deerwalk training center

Course Outline

The Smart Coder: Building Apps with AI Agents

Prompt. Build. Launch: The GenAI App Creator Course

This is a 40-hour Generative AI coding course tailored for 12th-grade students with little to no programming or terminal experience.

The Toolkit: Explaining the Tools to Beginners

We divide the tools into two categories: **The Instant Builders** and **The Local Agents**.

1. The Instant Builders (Web-Based)

- Figma Make (The Designer): You type, "I want a music player app," and it instantly draws the beautiful UI for you. It's for designing how things *look* before you build them.
- Google Stitch (The Prototyper): You type what you want, and it instantly generates a fully working web app right in your browser. It's perfect for testing ideas fast.
- Google AI Studio (The Laboratory): This is where you test how the AI "brain" works. You can tweak prompts and see how the Gemini model responds before you put it into your app.

2. The Local Agents & Plugins (On Your Computer)

- The Terminal & CLI Agents (`gemini-cli`, `claude-code`, `opencode`, `antigravity`): These are your "Robotic Interns." Instead of a web browser, they live on your computer. You tell them what to do, and they actually create the files, write the code, and save them to your folders.
- The `README.md` (The Blueprint): A text file where you write the master instructions for your app. The AI reads this to know exactly what to build.
- `claude-mem` / `save_memory` (The Sticky Note): AI forgets things. This tool teaches the AI to remember permanent rules (e.g., "My app is always Dark Mode").

- The `caveman` Plugin (The Money Saver): AI costs "tokens" for every word it reads/writes. This plugin forces the AI to speak like a caveman ("Fix bug. Red button.") to save you tokens, time, and money.
- The `everything` Plugin (The Rulebook): Gives the AI strict, professional coding standards so it writes good code, even if the student doesn't know what good code looks like.
- The `graphify` Plugin (The Map Maker): As your app gets bigger, it gets confusing. Graphify looks at your files and draws a visual map showing how everything connects.

The 40-Hour Curriculum

Module 1: The "Instant Magic" (8 Hours)

Goal: Hook the students immediately by letting them generate real apps in the browser without writing a single line of code.

- Hour 1-2: Welcome to the AI Era.
- What is Generative AI? How does it understand code?
- Introduction to **Google AI Studio**: Playing with prompts and seeing how the AI responds.
- Hour 3-4: Designing with Words (Figma Make).
- Students use **Figma Make** to generate UI screens from text prompts.
- *Activity:* Generate a design for a "Sneaker Store" app.
- Hour 5-8: Instant App Building with Google Stitch.
- Moving from design to working code.
- Students use **Google Stitch** to type a prompt and instantly get a working React web app in their browser.
- *Activity:* Prompting Stitch to build a working calculator or a simple weather dashboard.

Module 2: Stepping into the Developer Seat (8 Hours)

Goal: Transition from web-based toys to real, local software development using the Terminal and CLI Agents.

- Hour 9-10: Conquering the Terminal.

- Demystifying the black screen. Learning just 3 commands: `ls` (look), `cd` (move), `mkdir` (make folder).
- Hour 11-12: Meet your Robotic Interns.
- Introducing `gemini-cli` and `claude-code`.
- Having the first conversation in the terminal. Asking the AI to create a simple HTML file.
- Hour 13-14: Alternative Brains.
- A brief look at `opencode` and `antigravity` as different ways to power your terminal agent.
- Hour 15-16: README-Driven Development.
- *Crucial Skill:* You aren't writing code; you are writing instructions.
- How to write a perfect `README.md` that tells the AI exactly what the project is, what it looks like, and what it does.

****Module 3: The AI Manager's Toolkit (8 Hours)****

Goal: Teach students how to control the AI, save tokens, and manage project chaos using Plugins.

- Hour 17-18: Giving the AI Memory (`claude-mem`).
- Teaching students how to save project facts so the AI doesn't forget who they are or what the app does.
- Hour 19-20: Saving Tokens with `/caveman`.
- Understanding AI costs.
- Activating the Caveman skill. Comparing a normal, wordy AI response to a fast, cheap Caveman response.
- Hour 21-22: Enforcing Quality with the `everything` Plugin.
- How to make the AI write professional code without knowing how to code yourself.
- Hour 23-24: Visualizing the Matrix (`graphify`).
- Having the AI generate a 3-page website, then running `graphify` to generate a visual map of how the files connect.

****Module 4: Building a Real Web App & A Taste of Mobile (8 Hours)****

Goal: Use the CLI agents to build a local project, and take a very simple look at mobile apps.

- Hour 25-26: Planning the Web App.
- Students write a `README.md` for a simple web app (e.g., a Movie Searcher).
- Hour 27-28: AI Pair Programming (Web).
- Using the CLI agent to read the README and build the HTML/CSS/JS.
- *Debugging:* Learning how to copy browser errors and paste them into the CLI agent to let it fix itself.
- Hour 29-30: What is a Mobile App? (Flutter Intro).
- Explaining Flutter simply: "One code makes both an iPhone and Android app."
- Using the CLI Agent to generate a blank Flutter template.
- Hour 31-32: A Simple Flutter Screen.
- Prompting the AI to build *one* simple screen in Flutter (e.g., a digital business card). Running it in the Chrome browser (no complex phone emulators needed).

Module 5: The Capstone Project & Launch (8 Hours)

Goal: Students act as the "Director" and guide their AI Agent to build a final web app from scratch.

- Hour 33-34: The Blueprint.
- Students use **Figma Make** to visualize their idea, and write a detailed `README.md` to guide their local CLI Agent.
- Hour 35-36: The Build Phase.
- Students set up their Agent's memory, turn on the `caveman` plugin to save tokens, and command the CLI to build the app file-by-file.
- Hour 37-38: Debugging & Mapping.
- Students fix errors by pasting them to the AI.
- They use `graphify` to create a final map of their project.
- Hour 39-40: Demo Day!
- Using a free service (like Netlify Drop) to drag their folder onto the internet.
- Students present their live links and explain the prompts/README they used to command their AI.