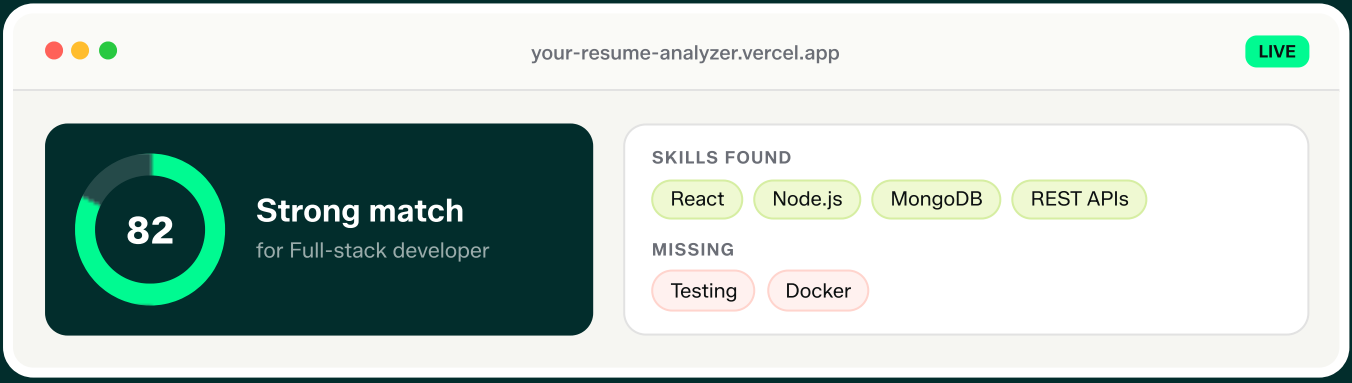
Build your first AI project like a senior engineer. Ship it in 2 days.

A live 2-day workshop in Tamil, with English where it helps. Start from an empty folder, build the screen, the server, the AI and the database, and log off with a link the whole world can open.

₹499 · both days | 2 live days | Tamil + English | Live on Zoom



1.2M+ YouTube subscribers

15,000+ students trained

3,000+ projects built by students

4 years of teaching developers

A LIVE WORKSHOP BY ERROR MAKES CLEVER

01 WHAT YOU'LL BUILD

An AI resume analyzer.
Built by you, layer by layer.

Someone uploads their resume. Your app reads it, scores it for the job they want, finds the missing skills and tells them exactly what to fix. You build every layer, and you'll know why each one exists.

- Upload a resume as a PDF
- An AI score out of 100 for the target role
- Skills found, and skills missing
- The top 3 fixes, in plain words
- Every result saved to your own database
- Live on the internet, on your own link

01 The screen — DAY 1
React + Tailwind CSS
Upload, loading, results and error states.

02 The server — DAY 1
Node.js + Express
Your own API that receives the file and keeps keys secret.

03 The AI layer — DAY 2
Claude API + Zod
Prompt rules, checked replies, retries and cost per call.

04 The database — DAY 2
MongoDB
What to save, what never to save, and result history.

05 Live on the internet — DAY 2
Vercel + Render
A real link anyone in the world can open.

What exists at the end of Day 2

A live link
A real URL anyone can open, on any phone.

AI that doesn't break
Checked answers, retries and a backup plan.

Your costs, in rupees
Per request, per user and per 1,000 users.

A plan for what breaks
Huge files, slow replies, crowds, tricky users.

Your decision document
5 big choices, the alternatives, and why.

An answer for "why?"
The real reasons, ready for any interview.

02 DAY 1 · BUILD

01

Build it like a senior engineer.

From an empty folder to your app running on your laptop.

DAY 1 · BUILD

1.1 Plan like a senior engineer

- Pick one real problem, and who it's for
- Decide what version one does, and what it doesn't
- Write the first entry in your decision document

1.2 Design the architecture

- Three ways to build the same app, side by side
- Rejected: calling the AI straight from the browser, because it exposes your API key
- Rejected: serverless functions only, because they're harder to see and debug on day one
- Chosen: a React screen talking to your own Node.js server

1.3 Set up like a professional

- A folder structure that stays clean as the app grows
- Git from the first minute, pushed to GitHub
- Secrets in environment variables, never in code

1.4 Build the screen

- React + Tailwind CSS, running with Vite
- Upload, loading, results and error states
- A layout that works on phones too

1.5 Build the server

- Node.js + Express, tested with Postman
- An API route that receives the resume and reads its text
- File size limits and clear error messages

HW Homework · 20 minutes

- Finish the results screen and post a screenshot in the group

YOU LOG OFF WITH

Your app running on your laptop

03 DAY 2 · SHIP

02

Ship it.
Let the world use it.

Plug in the AI, save the data, and put your app on the internet.

DAY 2 · SHIP

2.1 Connect the AI

- Write prompt rules the AI has to follow
- Ask for a structured reply (JSON), not free text
- Check every reply with Zod, retry once, then fall back

2.2 Know your costs, in rupees

- How AI pricing works, explained simply
- Cost per request, per user and per 1,000 users a month
- A cost cap, so one bug can't drain your wallet

2.3 Save it properly

- Design the MongoDB data for results and costs
- What to save, and what should never be saved
- Show a user their past results

2.4 Plan for what breaks

- Huge files and slow AI replies
- Many people using it at once, with rate limits
- Someone trying to trick the AI

2.5 Go live

- Deploy the server to Render and the screen to Vercel
- Environment variables and secrets on the internet
- Open the real link on your phone

2.6 The map and your next step

- The 8-level developer map, and where you stand now
- Finish your decision document
- Open Q&A: code or career

YOU LOG OFF WITH

A live link the whole world can open

04 TECH STACK

The tools you'll build with.

TOOL	WHAT IT DOES IN YOUR APP	WHEN
React	The screen people use	Day 1
Tailwind CSS	Styling, fast	Day 1
Vite	Runs the screen on your laptop	Day 1
Node.js	Runs your server	Day 1
Express	Your API routes	Day 1
Postman	Tests your API	Day 1
Git & GitHub	Saving and showing your work	Day 1
Claude API	The AI that reads and scores the resume	Day 2
Zod	Checks every AI reply is in the right shape	Day 2
MongoDB	Your database	Day 2
Render	Hosts your server	Day 2
Vercel	Hosts your screen	Day 2

What you won't get anywhere else

01

A senior engineer, thinking out loud

Why the code is written that way, what else was possible, and why it was rejected.

02

One app, end to end, in two days

Screen, server, AI, database and deployment in one project you finish.

03

AI that holds up in real use

Checked replies, retries, fallbacks, and the cost of every call.

04

Taught in Tamil

The hard ideas in the language you think in, with English where it helps.

05

Proof a recruiter can open

A live link, a decision document and a cost sheet: things someone can click, read and ask you about.

05 TAUGHT ON DAY 2

The 8-level developer map.

Each level says what you can build, and what still breaks. These two days touch all eight, in one real project.

STAGE 1

Make it appear

1. A page that exists on the internet
   HTML · CSS · JavaScript · Git
2. A screen that reacts to the person using it
   React · State · Hooks · Routing

STAGE 2

Own the whole thing

3. Your own server, end to end
   Node.js · Express · REST · Auth
4. An app that remembers, permanently
   MongoDB · Schema design · Queries
5. A complete product, alone, from a blank folder
   Structure · Architecture decisions

STAGE 3

Make it think

6. Features that weren't possible before AI
   AI APIs · Structured replies · Retries · Cost · RAG · Agents

STAGE 4

Put it in front of the world

7. Strangers use it and nothing breaks
   Security · Deployment · Rate limits · Cost caps
8. Proof for someone who will pay you
   Portfolio · Resume · GitHub · Interviews

AG

Akhshy Ganesh
AI engineer · Full-stack developer & solution architect at Aspire Systems. 5.5+ yrs building for real users, 1,000+ developers mentored. Builds live with you, both days.

QUESTIONS?

Talk to us before you book.
+91 8925836743 · enquiry@errormakesclever.com

₹499 for both live days
₹7,496 value · save ₹6,997

Where	**Live on Zoom**
Language	**Tamil + English**
Format	**2 live days**
Includes	**App · certificate · blueprints**

Reserve your seat →