

Agent Setup and Repository Clean-up (Dec 2025 update)

This document merges the findings from the **Repository State and Comparison** report with the latest environment guidance. Follow these tasks in your IDE (VS Code with gpt-5.1-Codex) to tidy your repository, verify environment configuration and prepare for agent integration.

1. Repository review and actions

The analysis shows significant duplication and some legacy artifacts:

- **Duplicate tree:** The parent folder `InsightfulAffiliate_NextGenCopyAI` mirrors the active Git repo under `CURRENT/`. Everything outside `CURRENT/` is an archive or backup. Work exclusively in `CURRENT/` and avoid running scripts from the parent directory. Quarantine or ignore files in `Archive_ready_to_sync/` and `.rootbak` backups **【Repo】**.
- **Stray files:** Remove any files named `eval "$(ssh-agent -s)"` and `.rootbak` backups in the parent root. These are leftovers from old shell commands and pollute searches **【Repo】**.
- **Duplicate assets:** Choose the canonical CSS and manifest from `website_code_block_ORGANIZED/assets/ngcai.css` and `website_code_block_ORGANIZED/site.webmanifest`. Archive or delete older variants in `landing_pages/assest_2/` to prevent confusion **【Repo】**.
- **Prompt duplication:** There are multiple copies of prompt files (e.g., `prompts/rewrite_to_house_style.txt` appears twice). Consolidate prompts into a single `prompts/` folder. Move Prompt-to-Profit templates into a subfolder with a README explaining their use **【Repo】**.
- **Legacy scripts:** The older `scripts/agent_codex.py.bak` and bootstrap script `init_git_url.sh` are for historical reference. Move them into an `archive/` folder or delete them to avoid accidental execution **【Repo】**.
- **Non-ASCII filenames:** Files like `copywriting/product_pages/≡fo¥...csv` may cause encoding issues. Rename them to ASCII-only names and update references accordingly **【Repo】**.
- **Missing requirements file:** There is no `requirements.txt`. Create one listing all Python packages used by your scripts (e.g., `openai`, `tiktoken`) **【Repo】**.
- **Old metadata:** Some Google Docs or HTML headers reference the old `golemmea@gmail.com`. Update these to your current brand email (`insightfulaf`) and correct any meta tags or script references **【Repo】**.

2. Environment variables check

The current suggested environment variables still suffice:

- `OPENAI_API_KEY` – required when running `agent_codex.py` with `--provider openai`. Keep this in your shell profile or Codespaces secrets.
- `AI_DEFAULT_MODEL` – optional default model (e.g., `gpt-3.5-turbo`); use to avoid specifying `--model` each run.
- `AI_FALLBACK_MODEL` – optional cheaper fallback model for error handling.

The optional variables `AI_PROVIDER` and `AI_ENABLE_PREVIEW` remain valid; set them only if you plan to use BYOK providers or preview features ¹. No new environment variables are required based on the repository or VS Code changes.

3. IDE automation tasks (prompt for gpt-5.1 Codex)

Use the following prompt with gpt-5.1 Codex in VS Code Chat to automate the cleanup and setup. Adjust paths if your environment differs. **Important:** ask for confirmation before deleting any files or directories.

```
You are an automation assistant for the repository at `~/Dropbox/
InsightfulAffiliate_NextGenCopyAI/CURRENT`. Perform the following tasks:

1. Clean duplicates and archive legacy items
  - Delete `.rootbak` files and any `eval "$(ssh-agent -s)"` scripts in the
    parent directory (outside `CURRENT/`).
  - Remove the `landing_pages/asest_2/` folder or move it to an `archive/`
    folder; we will use `website_code_block_ORGANIZED` as the canonical source for
    CSS and manifest files.
  - Consolidate prompts: keep one copy of `prompts/rewrite_to_house_style.txt`
    in `prompts/` and delete duplicates. Move Prompt-to-Profit templates into
    `prompts/Prompt_to_Profit/` with a `README.md` describing each template.
  - Move `scripts/agent_codex.py.bak` and `init_git_url.sh` into an `archive/`
    folder. Ensure only `scripts/agent_codex.py` remains in the active scripts
    directory.
  - Rename any files with non-ASCII names (e.g., under `copywriting/
    product_pages/`) to ASCII-only names. Update references in code or docs
    accordingly.

2. Create supporting files
  - Create a `requirements.txt` in the repo root listing all Python
    dependencies: `openai`, `tiktoken` (if used), and any others imported in
    `scripts/`.
  - If not already present, create `.env.example` containing placeholders for
    `OPENAI_API_KEY`, `AI_DEFAULT_MODEL`, `AI_FALLBACK_MODEL`, `AI_PROVIDER`, and
    `AI_ENABLE_PREVIEW`. Do not commit actual secrets.
  - Ensure that `ia-ngcai-main.code-workspace`, `.vscode/mcp.json`,
```

`devcontainer.json` and `codex_scope.txt` exist as per the latest environment plan. If missing, generate them based on the provided specs.

3. ****Update documentation****

- Search all Markdown, HTML and Google Doc metadata for `golemmea@gmail.com` and replace with your current brand email. Update any `` tags or script `data-email` attributes.
- Update references to CSS and manifest files in HTML/Markdown to use `website_code_block_ORGANIZED/assets/ngcai.css` and `website_code_block_ORGANIZED/site.webmanifest`.

4. ****Install and test****

- Create or activate a Python virtual environment in the repo root. Run `pip install -r requirements.txt`.
- Verify that `OPENAI_API_KEY` and optional AI variables are set in your shell or Codespaces secrets.
- Run `scripts/agent_codex.py` with `--provider echo` and `--dry-run` to ensure it reads from the correct files and respects `codex_scope.txt`. Do not commit changes during this test.
- Run the VS Code command `GitHub Copilot: Build remote index` after saving your workspace files to generate a search index for the repository ².

5. ****Commit and push****

- Stage the deleted and created files. Commit with a message such as `Cleanup duplicates, consolidate prompts, add requirements and environment config`. Push to `origin/main`.

Provide summaries of each operation. Always ask for confirmation before deleting user data or renaming files.

4. After completion

Once these tasks are executed and committed, your repository will be free of duplicates and legacy artifacts. You will have a reproducible environment (`requirements.txt`, `.env.example`) and canonical assets. The VS Code environment will be standardized across local and remote sessions, enabling efficient agent integration and deployment.

¹ Expanding Model Choice in VS Code with Bring Your Own Key

<https://code.visualstudio.com/blogs/2025/10/22/bring-your-own-key>

² March 2025 (version 1.99)

https://code.visualstudio.com/updates/v1_99