

Тестовое задание: Система мониторинга ресурсов

Общие требования:

- * Язык программирования: Python 3.12
- * Фреймворк: Django 5
- * База данных: MySQL 8
- * Дополнительные библиотеки: По усмотрению разработчика
- * Формат проекта: Django-приложение с поддержкой запуска через [manage.py](#) runserver или docker-compose (опционально)
- * Формат ответа: Ссылка на GitHub или zip архив с проектом

Задача 1: Сбор данных с удалённых машин

Описание: Реализовать периодический опрос удалённого HTTP-эндпоинта каждые 15 минут на 30 различных машинах. Допускается использование заглушки (mock-сервера) для тестирования, но в коде должно быть реализовано реальное сетевое взаимодействие.

Формат получаемых данных:

```
{ "cpu": 60, "mem": "30%", "disk": "43%", "uptime": "1d 2h 37m 6s" }
```

Требования:

- * Данные сохраняются в базу данных.
- * Опрос должен быть фоновым.
- * Архитектура должна быть масштабируемой и легко расширяемой.

Задача 2: Мониторинг и фиксация инцидентов

Описание: На основе собранных метрик из первой задачи реализовать систему мониторинга, фиксирующую выход параметров за допустимые пределы.

Пороговые значения:

- * CPU: > 85% на любом замере.
- * Память (mem): > 90% в течение 30 минут.
- * Диск: > 95% в течение 2 часов.

Требования:

- * При выходе за пределы нормы создается запись инцидента в БД.
- * Повторные инциденты по одной машине не дублируются, если уже зафиксирован активный инцидент того же типа.

Задача 3: Интерфейс отображения инцидентов

Описание: Реализовать веб-интерфейс для отображения зафиксированных инцидентов.

Требования:

- * Использовать ванильный JavaScript (без фреймворков).
- * Интерфейс должен автоматически обновляться без перезагрузки страницы.
- * Обязательная реализация аутентификации пользователей с использованием собственного middleware (не стандартного Django).