We are in DECEMBER 2025.

You have **6–7 months** to become job-ready in backend by **June/July 2026**.

I'll rebuild the entire plan strictly for **Dec 2025 → June 2026**, with zero fluff and zero wasted time.

✅ THE ONLY BACKEND PLAN YOU NEED (Dec 2025 → June 2026)

This is compressed, aggressive, and realistic.

You follow this → you're employable.

You don't follow it → no timeline will save you.

📌 DECEMBER 2025 (Right now) — Fix Fundamentals

Absolute must.
 Your questions show gaps in the basics.

Focus on:

- Python basics → OOP → error handling

- SQL basics

- Git basics

- Data structures: arrays, hashing, stacks, queues

Mini-projects (1 week each max):

1. **CLI Banking App (Python)**

2. **Student Record System (SQL)**

Don't overbuild. Purpose = build confidence + fundamentals.

📌 JANUARY 2026 — FastAPI + PostgreSQL Hands-On

No Django. Too heavy.
 FastAPI is perfect for quick mastery + jobs.

You must learn:

- Routes

- Pydantic validation

- JWT auth

- Async CRUD

- Database connection pooling

- Writing clean API layers

Mini projects:

1. **Auth System (Register/Login/JWT)**

2. **Todo API (CRUD + pagination + sorting + filtering)**

These two ALONE make your fundamentals solid.

📌 FEBRUARY 2026 — Build a REAL Backend

1 month = 1 serious project.

Pick ONE:

🔥 Recommended: E-commerce Backend

Why?
 Recruiters love it, it's complex enough, and it shows real skills.

Features you MUST include:

- User roles (admin/user)

- Product CRUD

- Cart + orders

- Payment flow (dummy integration)

- Auth + refresh tokens

- Rate limiting

- Logging

- `.env` management

- PostgreSQL relational schema

This becomes your **portfolio's main project**.

📌 MARCH 2026 — Cloud + Deployment

Backend is useless if you can't deploy it.

You will learn:

- Docker

- Docker Compose

- Basic CI/CD

- Deploy on Render/EC2

Your project MUST run live on the internet with:

- API docs

- public URL

- database hosted

- working endpoints

This step alone puts you above 80% of freshers.

📌 APRIL 2026 — DSA + Interview Prep

Not LeetCode hell.
 Just essentials:

Patterns:

- Two pointers

- Hashing

- Sliding window

- Trees basics

- Recursion

- Sorting/searching

Daily 1–2 questions → enough.

Also:

- System design basics (caching, load balancing, scaling API)

- Project explanation mastery

- Resume clean-up

📌 MAY 2026 — Resume + LinkedIn + Networking + Applications

Now you're ready to look employable.

You will:

- Build strong resume around **one major project**

- Build LinkedIn for backend identity

- Start applying to internships + full-time roles

- Reach out to leads

- Build GitHub profile clearly

You collect interview calls mostly this month.

📌 JUNE – JULY 2026 — Interviews & Job Offers

Your skillset will be:

- Python fundamentals

- FastAPI

- PostgreSQL

- Docker

- Deployment

- One major project

- Practical DSA

You're absolutely hireable by this point.

THE TRUTH

Your timeline is tight, but not impossible.

But if you want **mid-2026 placement**, you must treat the next 3 months like war mode.

No distractions.
 No jumping topics.
 No shortcuts.
 No "AI make my code."
 Real learning. Real projects.