SYS-101:00030 (Federated Learning)

Architecture (what's inside)

- `***src/data.rs***`: MNIST loader with resilient download, parsing IDX, and IID splitter `***split_train_data***`.
- `***src/common.rs***`: Model definitions (linear softmax classifier), FedAvg, training/accuracy helpers, shared proto types.
- `***src/bin/server.rs***`: Parameter server (gRPC). Keeps per-model state (params, status, registered clients, round). `***train***` runs federated rounds by simulating each registered client locally on IID splits of MNIST, then FedAvg, and updates global weights. No outbound RPC during training; simulation is internal.
- `***src/bin/client.rs***`: Interactive gRPC client. Can register (`join`), init model on server, trigger server training, fetch global model (`get`), run local train/test using server's model, and request server-side test.
- `***src/bin/simple_demo.rs***`: Non-gRPC, single-process demo of FedAvg on IID splits of MNIST.

How to use

1. Start server (new shell)

- *`RUST_LOG=info cargo run --bin server --features grpc -- --address 127.0.0.1:50051`*

2. Start client (new shell)

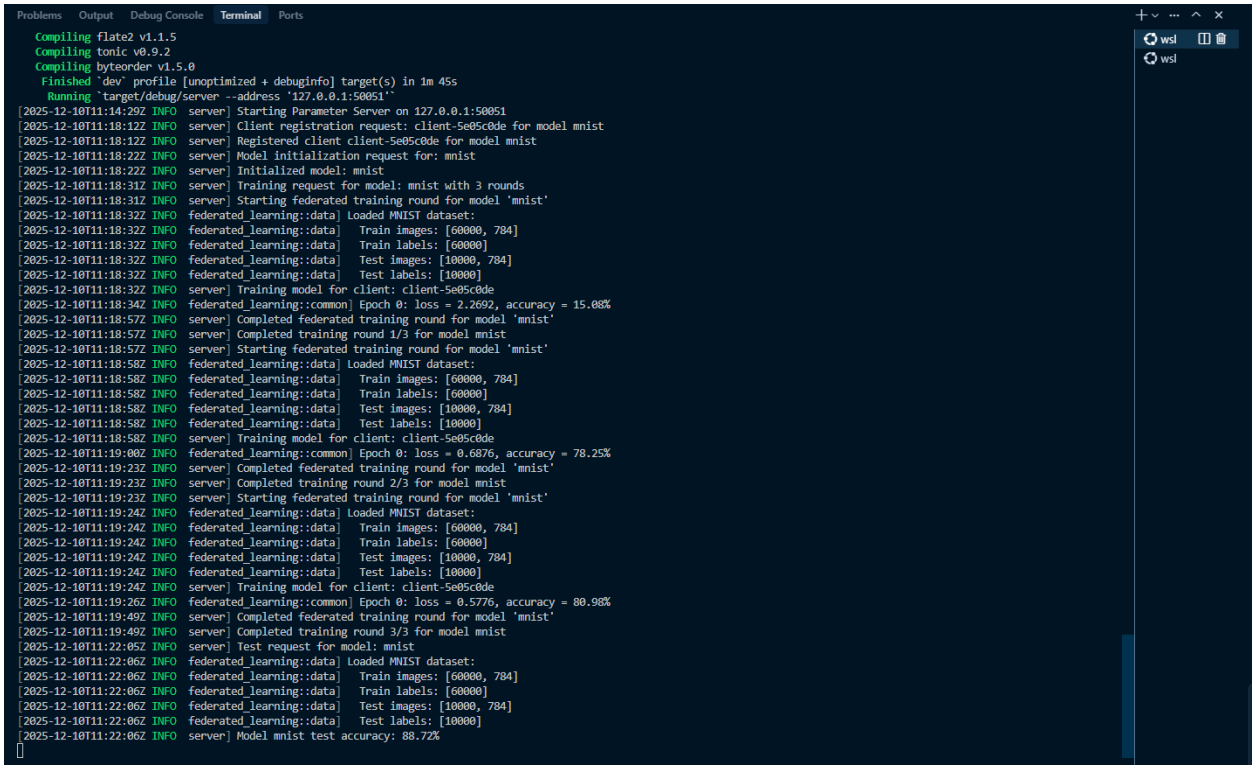
- *`cargo run --bin client --features grpc -- --server-address 127.0.0.1:50051 --client-id demo`*

3. In the client prompt, run in this order:

- *`join`* — register this client (required).
- *`server-init`* — initialize model on server.
- *`get`* — fetch current global model (baseline params/status).
- *`server-train 3`* — run 3 federated rounds (simulated local clients + FedAvg; server weights update).
- *`server-test`* — evaluate global model on server MNIST test set.
- *`get`* — confirm status/params changed.
- *`train`* — local train on this client using latest global params.
- *`test`* — local test on this client's data.

That's the whole flow: server simulates client training locally on IID MNIST splits, aggregates with FedAvg, and updates the global model; client CLI drives registration, init, training, testing, and local eval.

Example Run:
Server Side:



Client Side: