Enkai

Ask for features in Slack. Review PRs in GitHub.

A new way to build and evolve enterprise-ready apps.

The problem

Teams don't struggle to *want* features.
They struggle to *ship* them consistently.

- too much coordination
- too many handoffs
- not enough sustained execution
- quality drops when speed increases

The shift

Most teams already have:

- Slack for requests
- GitHub for source of truth
- PRs for review and safety

Enkai connects them into a single, repeatable shipping loop.

What Enkai is

An app-building system that:

- turns feature requests into GitHub issues
- plans changes in context of the full app
- implements features in isolation
- submits PRs for human review

The experience

1. You request a feature in Slack
2. Enkai converts it into a GitHub issue
3. Enkai plans the change (clearly)
4. Enkai builds and tests it
5. You review a PR and merge

You stay in control. Enkai does the work.

Why it's different

Most "AI dev tools" help you write code faster.

Enkai helps you **ship software** by:

- operating around your repo + workflow
- planning before coding
- batching work intelligently
- delivering reviewable PRs, not chats

Trust & safety (the adoption unlock)

- Nothing merges automatically
- Every change is traceable (issue → plan → PR)
- Works with your existing PR controls:
  - required reviews
  - CI gates
  - CODEOWNERS

What it can build

- product features (small to medium scope)
- internal tools
- integrations
- UX improvements
- refactors with clear PR boundaries

What you get

- consistent throughput
- lower coordination cost
- clean PRs you can audit
- a sustainable "shipping loop" for your product

Next steps

- connect a Next.js repo
- connect Slack
- start with a small set of feature requests
- evaluate PR quality and cycle time

Enkai: software progress, on demand.