& LLMOps

Milestone 1 Assignment

"From Notebook to Reproducible Repository"

Fall 2025

1. Goal

Milestone 1 verifies that each project team has translated its idea into a *production-ready repository skeleton*. By the end of it you must be able to **clone, build, and run** your working system

2. Required Deliverables

Place every artefact in the root of your public GitHub repo (url will have to be updated in the google sheet).

D1 README.md

- One-line elevator pitch + project logo (optional).
- Architecture diagram (draw.io / Mermaid) showing *data ingestion* → *training* → *inference API*.
- Quick-start: `git clone ... && make dev`.
- Section on *Make Targets* (`make test`, `make docker`, etc.).
- FAQ (common build errors, how to setup for Windows/Mac etc).

D2 CONTRIBUTION.md

- Names, student ERP IDs.
- Table mapping members → tasks, what exactly you did (data prep, API, CI, monitoring).
- Branch-naming convention you followed (`feat/...`, `fix/...`, `infra/...`).

D3 Dockerfile

- `python:3.11-slim` (or Alpine) base.
- Multi-stage build (install system libs, copy src, install deps).
- Non-root user `app`.
- Healthcheck script pinging `/health`.

D4 .github/workflows/ci.yml

- Triggers: `push` to `main` and PRs.
- Jobs:
  a. **Lint**—`ruff` & `black --check`.
  b. **Test**—`pytest` with coverage ≥80%.

c. **Build**—Docker image tagged `$GITHUB_SHA` & pushed to GHCR.
d. **Canary Deploy**—push same image to the `canary` environment using `docker run -e CANARY=true`.
e. **Acceptance Tests**—hit canary endpoint with 5+ golden-set queries; fail if any status $\neq 200$.

D5 ML Workflow Monitoring

- **MLflow** tracking URI hosted (local/minio/S3); model v1 registered and linked in README.
- **Evidently Dashboard** for data drift on a held-out test set, exposed at `localhost:7000`.
- **Prometheus + Grafana** stack collecting at least **three** metrics `gpu_utilisation`.
- Screenshot or public link in README.

D6 Pre-commit Hooks

- `pre-commit run --all-files` must pass locally.
- Mandatory hooks: trailing-whitespace, end-of-file-fix, detect-secrets.

D7 API Documentation

- FastAPI auto-generated `/docs`.
- Example cURL + JSON schema.

D8 Security & Compliance

- **LICENSE** file (MIT, Apache 2, etc.).
- `CODE_OF_CONDUCT.md`.
- Dependency vulnerability scan via `pip-audit` (fails build on Critical CVEs). *Bonus +1 pt.*

D9 Cloud Integration

- Use at least **2 distinct services** from a major cloud provider (AWS, GCP, or Azure). Examples:
  - AWS: EC2 (hosting inference API), S3 (data storage), Lambda (serverless jobs), CloudWatch (monitoring).
  - GCP: Compute Engine, Cloud Storage, Vertex AI, Stackdriver.
  - Azure: Azure VM, Blob Storage, Azure Functions, App Insights.
- Place artefacts as follows:
  a. Annotated screenshots of the running services in the **README.md**.
  b. Update **README.md** with a *Cloud Deployment* subsection that explains:
    - Which services were used and why.
    - How to reproduce the setup.
    - How the ML workflow (data, training, inference) interacts with those services.

3. Submission Checklist

- Push with Github tag `v1.0-milestone1` to GitHub by **23:59 PKT by the deadline**.
- Verify GitHub Actions passes on the tag commit.
- Submit the public repository URL on LMS.

5. Bonus Paths

If you choose to do any Bonus path please mention it the **README.md** as well

- Docker Compose with separate `dev`, `test`, `prod` profiles. And also for separate for each service, for example: `app`, `db`, and `prometheus` services.
- GPU-enabled image and self-hosted runner integration.
- IaC sample for e.g (`Terraform`) spinning up object storage locally (MinIO). IaC or deployment scripts (Terraform, CloudFormation, ARM templates, YAMLs) in `infra/` or `scripts/`.
- End-to-end load test script using `k6` with latency SLO assertions.
- Use **DVC or Git-LFS**

6. Resources

- MLflow Docs
- Docker Docs
- GitHub Actions Guide
- Evidently AI
- Prometheus & Grafana
- Data Version Control (DVC)