Problem Statement: Intelligent E-Commerce Ad Creative Generator with MLOps Pipeline

Modern e-commerce businesses require dynamic and visually compelling advertisement creatives for product marketing. However, generating high-quality promotional content manually for thousands of products is time-consuming and resource-intensive.

Your task is to design and deploy a **Generative AI-powered Ad Creative Automation System**, supported by a fully operational **MLOps pipeline** that ensures scalable deployment, monitoring, automated retraining, and continuous delivery.

Project Objective

Build a system that automatically generates marketing creatives (text *and optionally image layout*) for e-commerce products based on the product title and short description. The creatives should follow a standardized style and be suitable for social media posting.

The project must demonstrate the full machine learning lifecycle with **production-grade infrastructure** built using modern MLOps practices.

Core Functional Features

Feature	Description
Data Ingestion	Pull product metadata (name, category, description) from a CSV/API
Generative Model Training	Fine-tuned LLM for text creatives (e.g., T5, GPT-Neo, LLaMA-based small model)
Versioning	Track datasets and model versions using **MLflow**
Model Deployment	Containerize model inference service using **Docker**
Production Deployment	Deploy microservices on **Cloud-based Kubernetes (AKS/GKE/EKS/DigitalOcean)**
Continuous Integration/Delivery	Implement GitHub Actions pipeline for automated testing, container build, and deployment
Monitoring	Expose model performance and inference latency via a **custom Prometheus exporter**

Feature	Description
Visualization	Create **Grafana dashboards** for real-time system health and custom ML metrics
Scheduling	Use **Airflow** to schedule batch inferences and retraining workflows

---

System Architecture Requirements

Your solution must include the following components:

1. **Data Layer**

   - Data stored in cloud object storage/DB

   - Airflow workflow to fetch new product entries + push to feature store

2. **Model Lifecycle**

   - Training orchestrated via **Airflow**

   - Model version tracking, experiment comparison via **MLflow**

3. **CI/CD**

   - GitHub Actions pipeline:

     - Lint + test code

     - Build Docker image

     - Push image to **Docker Hub**

     - Auto-deploy to Kubernetes via manifest or Helm charts

4. **Deployment**

   - Inference exposed through REST API service (FastAPI/Flask)

   - Autoscaling enabled on Kubernetes

5. **Monitoring + Alerting**

   - Custom exporter publishing:

     - API latency

     - Model response validity score (content quality measure)

- Throughput (# requests/sec)
  - Prometheus scrapes metrics
  - <mark>Grafana shows system + ML performance dashboards with alert rules</mark>

Success Criteria

A solution will be evaluated based on:

Category	Evaluation Focus
MLOps Automation	Fully functional CI/CD, automated retraining
Cloud Infrastructure	Stable K8s deployment with autoscaling
Observability	Useful & accurate metrics, dashboards + alerting
Model Accuracy/Creativity	Quality and relevancy of generated ad content
Reliability	Low-latency responses, high availability
Documentation	Clear README, architecture diagrams, runbooks

Expected Deliverables

- Source Code + YAML configs in GitHub repo

- Training pipelines + Airflow DAGs

- Docker Images on Docker Hub

- MLflow tracking UI with experiments + models

- Deployed Kubernetes service with test endpoint

- Prometheus + Grafana dashboards

- CI/CD workflow scripts

- End-to-end demo video + project report

Extension Options (Optional Enhancements)

- Add **multi-modal generation** (images with captions)

- Dataset drift detection + automated alerts

- Canary deployment in Kubernetes

- Authenticated API + rate limiting

- Real-time feedback loop from users

Grading Rubric — Generative AI MLOps Project

Total Marks: 100

1. Problem Understanding & Documentation — 10 Marks

Criteria	Marks
Clear problem statement & system requirements documented	4
Proper architectural diagram (data flow + MLOps pipeline)	4
Well-structured README + usage instructions	2

2. Data Pipeline + Orchestration (Airflow) — 10 Marks

Criteria	Marks
Automated data ingestion workflow	5
Scheduled pipeline for training/inference	5

3. Model Development & Experiment Tracking (MLflow) — 15 Marks

Criteria	Marks
Model implemented for generative text output	5
MLflow tracking: experiments, metrics, models	7
Model registry used for versioning & promotion	3

4. Containerization & Image Management — 10 Marks

Criteria	Marks
Inference service properly containerized using Docker	5

Criteria	Marks
Docker image successfully pushed to Docker Hub	5

5. CI/CD (GitHub Actions) — 15 Marks

Criteria	Marks
Automated build + test pipeline	5
Automated Docker image creation	5
Auto-deployment to Kubernetes from GitHub Actions	5

6. Cloud Deployment on Kubernetes — 20 Marks

Criteria	Marks
Application deployed successfully on cloud K8s	8
Service exposed externally (LoadBalancer/Ingress)	5
Autoscaling enabled (HPA based on CPU or custom metric)	4
Secrets/config managed properly (ConfigMap/Secrets)	3

7. Monitoring: Prometheus + Custom Exporter — 10 Marks

Criteria	Marks
Custom metrics exported (latency, throughput, quality)	6
Prometheus successfully scrapes metrics	4

8. Visualization & Observability (Grafana) — 10 Marks

Criteria	Marks
Custom dashboard displaying meaningful ML metrics	7

Criteria	Marks
Alerts configured for failures/performance degradation	3

Bonus Marks (Up to +10 Extra)

Bonus Feature	Marks
Model monitoring: drift detection	+5
Canary / Blue-Green deployment strategy	+5
Multi-modal generation (ad layout + text)	+5

Maximum scored: 110 (including bonus)

Scoring Guide

Performance Level	Score Range	Description
Excellent	90–100+	Fully automated, scalable, production-ready
Good	75–89	Minor gaps but major MLOps components working
Average	60–74	Some automation missing; limited monitoring
Needs Improvement	<60	Mostly manual deployment; incomplete tracking/observability