**✅ PHASE 1: Task Checklist — *mini-netstack Project***

**🔧 1. Project Setup & Planning**

* Create mini-netstack/ folder structure
* Initialize Git repository and add .gitignore
* Write initial README.md with project overview
* Choose target platform: Raspberry Pi OR QEMU
* List required libraries/tools in docs/setup.md

**💻 2. Development Environment**

* Install gcc, make, tcpdump, wireshark, net-tools
* Enable promiscuous mode (ip link set eth0 promisc on)
* Set up permissions or use sudo for raw sockets
* Install optional: QEMU OR set up Raspberry Pi OS

**📁 3. Folder Structure Creation**

* Create main project folders: src/, include/, test/, docs/, kernel/
* Create initial C files: main.c, ethernet.c/.h, arp.c/.h, etc.
* Add Makefile or CMakeLists.txt

**🔍 4. Ethernet Sniffer**

* Open raw socket using AF\_PACKET, SOCK\_RAW
* Receive and print raw Ethernet frame details
* Filter based on EtherType
* Log MAC addresses, EtherType, frame size

**🔁 5. ARP Handling**

* Parse ARP headers from Ethernet payload
* Match target IP
* Send ARP reply with local MAC

**🛰 6. ICMP Echo Responder**

* Detect ICMP Echo Request (type 8)
* Validate checksum
* Construct Echo Reply (type 0)
* Swap IPs and MACs

**📦 7. UDP Echo Server**

* Parse UDP header (src/dst ports, checksum)
* Extract payload
* Send UDP response to sender

**📤 8. UDP Packet Sender (CLI)**

* Accept target IP + message from CLI
* Build UDP packet manually
* Send using raw socket

**🧪 9. Testing & Demo**

* Record traffic with tcpdump/Wireshark
* Test each layer with:
  + ping, arping, nc, iperf
* Create demo.sh with test cases
* Save sample .pcap file

**🧠 10. Documentation**

* docs/architecture.md: Protocol flows, diagrams
* README.md: Full instructions, screenshots
* Diagrams: ARP flowchart, ICMP diagram
* Optionally record terminal demo (asciinema, peek)

**🧪 11. (Optional) Kernel Module**

* Write basic Netfilter hook module
* Log or drop packets based on condition
* Load via insmod, unload via rmmod