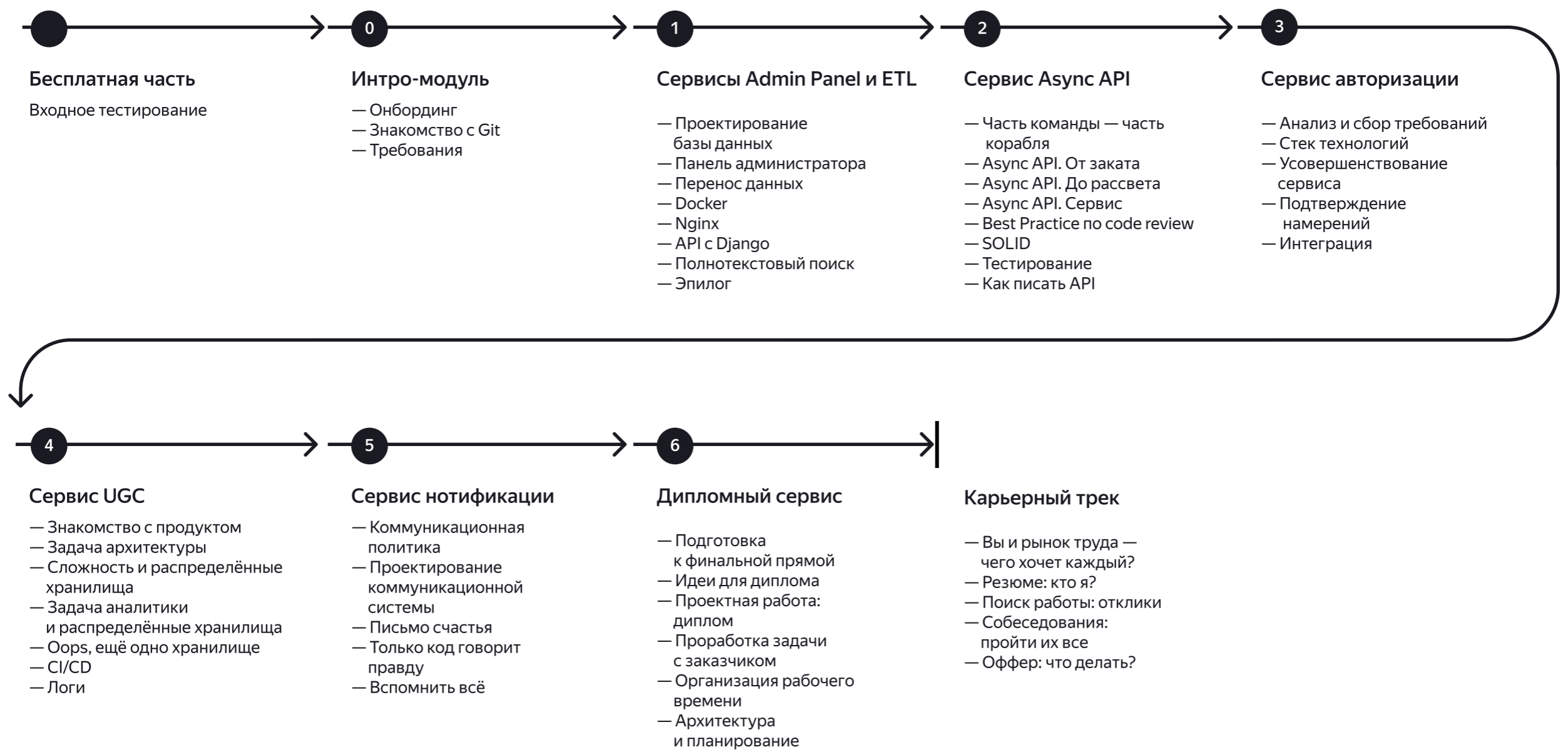


Мидл Python-разработчик



Бесплатное вступительное тестирование

2–3 часа

Серия вопросов про бэкенд-разработку.
Тест рассчитан на 2–3 часа спокойного прохождения.
Секции про Python, хранилища данных, архитектуру и DevOps.
Подробности про программу и формат курса.

Вступительное тестирование

Тема: Ты не пройдёшь?

- Middle-Python — это не только про Python
- Python
- Базы данных
- Архитектура
- Контейнеризация
- Soft-skills

0 Интро-модуль

5 часов

Модуль про требования к коду, работу с Git, формат курса и первая практика.

Темы:

- Небольшой онбординг
- Знакомство с Git
- Требования

1 Сервисы Admin Panel и ETL

115 часов

Создание и настройка панели администратора для онлайн-кинотеатра на Python с применением Django. Детальное изучение самых используемых компонентов Django — админки и ORM. Знакомство с индексами, миграциями и оптимизацией запросов. Проектирование реляционной базы данных. Устройство Docker и работа Nginx. Знакомство с архитектурным паттерном ETL и создание своего ETL для перекачки данных из PostgreSQL в Elasticsearch.

Технологии

- Python + UWSGI + Django
- PostgreSQL
- Nginx
- Docker
- ETL
- Elasticsearch

Одиночный проект
+2 проекта в портфолио

Проект: админка онлайн-кинотеатра +
ETL для перекачки данных

Спринт 1

Проектирование базы данных, знакомство с индексами и потреблением памяти

Знакомство с процессом вставки данных, dataclasses и работой со временем в базах данных

Вводная информация о базах данных без SQL, критерии выбора фреймворка и подробное изучение Django: от установки до админки

Проектная работа 1 спринта

- описание схемы базы данных PostgreSQL, с которой будут работать участники курса;
- создание скрипта для миграции данных с SQLite на PostgreSQL;
- разработка сервиса, который позволит создавать и редактировать записи в базе данных.

Спринт 2

Глубокое изучение технологии ORM

Реализация API в сотрудничестве с Django и тестирование результата

Подробное знакомство с Docker

Обработка запросов в production-среде и конфигурирование Nginx

Проектная работа 2 спринта

- реализация API на Django, который возвращает список фильмов;
- запуск Django-приложения в Docker;
- настройка работы приложения с помощью Nginx + uWSGI.

Спринт 3

Подробное изучение Elasticsearch, полнотекстового поиска и архитектуры ETL

Проектная работа 3 спринта

- создание отказоустойчивого переноса данных из Postgres в Elasticsearch.

2 Сервис Async API

80 часов

Погружение в асинхронный Python и принципы его работы и применение Fast API на практике. Знакомство с итераторами, генераторами и корутинами. Разбор многопоточности и многопроцессности.

Участники курса работают в командах по 2–3 человека. Процесс разработки имитирует разработку в настоящей продуктовой компании — у команд есть тимлид, код-ревью и еженедельные демо с наставниками.

Технологии

- Python
- ASYNCIO
- FastAPI
- Redis
- Итераторы и генераторы
- Корутины

Командный проект
+1 проект в портфолио

Проект: API онлайн-кинотеатра

Спринт 4

Принципы работы в командном модуле и ТЗ на спринт
Знакомство с асинхронным подходом, итераторами, генераторами и корутинами
Разбор асинхронных фреймворков: Twisted, AIОHTTP и FastAPI
Детальное изучение FastAPI

Проектная работа 4 спринта

- создание базовой структуры приложения FastAPI;
- настройка Dockerfile и Docker-compose;
- создание модели для фильмов и связанных данных;
- создание сервиса для получения информации о фильмах;
- доработка существующего ETL;
- добавление сервиса, который обращается к Elasticsearch и возвращает информацию о персоналиях и жанрах;
- кэширование данных в Redis.

Спринт 5

Подробное описание общих принципов код-ревью
Разница между солидным и несоллидным кодом
Функциональные тесты для сервиса и инфраструктура тестов
Понятие API и инструкция по написанию документации

Проектная работа 5 спринта

- ревью двух заданий своих коллег по команде;
- выполнение задач по SOLID;
- проведение функциональных тестов API.

3 Сервис авторизации

80 часов

Разработка сервиса, отвечающего за создание, хранение и логирование профиля юзера. Он позволит пользователям регистрироваться в кинотеатре и управлять своим аккаунтом. Подробный разбор, как правильно хранить пароли и обеспечить безопасность пользовательских данных.

Участники курса самостоятельно проектируют архитектуру сервиса, защитив свои решения на ревью. Они продумывают систему взаимодействия между сервисами, обращая внимание на обработку ошибок и возможное отключение одного из сервисов.

Погружение в синхронные фреймворки на примере Flask и работа с реляционными базами данных на примере PostgreSQL. Фокусировка на компьютерной безопасности и правильной организации сбора, защиты и хранения данных.

Технологии

- Python + Flask
- Redis
- PostgreSQL (реплики/шарды)
- OpenID
- oauth2
- jwt
- captcha + Proof-of-Work

Командный проект
+1 проект в портфолио

Проект: сервис авторизации
пользователей с защитой данных

Спринт 6

Объяснение, для чего нужен сервис Auth

Анализ и сбор требований, работа с чувствительными данными, безопасность, шифрование, хеширование и управление доступом к ресурсам

Изучение способов авторизации и проработка требований к сервису. Продолжение изучения Flask и Redis

Проектная работа 6 спринта

- создание основы сервиса авторизации и реализация всех базовых требований к нему.

Спринт 7

Детальный рассказ о работе OAuth 1.0, OAuth 2.0 и openID

Работа с каптчей и двухфакторной авторизацией

Изучение общения между микросервисами.

Настройка устойчивости сервиса к нагрузкам и изменениям API, внедрение в AsyncAPI авторизации, управление уровнем нагрузки и быстрое нахождение проблем

Проектная работа 7 спринта

- упрощение регистрации и аутентификации пользователей с помощью входа через социальные сервисы

4 Сервис UGC

80 часов

Сервис разрешает пользователям создавать свой контент в вашем кинотеатре — оставлять лайки, комментарии, отзывы к фильмам. Участникам курса нужно организовать сбор и хранение данных об активности пользователей.

Самостоятельный выбор асинхронного фреймворка и хранилища данных для сервиса. Проведение собственного исследования производительности разных хранилищ данных. Защита полученных результатов, выводов и принятых решений на демо с наставником.

Технологии

- Python
- Mongo
- ApacheKafka
- Clickhouse
- Apache Spark
- CI / CD
- ELK

Командный проект
+1 проект в портфолио

Проект: организация сбора и хранения
данных об активности пользователей

Спринт 8

Что такое UGC и зачем оно нужно. Планирование сервиса

Описание архитектуры и полезные советы по ее проектированию

Выбор хранилища, событийная работа с данными и работа с Kafka

Подходы к хранению и обработке данных, которые не помещаются на один сервер. Локальный подъем вычислительного кластера и работа с данными внутри него

Проектная работа 8 спринта

- отрисовка архитектуры сервиса;
- работа с Kafka и Spark;
- задача по ClickHouse;
- создание ETL для перегрузки данных в аналитическое хранилище.

Спринт 9

Ещё одно распределённое хранилище, хранение данных и оптимизация хранения и обработки

CI/CD: Построение непрерывной интеграции для приложения

Знакомство с логированием в микросервисной архитектуре

Проектная работа 9 спринта

- исследование по выбору хранилища;
- проектирование системы работы с данными;
- построение непрерывной интеграции;
- настройка логирования онлайн-кинотеатра.

5 Сервис нотификации

60 часов

Сервис нотификации отвечает за коммуникацию с пользователем через Email, мобильные пуш-уведомления, websocket в браузере. Это часто решаемая бизнес-задача в коммерческой разработке — от правильной коммуникации с пользователем зависят показатели бизнеса.

Изучение архитектурных проблем веб-разработки. Самостоятельное прохождение пути от бизнес-задачи до рабочего решения. Самостоятельный выбор стека технологий и отрисовка архитектуры сервиса.

Новые технологии: брокеры сообщений на примере RabbitMQ и Websocket.

Технологии

- Python
- FastAPI
- RabbitMQ
- CI/CD

Командный проект
+1 проект в портфолио

Проект: сервис нотификаций через Email, мобильные пуш-уведомления, websocket в браузере.

Спринт 10

Описание принципа работы сервиса нотификаций

Механизм отправки сообщений пользователям и единая политика контактов

Проектирование коммуникационной системы

Устройство Email и шаблонизаторов. Изучение, какие протоколы или технологии можно применить в разных ситуациях, когда нужно связаться с клиентом

Использование брокера сообщений RabbitMQ. Сервисы транзакционной рассылки писем и сокращенные ссылки

Возвращение к CI/CD-процессу

Проектная работа 10 спринта

- проектирование сервиса email-рассылок с учётом потенциального расширения количества способов связи;
- настройка отправки одинаковых писем всем пользователям;
- настройка отправки персонифицированных писем;
- создание модуля для работы с Websocket;
- приветственное письмо после регистрации пользователя.

6 Финальный проект

80 часов

Участники курса сами выбирают бизнес-задачу или придумывают её, планируют своё время, проектируют архитектуру решения, выбирают и исследуют технологии, затем защищают разработанное приложение.

Завершающая точка — защита финального проекта, где участники курса показывают результат работы своей команды.

Технологии

- Python + Framework (фреймворк участники курса выбирают сами)
- Хранилище данных и остальные технологии участники курса выбирают сами

Командный проект
+1 проект в портфолио
+ онлайн-кинотеатр в GitHub

Проект: сервис на выбор

Спринт 11

Организация работы в финальном модуле

Готовые идеи для реализации в финальном проекте

Требования для разработки и виды заказчиков.

Декомпозиция проекта и самостоятельная подготовка ТЗ для заказчика

Полезные советы, как организовать рабочее время и наладить прозрачные процессы в команде

Архитектура, риски и планирование задач по проекту

Проектная работа:

- Выполнение финального проекта.

Вебинары

В первом модуле вебинары проходят еженедельно для всех участников когорты. Наставники выбирают тему, участники курса пишут свои вопросы в треде анонса вебинара. Исходя из этого наставники формируют структуру вебинара.

Со второго модуля вместо вебинаров проходят еженедельные демо-звонки с наставниками. На демо каждая команда выбирает 15-минутный слот и показывает свои текущие результаты по проекту.

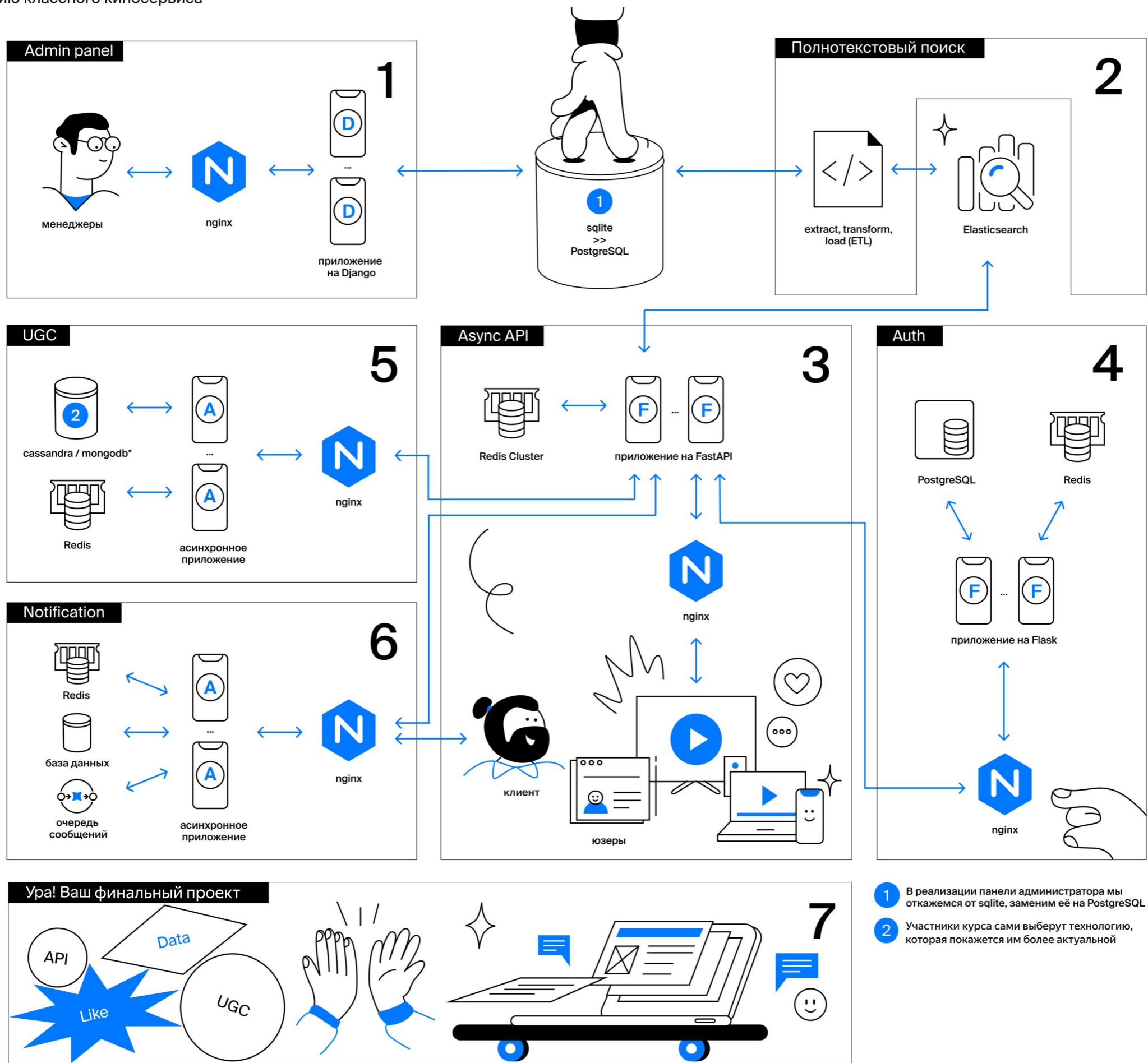
В завершающем модуле участников курса ждут демо и вебинары — каждую неделю, кроме последней. В первую неделю проходит демо с наставником, во вторую — вебинар с экспертом, в третью — демо с ревьюером.

Трек программы трудоустройства

Карьерный трек разделён на семь блоков. Сначала участник курса думает, точно ли ему нужно менять место, и определяет собственные критерии для комфортной работы. Далее он составляет резюме вместе с опытным HR-консультантом и тщательно готовится ко всем видам собеседований, а мы будем предлагать актуальные вакансии партнёров.

Инструкция

по созданию классного киносервиса



- 1 В реализации панели администратора мы откажемся от sqlite, заменим её на PostgreSQL
- 2 Участники курса сами выберут технологию, которая покажется им более актуальной