[Intro: Setting the Scene]

[Camera pans to you, looking serious with a bunch of code on the screen. Dramatic background music.]

"In a world full of programming languages… from Python to Java, from C++ to… whatever Haskell is supposed to be... one coder dared to ask: Why follow the rules when you can make your own?"

Or

Picture this: Python and Java have their little fan clubs, C++ is for brave souls, and Haskell… well, no one’s quite sure what Haskell is doing, but it’s there. So I thought, why not throw my own language into the mix?

One day, I caught myself typing *yet another* System.out.println and thought, "Why not make a language where I can just *shout*?"—literally.

Or

Ever thought, 'Wow, there just aren’t enough programming languages out there'? Yeah, me neither. But here we are. I decided to make my own anyway.

[Text on screen: "Introducing GuavAI: The Totally Serious New Language"]

Opening Scene: The Idea Moment

[Cut to you sitting at your desk, staring at the screen. Sudden “lightbulb” sound effect, and you look directly into the camera.]

[Screen zooms in on your face with an exaggerated look of inspiration. Text overlay: 'THE GUAVAI ORIGIN STORY']

Now I need a name for this language. Uhhhhhhhhhhhhhh Java. Guava. Yup its going to be called Guava.

"Introducing… GuavAI. A language where `shout()` \*actually\* means `System.out.println()`. It’s like Java, but with a twist. And that twist is me being too lazy to write boilerplate code."

Now whats unique about this language you ask? Well it has AI in it duh. If you are very feeling too lazy to write the code you can just let the AI do that for you. Now, let’s dive into the programming!

Scene 2: Coding the Language

[Cut to you coding with exaggerated focus, typing furiously. Upbeat, quirky background music begins.]

"Now, first things first. We needed a way to \*compile\* GuavAI. So, of course, I thought, ‘Why not use Python?’ Because nothing says solid code like mixing languages that have no business talking to each other!"

"Here’s the deal. Instead of writing `main()` and `System.out.println`, you can now just write `shout()` for your prints, and `main()` becomes this super-short function."

"The best part? I use Google Gemini AI to fill in the gaps. Yeah, like, ‘Hey AI, why don’t you write this function for me?’ Who needs human coding when you can just… outsource it to AI?"

Now, first things first. We needed a way to \*compile\* GuavAI. And I don’t know how to make a compiler. So I am just going to use the Java’s compiler. But we need to parse the .guav file for the compiler to understand. So, of course, I thought, ‘Why not use Python?’ Because nothing says solid code like mixing languages that have no business talking to each other!"

Scene 3: The “Challenges”

[Text on screen: “The Debugging Nightmare Begins…”]

"So, obviously, it worked perfectly the first time… said no coder ever. After a few... \*minor\* hiccups..."

[Cut to you scrolling through pages of error logs, with dramatic sighing.]

"I realized maybe I need to tell my language exactly what each line does. Little details, like 'What type is that variable?' or 'What even \*is\* `shout` in Java?' But who needs types when you have enthusiasm, right?"

[Camera zooms in on an error: `NoneType` Object Has No Attribute...\*\*]

"And then, there was the `NoneType` error. I’m pretty sure it’s Python's way of saying, ‘I give up on you.’”

Now the best part of this language is its AI. So basically how this works is you write a line in this syntax and through an AI model its going to parse and compile that line into java compiler.

"So, obviously, it worked perfectly the first time… said no coder ever. After a few... \*minor\* hiccups..."

[Cut to you scrolling through pages of error logs, with dramatic sighing.]

Now after few weeks of programming I finally have a defined syntax for my language, a parser, compiler and an AI function generator ready but its still not good enough. I need to make this look good.

Scene 4: Creating the Extension

Now I have no idea how to make a vs code extension.

<I type how to make a vs code extension>

After few hours of reading, I can *finally* start making my extension. I also found this tutorial on YouTube by Tommy on how to add custom syntax highlighting for a new language. You can refer to this if you ever wanted to make your own programming language.

Scene 5: The Final Product

[Cut to you sitting proudly in front of your screen with the GuavAI logo on it.]

"And there you have it! A programming language so unique, so innovative, that it practically yells at you. GuavAI: when you’re ready to code with AI that does \*most\* of the work and syntax that basically… \*gets you\*."

[Show a quick demo where `shout(add(a,b));` prints something funny.]

"Now, whenever I need to print, I just shout. Programming has never been easier—or louder."

Ending: The Call to Action

[Screen fades to text: "GuavAI. Because we can."]

"If you want to make your own language and take on the endless debugging adventures, go for it! Just remember, coding is about creativity… and a bit of caffeine-fueled insanity."

[Screen fades to black with text: “Like and Subscribe for more adventures in coding chaos!”]

And there you have it: a funny, relatable journey through the trials and triumphs of making GuavAI!