Real Estate AI Work Systems Course

Hands-on training for ChatGPT, Codex, skills, listing workflows, approval dashboards, harnesses, and self-improvement.

Mission List

01. From ChatGPT User To AI Worker - Starter
02. Spot Repetitive Real-Estate Work - Starter
03. Open Codex And Run Your First Workspace Task - Codex
04. Improve One Listing With Evidence - Codex
05. Create Your Real-Estate Project Rules With AGENTS.md - Codex
06. Turn The Listing Workflow Into A Skill - Skills
07. Batch Process 5, 10, And 50 Listings - Skills
08. Build A Listing Approval Dashboard - Apps
09. Add Quality Harnesses And Gold Examples - Quality
10. Make The Workflow Improve Over Time - Improvement

01. From ChatGPT User To AI Worker

Starter | 35 minutes

Most people use AI as a clever answer box. In this mission, the learner experiences the difference between asking for help and delegating a small piece of work with a result, quality bar, and next action.

Do This Now

- Open ChatGPT in one tab and this course in another.
- Copy the starter prompt below into ChatGPT.
- Answer the questions it asks using a real or imagined real-estate task.
- Save the best answer into your notes box on this page.

Where To Paste This

Paste this into ChatGPT, not Codex. This first exercise is about thinking clearly before you open a workspace.

Prompt

I use AI mostly for questions and quick drafts. I want to start using it as a worker.

Ask me 6 practical questions to identify one repetitive real-estate task that could become an AI workflow.
Focus on listings, buyer follow-up, seller updates, neighborhood descriptions, social posts, lead triage, or market summaries.

After I answer, produce:
1. the task name,
2. the inputs it needs,
3. the output it should create,
4. what a good result looks like,
5. what should still require human approval,
6. whether this should stay a chat prompt, become a Codex workspace task, or become a skill.

What You Should See

ChatGPT should interview you, then turn your answers into a short workflow candidate. The output should name a specific repeated task, not a vague goal like 'do marketing better'.

If It Goes Wrong

- If it gives a generic answer, say: 'Ask me the questions one at a time and wait for my answers.'
- If it tries to automate too much, say: 'Make the first workflow smaller and safer.'
- If it ignores real estate, say: 'Use property listings and client follow-up examples only.'

Practical Output

A named workflow candidate and a first understanding of what should become agentic work.

Codex task card	Details
Files to inspect	None yet.
Expected files	None yet.

Codex task card	Details
Verify	You have one concrete workflow candidate with inputs, outputs, and approval boundaries.
Follow-up	Make this workflow smaller. I want something I can test in under 30 minutes with one listing.

02. Spot Repetitive Real-Estate Work

Starter | 45 minutes

Agents are most useful when the work repeats. This mission turns a vague pile of tasks into a ranked list of workflow candidates.

Do This Now

- List 10 real-estate tasks you do more than once.
- Score each one for frequency, pain, risk, and ease of checking.
- Choose one low-risk workflow to automate first.
- Download your workflow card as Markdown.

Where To Paste This

Paste this into ChatGPT if you are brainstorming. Paste it into Codex only if you already have files or examples in a folder.

Prompt

```
Help me rank real-estate tasks for AI automation.

Here are tasks I repeat:
[paste 5-10 tasks]

Create a table with:
- task
- frequency
- pain level
- risk if AI gets it wrong
- ease of checking
- best first AI step
- recommended surface: ChatGPT, Codex workspace, skill, or harness

Recommend the safest high-value task to start with and explain why.
```

What You Should See

You should see a ranked table and one recommended first workflow. The best first workflow should be repeated, easy to inspect, and not risky to publish without approval.

If It Goes Wrong

- If every task is marked high priority, ask it to choose only one first workflow.
- If it recommends publishing automatically, remind it that human approval is required.
- If the task is too broad, ask for a version that uses one listing and one output.

Practical Output

A workflow card for the first real-estate task to move into Codex.

Codex task card	Details
Files to inspect	Later: starter-workspace/examples and starter-workspace/inputs.
Expected files	A workflow card if you ask Codex to save one.
Verify	The chosen workflow has a clear input and a clear output.
Follow-up	Turn the recommended workflow into a one-page workflow card with trigger, inputs, steps, outputs, checks, and

03. Open Codex And Run Your First Workspace Task

Codex | 60 minutes

Now the learner moves from conversation into a workspace. Codex can inspect files, produce reports, and create artifacts in folders.

Do This Now

- Download and unzip the starter workspace from the course.
- Open Codex and point it at the starter workspace folder.
- Ask Codex to inspect the sample listings and explain the folder structure.
- Save the summary Codex produces in your notes.

Where To Paste This

Paste this into Codex after opening the starter workspace folder.

Prompt

```
Inspect this real-estate starter workspace before making changes.

Please:
1. List the folders and what each one is for.
2. Read AGENTS.md and summarize the working rules.
3. Inspect inputs/sample-listings-10.csv and examples/good-listing.md.
4. Tell me what task this workspace is designed to practice.
5. Do not edit files yet.

Return a short beginner-friendly summary and a suggested next task.
```

What You Should See

Codex should mention inputs, examples, rubrics, skills, and reports. It should not edit files yet. It should suggest improving or auditing listings as the next task.

If It Goes Wrong

- If Codex says it cannot see files, check that you opened the correct starter-workspace folder.
- If it starts editing immediately, stop it and rerun the prompt with 'inspect only'.
- If it ignores AGENTS.md, ask: 'Which instruction files did you load?'

Practical Output

A working Codex project folder and a first file-inspection summary.

Codex task card	Details
Files to inspect	AGENTS.md, inputs/sample-listings-10.csv, examples/good-listing.md.
Expected files	No files in this mission. Inspection only.

Codex task card	Details
Verify	Codex names the folder structure and working rules accurately.
Follow-up	Now create reports/first-workspace-inspection.md with the folder summary and recommended next task.

04. Improve One Listing With Evidence

Codex | 70 minutes

The first useful win is improving one weak listing and explaining why the rewrite is better. The point is not just nicer copy. The point is evidence-backed improvement.

Do This Now

- Ask Codex to choose one weak listing from the sample CSV.
- Have it compare the listing to the good and bad examples.
- Have it create one rewritten listing plus quality notes.
- Review whether the rewrite is specific, accurate, and safe.

Where To Paste This

Paste this into Codex inside the starter workspace.

Prompt

```
Use the real-estate listing improver workflow manually on one listing.

Inspect:
- inputs/sample-listings-10.csv
- examples/good-listing.md
- examples/bad-listing.md
- rubrics/listing-quality-rubric.md

Choose one weak listing and create reports/one-listing-improvement.md with:
1. listing id and original text,
2. quality score before,
3. specific problems,
4. improved listing copy,
5. quality score after,
6. what facts must be verified before publishing,
7. why the rewrite is better.

Do not invent property facts. If a detail is missing, mark it as needs verification.
```

What You Should See

Codex should create a report for one listing. The rewrite should improve buyer positioning and specificity without inventing facts.

If It Goes Wrong

- If it invents amenities, say: 'Rewrite using only facts present in the input.'
- If it writes generic luxury copy, say: 'Explain the target buyer and strongest concrete viewing moment.'
- If it does not create a file, ask it to save the report under reports/one-listing-improvement.md.

Practical Output

A first listing-improvement report saved by Codex.

Codex task card	Details
Files to inspect	inputs/sample-listings-10.csv, examples, rubrics.
Expected files	reports/one-listing-improvement.md.
Verify	The report includes original, problems, rewrite, score before/after, and verification warnings.
Follow-up	Now rewrite the same listing in three tones: luxury buyer, investor, and family buyer. Keep facts unchanged.

05. Create Your Real-Estate Project Rules With AGENTS.md

Codex | 60 minutes

AGENTS.md teaches Codex durable project rules. In real estate, those rules protect accuracy, privacy, brand voice, and approval boundaries.

Do This Now

- Read the starter AGENTS.md.
- Ask Codex to propose changes for your market and business.
- Review the proposed rules before accepting them.
- Keep the file short and operational.

Where To Paste This

Paste this into Codex inside the starter workspace.

Prompt

```
Read AGENTS.md and propose a safer, more specific version for my real-estate business.

Business context:
- Market: [city/region]
- Property type: [sales/rentals/luxury/investment/commercial]
- Brand voice: [describe voice]
- Legal/compliance sensitivities: [fair housing, MLS, privacy, local requirements, etc.]

Do not edit the file yet. First return:
1. rules you would keep,
2. rules you would add,
3. rules that may be too vague,
4. a revised AGENTS.md draft,
5. questions I should answer before we apply it.
```

What You Should See

Codex should produce a revised AGENTS.md draft and questions. It should not edit until you approve.

If It Goes Wrong

- If the draft is too long, ask for a version under 700 words.
- If it includes legal claims, ask it to convert them into review warnings.
- If it is generic, add your market, property types, and brand examples.

Practical Output

A project instruction file that future Codex tasks will follow.

Codex task card	Details
Files to inspect	AGENTS.md.
Expected files	Optionally updates AGENTS.md after approval.
Verify	The file gives durable rules without becoming a giant training manual.
Follow-up	Apply the revised AGENTS.md exactly as drafted, then summarize what changed.

06. Turn The Listing Workflow Into A Skill

Skills | 90 minutes

A skill is how the learner stops rewriting the same prompt. The starter workspace already includes a first skill draft for listing improvement.

Do This Now

- Ask Codex to read the existing listing improver skill.
- Have it compare the skill against the one-listing report.
- Improve the skill so it captures what worked.
- Run the skill again on a second listing.

Where To Paste This

Paste this into Codex inside the starter workspace.

Prompt

```
Use the starter skill at skills/real-estate-listing-improver/SKILL.md.

First inspect:
- skills/real-estate-listing-improver/SKILL.md
- reports/one-listing-improvement.md if it exists
- rubrics/listing-quality-rubric.md
- examples/good-listing.md
- examples/bad-listing.md

Then:
1. Explain what the skill currently teaches Codex to do.
2. Identify what is missing based on the one-listing report.
3. Propose a tighter SKILL.md revision.
4. Do not apply changes until I approve.
5. After approval, test the revised skill on a different listing and create reports/skill-test-listing.md.
```

What You Should See

Codex should treat the skill as reusable workflow instructions, not just a prompt. It should propose a revision and then test it only after approval.

If It Goes Wrong

- If it rewrites the skill too broadly, ask for the smallest useful change.
- If it puts lots of examples into SKILL.md, ask it to move examples into references or examples instead.
- If the second test is worse, ask it to compare against the first report and diagnose the regression.

Practical Output

A reusable listing-improver skill and a second test report.

Codex task card	Details
Files to inspect	skills/real-estate-listing-improver/SKILL.md, rubrics, examples, reports.
Expected files	Updated skill after approval and reports/skill-test-listing.md.
Verify	The second run should be easier, more structured, and closer to your quality bar.
Follow-up	Turn the lessons from reports/skill-test-listing.md into one specific improvement to the skill and one regression c

07. Batch Process 5, 10, And 50 Listings

Skills | 90 minutes

This is where the learner feels the leverage. The same workflow that improves one listing can evaluate and rewrite many, but only if the output is structured for review.

Do This Now

- Run the skill on 5 listings first.
- Review the report and fix any pattern problems.
- Run it on 10 listings.
- Only then run it on the 50-listing JSON sample.

Where To Paste This

Paste this into Codex inside the starter workspace.

Prompt

```
Run the real-estate listing improver skill as a batch workflow.

Use:
- inputs/sample-listings-10.csv for the first batch
- skills/real-estate-listing-improver/SKILL.md
- rubrics/listing-quality-rubric.md

Step 1: Process only 5 listings and create reports/batch-5-listings.md.
For each listing include:
- id
- score before
- top 2 problems
- suggested rewrite
- score after
- verification warnings
- approve/review/reject recommendation

Stop after 5 and summarize repeated issues before processing more.
```

What You Should See

Codex should stop after 5 listings and summarize patterns. This prevents scaling a bad workflow across 50 items.

If It Goes Wrong

- If it processes all 50 immediately, remind it to stop after 5.
- If outputs are hard to review, ask for a table with one row per listing.
- If rewrites feel samey, ask for target-buyer and strongest-feature variation.

Practical Output

Batch report for 5 listings and a safe path to 10 and 50.

Codex task card	Details
Files to inspect	inputs/sample-listings-10.csv, inputs/sample-listings-50.json, skill, rubric.
Expected files	reports/batch-5-listings.md, later reports/batch-10-listings.md and reports/batch-50-listings.md.
Verify	The workflow stops for review before scaling.
Follow-up	The 5-listing batch looks acceptable. Now process all 10 CSV listings and create reports/batch-10-listings.md u

08. Build A Listing Approval Dashboard

Apps | 90 minutes

Batch work needs a review interface. The learner should approve, reject, or request changes instead of hunting through a giant markdown file.

Do This Now

- Open the built-in demo dashboard from this course.
- Approve and reject a few sample listing rewrites.
- Export your decisions as Markdown or JSON.
- Ask Codex to generate a similar dashboard from the batch report.

Where To Paste This

First use the dashboard in this browser. Then paste the prompt into Codex.

Prompt

```
Create a static HTML listing approval dashboard from my batch listing report.

Inspect:
- reports/batch-10-listings.md
- inputs/sample-listings-10.csv

Create:
- reports/listing-approval-dashboard.html

The dashboard should:
1. show original and suggested listing copy side by side,
2. show quality score before/after,
3. show verification warnings,
4. let me mark approve, revise, or reject,
5. store decisions in browser localStorage,
6. export decisions as JSON and Markdown.

Keep it static. No backend, no login, no external dependencies.
```

What You Should See

Codex should create a standalone HTML dashboard. Opening it should show listings with approval buttons and export controls.

If It Goes Wrong

- If the dashboard is blank, ask Codex to inline the data instead of fetching local files.
- If decisions do not persist, ask it to use localStorage.
- If the dashboard is ugly or hard to scan, ask for side-by-side cards and clear status colors.

Practical Output

A review dashboard that turns batch AI output into human-approved decisions.

Codex task card	Details
Files to inspect	reports/batch-10-listings.md, inputs/sample-listings-10.csv.
Expected files	reports/listing-approval-dashboard.html.
Verify	You can approve/reject, refresh, and export decisions.
Follow-up	Improve the dashboard so I can filter by status and score improvement.

09. Add Quality Harnesses And Gold Examples

Quality | 90 minutes

A harness lets the learner stop trusting vibes. It checks whether listing outputs meet a quality bar and catches repeated mistakes.

Do This Now

- Read the listing rubric.
- Create a gold set with good, bad, and borderline listing examples.
- Ask Codex to evaluate 10 listings against the rubric.
- Turn any repeated failure into a regression case.

Where To Paste This

Paste this into Codex inside the starter workspace.

Prompt

```
Create a simple listing quality harness.

Inspect:
- rubrics/listing-quality-rubric.md
- examples/good-listing.md
- examples/bad-listing.md
- reports/batch-10-listings.md if it exists

Create:
- reports/listing-quality-harness.md
- examples/borderline-listing.md
- reports/listing-regression-cases.md

The harness should check:
1. no invented facts,
2. target buyer is clear,
3. strongest feature is concrete,
4. generic adjectives are reduced,
5. verification warnings are present,
6. rewrite is different enough from the original,
7. human approval is required before publishing.

Evaluate 10 sample listings and name the repeated failure patterns.
```

What You Should See

Codex should create a harness document and identify repeated failure types. It should not claim the workflow is perfect.

If It Goes Wrong

- If it only gives scores, ask for specific evidence and failure types.

- If it approves everything, ask it to apply an adversarial reviewer lens.
- If the rubric is too vague, ask for measurable checks and examples.

Practical Output

A listing quality harness, a borderline example, and first regression cases.

Codex task card	Details
Files to inspect	rubrics, examples, reports.
Expected files	reports/listing-quality-harness.md, examples/borderline-listing.md, reports/listing-regression-cases.md.
Verify	Repeated failures become named cases, not vague advice.
Follow-up	Update the listing improver skill so it explicitly avoids the top two regression failures.

10. Make The Workflow Improve Over Time

Improvement | 90 minutes

The final mission teaches the loop: when the learner corrects the agent, the correction becomes a better example, rubric, skill, or harness check.

Do This Now

- Choose one bad AI rewrite from the batch workflow.
- Write a correction in plain language.
- Ask Codex to turn the correction into a durable update.
- Rerun the workflow on the old failure case.

Where To Paste This

Paste this into Codex after you have at least one batch report or dashboard decision.

Prompt

```
Turn my correction into a durable workflow improvement.

Here is a listing output I did not like:
[paste listing id, original, AI rewrite, and your correction]

Inspect:
- skills/real-estate-listing-improver/SKILL.md
- rubrics/listing-quality-rubric.md
- examples/good-listing.md
- examples/bad-listing.md
- reports/listing-regression-cases.md if it exists

Return:
1. failure type,
2. why it matters,
3. whether this should become a gold example, rubric update, skill update, or regression case,
4. the exact proposed change,
5. a rerun prompt to test whether the mistake is fixed.

Do not apply changes until I approve.
```

What You Should See

Codex should classify the failure and propose a small durable update. The correction should not stay trapped in the chat.

If It Goes Wrong

- If it proposes a giant skill rewrite, ask for the smallest change that prevents this exact failure.
- If it blames missing data, ask what verification warning should have been shown.
- If it cannot test the fix, ask for a regression case.

Practical Output

A self-improving workflow where mistakes become reusable standards.

Codex task card	Details
Files to inspect	skill, rubric, examples, reports.
Expected files	A proposed durable update and rerun prompt.
Verify	The same failure is caught or avoided on the next run.
Follow-up	Apply the approved change and rerun the old failure case. Report whether the mistake is fixed.

Sources

OpenAI Codex Manual: https://developers.openai.com/codex/

Codex surfaces, workflows, customization, skills, AGENTS.md, MCP, subagents, automations, and safety.

Codex Agent Skills: https://developers.openai.com/codex/skills

Skills as reusable workflows with progressive disclosure, descriptions, references, and optional scripts.

Codex AGENTS.md: https://developers.openai.com/codex/guides/agents-md

Durable instruction files for global, project, and nested project guidance.

Codex MCP: https://developers.openai.com/codex/mcp

Model Context Protocol for connecting Codex to tools and context.

Codex Subagents: https://developers.openai.com/codex/subagents

Parallel specialized agents and custom agent workflows.

OpenAI Agents SDK: https://developers.openai.com/api/docs/guides/agents

Agent workflows, tools, handoffs, guardrails, and tracing.

OpenAI Evals: https://developers.openai.com/api/docs/guides/evals

Evaluation loops for measuring model and workflow quality.

OpenAI Graders: https://developers.openai.com/api/docs/guides/graders

Structured scoring and model-graded evaluation patterns.