Ultimate college blueprint by Ankush

Read this first

I did not “get lucky.” I built proof. By early college, I had cracked paid internships, started a small development agency, taught technology at a startup, and won Asia’s largest design and coding hackathon. These achievements came from shipping quickly, sharing work publicly, and stacking skills in the correct order.

This blueprint is that order—revised for you.

The 3-Year Stack (simple, focused, effective)

- **Year 1:** Select **one coding skill** and master it. Begin building in public from week one.
- **Year 2:** Go all-in on **DSA** and **major computer science subjects**. Finish the year with internship-ready proof.
- **After Year 2:** Apply consistently for internships with a portfolio full of shipped work and public proof.
- **Year 3:** Learn **System Design** and specialize in Cloud Computing, Machine Learning, or Cybersecurity. Continue building and posting.
- **Throughout:** Use **Generative AI** as an accelerator, not a substitute.

Year 1 — One Skill, One Identity, Consistent Output

1) Choose Your One Skill

- Frontend (React)
- Backend (Node/Express)
- Mobile (Flutter/React Native)
- Data Foundations (Python + Pandas)

2) 90-Day Execution Plan

Cycle: Learn → Build → Share → Repeat (every 7 days)

- Monday–Thursday: Learn from one structured source.
- Friday: Clone a micro-feature.
- Saturday: Ship a small project.
- Sunday: Document the story, push to GitHub, deploy.

3) Online Presence: Exact Posting Guide

Set up GitHub, Portfolio, LinkedIn, X/Twitter, and a Notion “Changelog.”

Weekly post template: Hook → Three bullet points → Link → Call to action.

Send direct messages to founders and engineers with a proof-first approach.

Year 2 — DSA, Core CS, and Internship Preparation

1) DSA: Go All-In

Follow a topic sequence, solve two problems per day, and maintain pattern notes.

2) Major Computer Science Subjects

- Operating Systems
- Database Management Systems
- Computer Networks
- Object-Oriented Programming and Design

3) Capstone Project

Build one production-level application with authentication, tests, CI/CD, and analytics.

4) Internship Preparation

Apply to 20 targeted opportunities each week, reach out directly to founders, and publish a capstone teardown.

Year 3 — System Design and Specialization

1) System Design

Learn the fundamentals and design five real-world systems. Document them and record Loom walkthroughs.

2) Specialization Tracks

- **Cloud:** Terraform, Docker, Kubernetes basics, observability.
- **Machine Learning:** Data preparation, model training, serving, small RAG/LLM feature.
- **Cybersecurity:** OWASP, secure your capstone, write bug reports.

Generative AI — Your Competitive Advantage

- Use it for learning concept maps, practice questions, and analogies.
- Generate boilerplate code, tests, and refactors.
- Summarize documentation, compare libraries, and extract API edge cases.
- Debug DSA problems using it as a “rubber duck.”
- Convert commits into clean weekly posts.
- Simulate interview scenarios.

Weekly Rhythms

- Maker hours: 10–14 per week.
- DSA and CS: 6–8 per week.
- Public proof: 2–3 posts per week.
- Career motion: 2–3 actions per week.
- Weekly review: 30 minutes every Sunday.

Mindset

- Clarity: One skill per season.
- Speed: Ship small, post often.
- Proof: Public build logs outweigh private potential.
- Leverage: People hire momentum.