

ACCELERATE ON AI • COHORT 2026

Ship something *real*.

A hands-on, outcomes-driven bootcamp to help you build and deploy a personal or community solution — with modern AI tools, clear frameworks, and mentorship.

PROJECT

Personal or community
solution

DELIVERABLES

Public demo · docs · deployed

TRACKS

Ramp · 2-week · 4-week

ON COMPLETION

Stipend · Certificate · AI credits

Made possible *together.*

Skill Forward exists because a coalition of people and institutions put their time, funding, and belief behind the idea that anyone can ship something real with AI. Thank you.

i. Worksystems

For the Dislocated Worker grant that funds stipends, AI credits, and the path back to work for every participant.

ii. PCC

For the partnership, the facilitation, and for opening the program up to a community of learners ready to build.

iii. The participants

Every builder, mentor, and teammate who showed up, picked a problem, and shipped a solution. This program is nothing without you.

Four tracks. Same outcome: *ship something real.*

Pick the commitment that fits your schedule. Each track lands you on a deployed solution, a public demo, and public documentation.

RAMP · R1

5 hr

Proof of Work Portfolio

Apr 27 → Apr 30

START → END

Beginners who want a portfolio artifact quickly.

RAMP · R2

10 hr

Idea to Reality

Apr 29 → May 8

START → END

From use-case to deployed solution, with mentor office hours.

BOOTCAMP · 01

20 hr

2-week AI Bootcamp

Starts May 18

KICKOFF

Fast prototype, quick validation, certificate + stipend.

BOOTCAMP · 02

40 hr

4-week AI Bootcamp

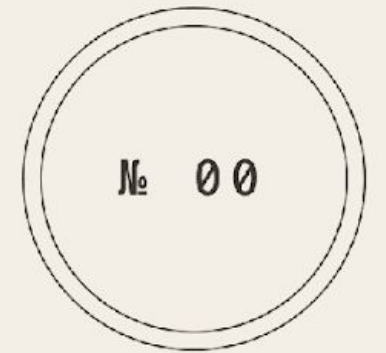
Starts Jun 2

KICKOFF

Deeper build, stronger portfolio asset, full mentoring.

IN
PERSON

All live sessions are in person on these dates — and parking is covered by the Dislocated Worker grant



Pick the *right AI* before you build.

The model, the app, and the harness each do different work. Knowing which layer you're in changes what you should prompt — and what you should expect back.

PART I

Orient • 4 slides

PART II

Build • 6 slides

AUDIENCE

Designers, devs, creatives

READ TIME

~10 minutes

Every AI product sits on *three layers*. Know which you're using.

i. Models

The raw intelligence — the trained network itself. Differences here set the ceiling on what's possible. Picking a weak model means a weak everything downstream.

ii. Apps

The chatbot, the IDE plugin, the image tool — the interface wrapped around a model. Same model, different app, different experience. Choose the app that matches your task.

iii. Harnesses

The scaffolding that lets a model take actions: run code, browse, read files, call tools. Harnesses turn a chat partner into an agent that actually does work.

Three models lead the *frontier*. Use any one of them — not the cheap defaults.

LAB 01

GPT-5.2 Thinking

OpenAI

Strong generalist with deep reasoning mode.
Good first reach for code, writing, and
research.

LAB 02

Claude Opus 4.6

Anthropic

Long-context thinking, careful writing, and
agentic coding. Pairs well with a code
harness.

LAB 03

Gemini 3 Pro

Google DeepMind

Broad knowledge and tool use, strong on
data and search-backed answers.

Rule of thumb: *pay the \$20 a month and use the advanced model. Free tiers default to lighter models and will quietly underdeliver on anything harder than a question.*

The chat box is the *entry point* — not the ceiling.

For real work, switch to a harness or a purpose-built app. The model is the same; the scaffolding around it changes what you can get done.

-
- i.* **Coding harnesses**
Claude Code, Cursor, Codex — the model reads your repo, runs commands, and edits files end-to-end.

 - ii.* **Research & reading**
NotebookLM and Deep Research modes turn a library of sources into a grounded, cited answer.

 - iii.* **Building apps & sites**
Prompt-to-site tools turn a description into a working frontend — exactly what part ii of this deck is about.

 - iv.* **Voice & co-work modes**
Voice, canvas, and shared-doc modes are better than chat for thinking out loud or drafting alongside the model.
-

Design the *look* first — before a single line of code.

These tools move from a sketch or prompt to a real interface you can hand off, fork, or feed into a build tool. Use them to lock the visual direction before generating the site.

TOOL 01

Claude Design

Anthropic

Prompt-driven UI explorations with strong typography and layout instincts. Great for hi-fi variations.

TOOL 02

Google Stitch

Google

Generates screens from prompts or screenshots, with a Figma-style canvas for editing and iterating.

TOOL 03

Figma Make

Figma

In-Figma prompt-to-design workflows that stay inside your existing components and libraries.

TOOL 04

v0 Vercel

Component-level generation tuned for React and Tailwind — good bridge between design and build.

Four *categories* of coding AI — each suited to a different kind of work.

CATEGORY	PURPOSE	TOP TOOLS	STRENGTHS	LIMITATIONS
Vibe Coding Apps	Natural language → app generation	Lovable Replit Bolt	Rapid prototyping; easy for non-devs	Weak for legacy code and mods
CLI Agentic Coding	Terminal-based development automation	Claude Code Codex CLI	Refactoring, bugfixes, multi-file ops	Not for quick app creation
IDE Integrations	AI built into the code editor	Cursor Copilot	Multi-file edits, chat, autocomplete	Locked to a specific IDE
Auto-Debugging	Autonomous code monitoring and fixing	Windsurf Cascade	PR review, bugfixes, continuous quality	Still evolving

Deploy *directly* from the tool that built it.

If you used a vibe-coding app, your fastest path is its own one-click deploy. No git, no infra — a public URL in under a minute.

TOOL 01

Lovable

Click *Publish* in the top-right. You get a free `your-site.lovable.app` URL. Connect a custom domain on a paid plan.

Publish → live URL ✓

TOOL 02

Bolt

Click *Deploy* to ship to Netlify in one tap, or *Connect to GitHub* to keep a copy of the code in your account.

Deploy → Netlify ✓

TOOL 03

Replit

Use the *Deploy* tab — Static, Autoscale, or Reserved. Free static deploys are fine for a portfolio.

Deploy → Static ✓

Save your work to *GitHub*.

GitHub is a cloud home for your code. Every change is versioned, every version is recoverable, and other tools — deploys, AI agents, collaborators — can read from it.

WHY BOTHER

- One source of truth across machines and tools
- Free hosting tier (GitHub Pages)
- Required by Vercel, Netlify, and most agents
- Public profile that doubles as a portfolio signal

> FIRST PUSH

- i. Create a new repo on github.com
- ii. In your project folder, run:

```
$ git init
$ git add .
$ git commit -m "first commit"
$ git remote add origin <repo url>
$ git push -u origin main
```

- iii. Refresh the repo page — your code is online.

From GitHub to a *live URL*. Three options, all free.

Once your code is on GitHub, point a host at the repo. Each push afterwards re-deploys automatically.

HOST 01

GitHub Pages

Settings → Pages. Pick a branch (usually main) and folder. Best for static sites and HTML portfolios.

`user.github.io/repo`

HOST 02

Vercel

vercel.com → *Add New Project* → import the repo. Auto-detects Next, React, Vite. Custom domains and previews per branch.

`your-app.vercel.app`

HOST 03

Netlify

app.netlify.com → *Import from Git*. Set build command and publish directory; Netlify handles the rest on every push.

`your-app.netlify.app`

Once it's on GitHub, an *agent* can take over.

> CLAUDE-CODE · MY-
PORTFOLIO

TERMINAL

```
$ git clone https://github.com/lucas/my-portfolio
$ cd my-portfolio
$ claude
```

> Add a `/case-studies` route with three project pages. Pull copy from `data/projects.json`. Match the existing serif type and grid spacing. Open a PR named `feat/case-studies`.

- Reading repo structure...
- Editing 4 files, creating 3 routes
- Running tests · all green
- Pushing branch · PR opened

```
$ git pull origin feat/case-studies
```

THE LOOP

- Pull**
git clone brings the repo down. Now Codex or Claude Code can read every file.
 - Prompt**
Describe the change in plain English. Reference files, routes, copy sources — be specific.
 - Review & merge**
Agent opens a pull request. You review the diff, merge — and your live host redeploys.
-

FIVE SLIDES / THREE IDEAS / TWO PROMPTS



Build a portfolio *worth sharing.*

TOPIC

Digital portfolios, distilled

FORMAT

3 info · 2 prompts

AUDIENCE

Designers, devs, creatives

READ TIME

~7 minutes

Your portfolio is the *first interview* — before anyone meets you.

PURPOSE

01

Prove your craft with real work, not a list of skills on a résumé.

SIGNAL

02

Presentation quality shapes early judgments — orderly beats ornate.

LEVERAGE

03

A good site compounds: clients share the link, referrals arrive pre-sold.

Four properties every strong portfolio *shares*.

The style varies — minimalist grid, brutalist slab, cinematic reel — but the underlying structure is consistent across categories.

i.

Visually appealing

Clean, modern design with high-quality imagery that matches the caliber of the work itself.

ii.

Intuitive to browse

Simple navigation; visitors find relevant work in seconds, not scrolls.

iii.

Content-focused

Work up front, supporting descriptions behind. The layout recedes; the projects speak.

iv.

Personalized

Reflects your style, skills, and personality — not a template with your name pasted on top.

Pick the *archetype* closest to your field.

Each direction carries its own conventions — galleries, case studies, reels, booking forms. Start from the archetype, then make it yours.

01 Web Developer Interactive UI + blog	02 Creative Director Case studies + reel	03 Developer + Blog GitHub + dark mode
04 Freelancer Testimonials + inquiry form	05 Photographer Hi-res galleries + booking	06 Illustrator Scrolling showcase
07 UI / UX Designer Prototypes + case studies	08 3D Artist Embedded 3D models	09 Motion Designer Video reels + highlights
10 Music Producer Audio samples + booking	11 Filmmaker Short films + awards	12 Personal Brand Keynotes + press
13 E-Sports Player Tournaments + sponsors	14 Model Masonry grid + digitals	15 Interior Designer Full-width slideshows

Paste this, then *edit the name*.

USE-CASE · WEB DEVELOPER / SOFTWARE ENGINEER

```
> PROMPT_01.TXT COPY · PASTE ·  
EDIT
```

Build a **web developer portfolio** for
a developer named **Lucas**.

Include:

- A hero with animated transitions
- An interactive project grid with hover states
- A personal blog with MDX posts
- Dark mode by default, light-mode toggle
- GitHub integration showing recent repos
- A contact form routed to email

Style: minimal, monospace accents,
generous whitespace, motion used
sparingly. ■

TIPS BEFORE YOU SEND IT

-
- i.* Be specific, not clever.
Name the sections, the tone, the interactions. Vague prompts return vague sites.
-
- ii.* Swap the details.
Replace the name, stack, and style words with your own before hitting generate.
-
- iii.* Iterate on sections.
Once the scaffold lands, refine one section at a time — hero, then projects, then blog.
-