

How to Win an AI Hackathon in 48 Hours

Using **Python + JavaScript + AI** – a battle-tested playbook for speed, impact, and demo-day domination.





S O F T W A R E D E V E L O P E R

PYTHON | JAVASCRIPT | FASTAPI | DJANGO |
REACT.JS | NEXT.JS

LET'S
WORK
TOGETHER



What We'll Cover

The way i won the hackathon:

01

The Winning Mindset

Why most developers fail and what winners do differently

02

The Tech Stack

AI + FastAPI + React + Vision Models

03

The 48-Hour Plan

My execution strategy

04

The Demo

How to build a moment judges remember

The Stack That Wins Hackathons



FastAPI — Backend

- Build APIs in minutes, not hours
- Auto-generated docs (Swagger UI)
- Asynchronous support out of the box and perfect for AI calls

React / Next.js — Frontend

- Rapid UI development with components
- Huge ecosystem of libraries
- Seamless API integration

AI Coding Assistant

- Claude Opus / Sonnet— plug and play
- Very Powerful AI coding models
- Production-grade intelligence instantly

The Reality of Hackathons

"The winner is not the developer with the best code. The winner builds the most **convincing production-ready app** in 48 hours."

⚡ Speed Over Perfection

Judges see demos, not architecture. Ship fast, polish the experience.

🎯 Small but Impressive

One sharp feature beats a sprawling unfinished product every time.

✨ Optimize for Wow

The demo moment is what they remember. Design for that moment.



The Winning Formula

Why Most Developers Fail

✗ Try to build too much

✗ Spend 20 hours on backend architecture

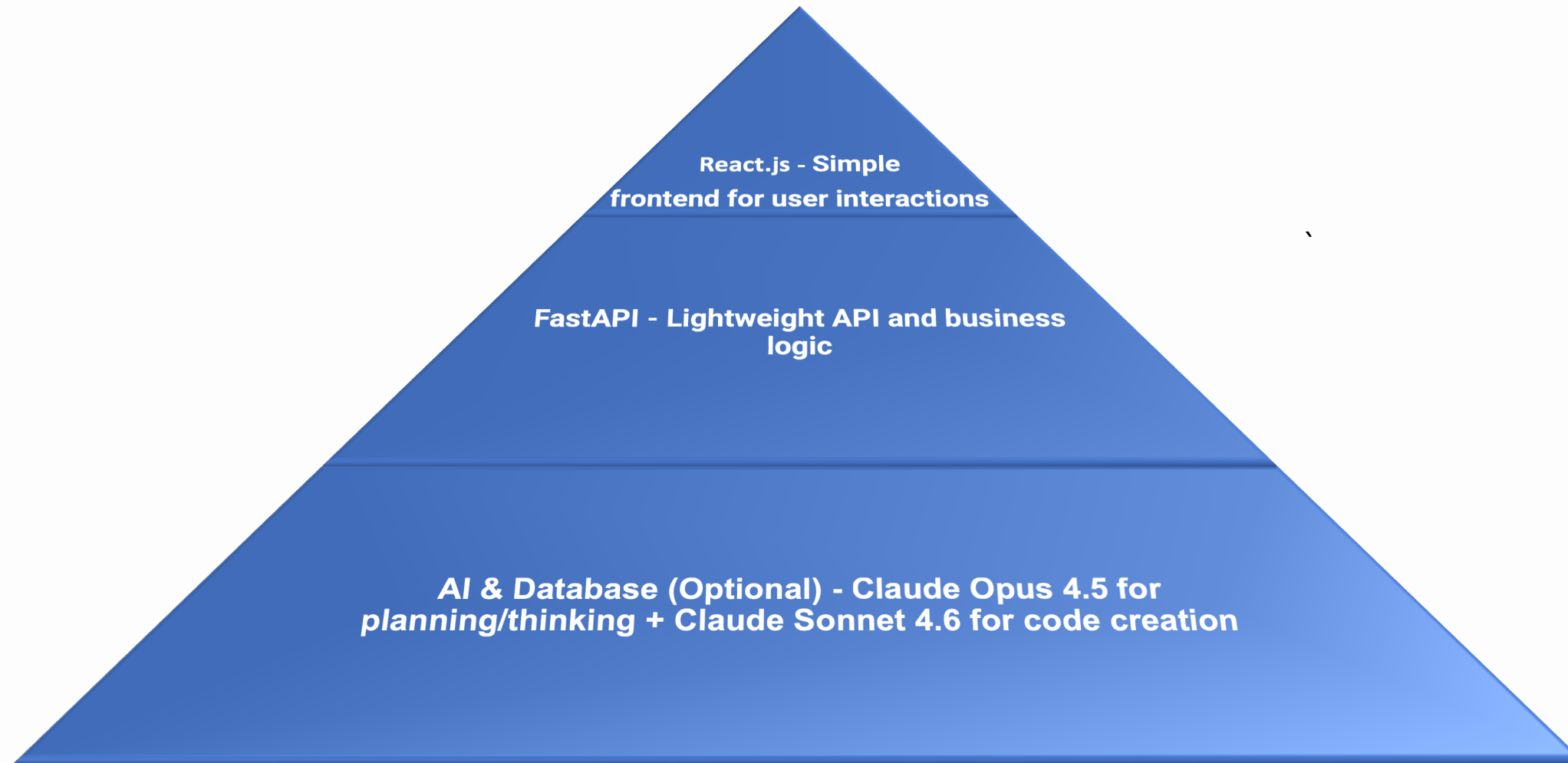
✗ Over rely on AI models

What I did

- ✓ Used existing Vision Models (YOLO)
- ✓ Built a fast MVP with one killer feature
- ✓ Obsessed over the demo experience

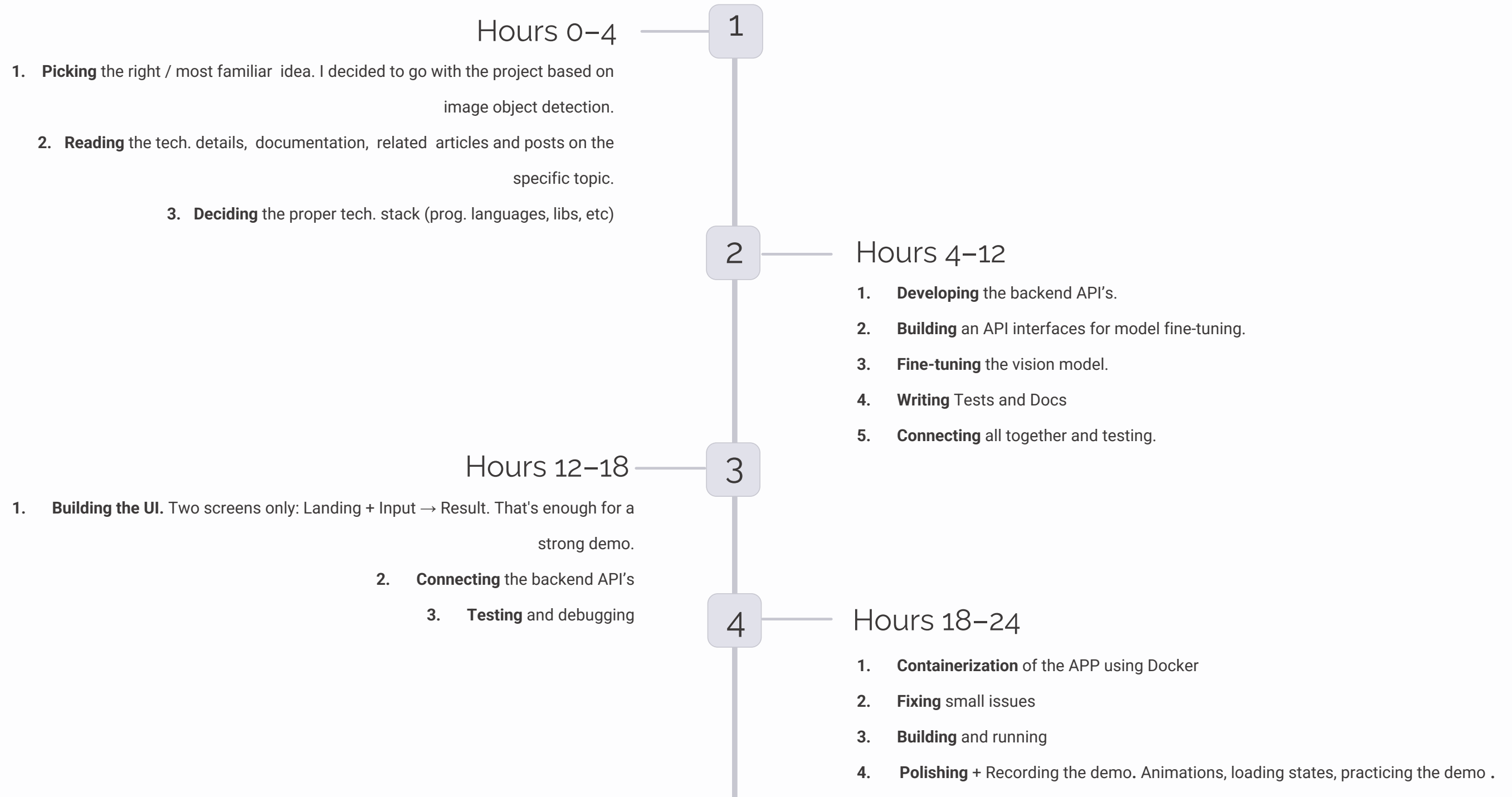
📄 **Formula:** FastAPI / React.js + AI + Great Demo = Win

The Perfect Hackathon Architecture



Keep it simple.
In hackathons, **simplicity beats scalability** every time.
Avoid microservices, avoid over-engineering.
One clean flow from UI to AI and back is all you need to impress judges.

The 48-Hour Execution Plan



The Golden Rule

"The developer that wins a hackathon is not the the best engineer.
It's a developer that builds the most **convincing future** in short period of time."

Simple Idea and execution

One sentence. One pain.

AI Feature and/or existing models

Plug into an existing API or pre-trained models.

Beautiful Demo

UX over architecture.

Clear Pitch

Problem → Solution → Impact.

2

Runner-Up #2: Andriy Kostenko

PPE Vision Detector

While everyone else grabbed pre-trained models, Andriy went the extra mile and fine-tuned YOLO11 from scratch. The memory-aware training logic that adjusts batch sizes based on available VRAM? Chef's kiss. Plus the multi-stage Docker setup and professional PDF generation service make this production-quality all the way through.

