Техническое задание (ТЗ)

# 1. Общие сведения

Наименование проекта: Микросервисное cloud-native приложение для управления онлайн-заказами.  
Цель: Разработать распределённую систему для управления каталогом товаров, корзинами и заказами с поддержкой контейнеризации и масштабирования.

# 2. Назначение разработки

Приложение предназначено для моделирования интернет-магазина с архитектурой микросервисов, обеспечивающей отказоустойчивость, удобство масштабирования и расширяемость.

# 3. Функциональные требования

- Управление товарами (CRUD).  
- Управление корзиной (добавление, удаление, подсчёт суммы).  
- Создание и обработка заказов.  
- Поддержка статусов заказов.  
- Интеграция с сервисом оплаты (эмуляция).

# 4. Архитектура приложения

Система реализуется в виде набора микросервисов:  
- catalog-service — управление товарами;  
- cart-service — управление корзиной;  
- order-service — создание заказов;  
- payment-service — эмуляция оплаты;  
- API Gateway — единая точка входа, маршрутизация.  
  
Взаимодействие осуществляется через REST API (JSON). Каждый сервис контейнеризируется и разворачивается в Kubernetes.

# 5. Технологический стек

- Python 3.12+ (FastAPI/Flask).  
- PostgreSQL.  
- Docker, Kubernetes.  
- API Gateway (NGINX Ingress).  
- CI/CD (GitHub Actions).

# 6. Требования к коду

- Соответствие PEP8.  
- Асинхронность (FastAPI, async/await).  
- Документирование (docstrings, автогенерация Swagger UI).  
- Конфигурации через .env.  
- Логирование.  
- Unit-тесты (pytest, покрытие ≥ 50%).

# 7. Инфраструктурные требования

- Каждый сервис имеет собственный Dockerfile.  
- Для локальной разработки — docker-compose.yml.  
- Для продакшена — Kubernetes-манифесты (Deployment, Service, Ingress).  
- Масштабирование осуществляется средствами Kubernetes.

# 8. Тестирование

- Тесты для API и бизнес-логики.  
- Нагрузочное тестирование.  
- Проверка отказоустойчивости (перезапуск подов).