cleanup tracking files:

```
mkdir -p platform/cleanup
touch platform/cleanup/CHANGE_LOG.md
touch platform/cleanup/OPEN_QUESTIONS.md
```

3. For each top-level folder, document:

```
## Folder: <path>
- Contains:
- Category guess:
- Confidence:
- Notes:
```

4. PHASE 2 — CREATE TARGET SKELETON

Create missing structure:

```
mkdir -p product/backend
mkdir -p product/frontend
mkdir -p product/docs/researcher-dashboard
mkdir -p product/docs/team-lead-tool
mkdir -p product/docs/onboarding-app
mkdir -p swarm-lab/{agents,scripts,prompts,history,docs}
mkdir -p platform/{devcontainer,ci-cd,infra,cleanup}
mkdir -p data
mkdir -p third_party
```

Move infra:

```
git mv .devcontainer platform/devcontainer
```

```
git mv .github/workflows platform/ci-cd
```

Log all moves.

5. PHASE 3 — CLASSIFY & MOVE MAJOR BLOCKS

Onboarding App

Move student-facing LMS code & docs:

```
git mv apps/onboarding-lms/frontend-react product/frontend/onboarding-app
```

Move LMS docs:

```
git mv docs/lms/* product/docs/onboarding-app/
```

Add missing templates if needed.

Team Lead PM Tool

Move PM-related docs:

```
git mv docs/*PM* product/docs/team-lead-tool/
```

Create:

```
mkdir -p product/frontend/team-lead-tool  
mkdir -p product/backend/team-lead-tool
```

Researcher Dashboard

Move analytics/insight docs:

```
git mv docs/*analytics* product/docs/researcher-dashboard/
```

Create app folders.

Log all uncertainties.

6. PHASE 4 — SWARM / AGENT ISOLATION

Move swarm docs:

```
git mv AGENT_TEAM_CHAT.md swarm-lab/docs/
```

Move swarm scripts:

```
git mv scripts/*agent* swarm-lab/scripts/
```

Move history:

```
git mv agent_work_queues swarm-lab/history/
```

If unsure, log in [OPEN_QUESTIONS.md](#).

7. PHASE 5 — DATA, EXPORTS, ARCHIVES

Move archives:

```
mkdir -p data/archives  
git mv ascent_basecamp_core_full_docs.zip data/archives/
```

Move logs:

```
mkdir -p data/logs  
git mv import_log.txt data/logs/
```

Ensure [.gitignore](#) covers: - [data/tmp/](#) - large archives - caches

8. PHASE 6 — THIRD-PARTY ISOLATION

Move:

```
git mv react-notion-x-master third_party/react-notion-x-master
```

Add [third_party/README.md](#).

9. PHASE 7 — PRODUCT DOC TEMPLATES

Ensure each folder contains: - `OVERVIEW.md` - `ARCHITECTURE.md` - `UX-FLOW.md` - `API.md`

If missing → create using the provided templates.

If duplicates/conflicts → preserve both, annotate with TODO.

10. PHASE 8 — FINAL CHECKS

1. `git status` → review additions.
 2. Look for accidentally tracked secrets.
 3. Update `.gitignore` as needed.
 4. Validate doc paths in existing READMEs.
 5. Log all decisions in `CHANGE_LOG.md`.
-

11. OUTLIER HANDLING

Mixed-purpose docs

→ Move to `product/docs/shared/` or leave + log.

Old drafts

→ Keep, annotate, log.

Potential secrets

→ Stop, log; do not modify or scrub.

Unclear items

→ Leave in place + add to `OPEN_QUESTIONS.md`.

This task defines the complete cleanup mission. Execute in phases, log meticulously, and prioritize safety and clarity over speed.