а) Задача заключается в реализации многосервисного приложения.

Первый микросервис — апишка, с одним единственным раутом(принимающая на вход страну, для которой нужно получить гимн и почту, куда его отправить). Также апишка смотрит сначала в кэш и ищет необходимую ссылку на страну. Если не найдет в кэше, то Апишка отправляет сообщение в четвертый микросервис, иначе сразу в пятый.

Второй микросервис - база данных, хранящая информацию о почте, о запрашиваемом городе и времени запроса.

Третий микросервис - Celery на RabbitMQ (можно использовать просто RabbitMQ без Celery), отвечающий за отправку сообщений, содержащих страну, для которой нужно скачать гимн) в четвертый микросервис.

Четвертый микросервис ищет страну, например здесь(<https://en.wikipedia.org/wiki/List_of_countries_by_population_(United_Nations>)), записывает в кэш и отправляет через Celery или RabbitMQ в пятый микросервис ссылку на искомую страну.

Пятый микросервис скачивает гимн и отправляет его на указанную человеком почту.

Шестой микросервис Redis — для кэша, в котором будет страна и ссылка на страну(<https://en.wikipedia.org/wiki/Israel>).

Седьмой микросервис с заданной частотой будет отправлять сгенерированный pdf файл, содержащий информацию из БД о пользователях и интересующих их странах и времени в виде pdf файла всем заданным подписчикам. Подписчики могут храниться в в другой таблице в БД.

б) Реализовать cicd, например, в gitlab, с выполнением тестов, сохранении образа в докер регистри(например, dockerhub), разворачивании в облачном сервере(aws, azure, heroku…)

в) Добавить load balancer и развернуть ваше приложение на нескольких физических серверах.

г) Использовать систему мониторинга SENTRY

д) Добавить отчет, содержащий описание ваших действий, ссылку на сайт, ссылку на VCS git (github, gitlab и пр.), трудности, с которыми столкнулись и как их преодалели.

Обратить внимание на:

* Корректное решение поставленной задачи
* Общая архитектура приложения, выбранный стек технологий
* Корректная обработка ошибок (коды и сообщения)
* Наличие спецификации API и тестов

Основной стек - Python 3.x. и любой фреймворк(flask, aiohhtp, fastapi, sanic ...)

В качестве хранилища можно выбрать любую удобную базу данных.

Для ведения разработки желательно использовать любой открытый ресурс с VCS git (github, gitlab и пр.)

Максимальное количество баллов - 30