

«Национальный исследовательский университет
«Высшая школа экономики»
Лицей

Индивидуальная выпускная работа

ИТ-Проект: Di-Just

<https://github.com/qrvmil/Di-Just/tree/main>

Выполнила: Кривова Милана Дмитриевна

Группа: 11И1

Москва – 2023

1) Введение

Мой основной опыт в программировании появился благодаря обучению в Яндекс Лицее. Там я изучила основы Python, принципы работы API, разработку бэкенда на Flask и создала свои первые приложения, в том числе с графическим интерфейсом. Однако до нынешней ИВР с Django и React никогда не сталкивалась.

2) Проблемное поле

Данная работа появилась в результате моей большой любви к сервису Pinterest, который позволял находить и создавать подборки картинок на любой случай жизни от выбора прически до создания мудбордов для мотивации. Однако меня всегда огорчала ограниченность его функционала. Изображения сохраненные на доску (так называются подборки в пинтересте) смешивались в кучу, было невозможно добавить комментарий, в целом не хватало структурированности. Также расстраивало, что пинтерест позволяет сохранять только изображения, в то время как была необходимость составлять подборки, в том числе и из ссылок, например в случае подборок фильмов или образовательных ресурсов. Позднее я узнала, что такой тип подборок называется “дайджест”, так и родилась идея и название моего проекта.

3) Образ продукта

- Работа с аккаунтом / профилем

Так как мой сервис позволяет обмениваться дайджестами с людьми безусловно были необходимы базовые принципы социальных сетей - профили. Так как профиль это визитная карточка в соцсети он должен отражать вас, поэтому есть возможность добавить фотографию, описание и возраст.

- Создание дайджестов разных типов

Безусловно это основная часть моего сервиса, как и было указано в проблемном поле именно формат дайджеста (*блок ресурсов, например ссылок или картинок, и краткое описание к блоку*) позволяет эффективно и структурировано хранить подборки ресурсов, а также впоследствии ими делиться.

- Удаление дайджестов

Также необходимая часть при хранении и обмене ресурсами - возможность удалить подборку, если она перестала быть нужной и актуальной.

- Возможность комментировать дайджесты

Также необходимое требование как в любой другой социальной сети - возможность делиться мыслями с другими людьми. А также выражать благодарность за проделанную работу в составлении дайджеста.

- Просмотр статистики созданных дайджестов

Это также может быть полезно, для составителей дайджестов, чтоб понимать какие темы больше нравятся подписчикам.

- Сохранение к себе в профиль дайджестов других авторов

Безусловно необходимая часть, как и было заявлено в проблемном поле, достаточно легко потерять подборки без моего сервиса. Сохранение в профиль решает эту проблему, ведь они навсегда останутся с пользователем.

- Возможность установить напоминание прочитать дайджест через телеграм-бота

Ну и, конечно, не всегда есть время прочесть сразу дайджест, и чтобы о нем не забыть, в моем сервисе есть возможность в два клика поставить напоминание в телеграм боте о прочтении дайджеста.

4) Сделанный продукт

- 1. Регистрация пользователя / вход / редактирование профиля**
Была реализована возможность регистрации, логина по уникальному имени (username) и паролю. Также была реализована возможность восстановления аккаунта по ссылке на указанную при регистрации почту.
- 2. Создание дайджеста**
Реализована возможность создавать дайджесты двух типов: картинками и ссылками. В каждом дайджесте можно указать такие параметры как введение, топик (тема), заключение.
- 3. Комментирование дайджестов**
Имеется возможность комментировать каждый дайджест. В комментариях сохраняется имя автора и ссылка на его профиль.
- 4. Просмотр главной страницы / работа с дайджестами**
На главной странице можно подобрать удобный формат дайджестов. Можно выбрать дайджесты определенной темы (topic), а также посмотреть более новые или, наоборот, старые дайджесты.
- 5. Сортировка дайджестов пользователя**
Пользователю доступна возможность разделить дайджесты на созданные и сохраненные, а также на дайджесты картинками и дайджесты ссылками.
- 6. Взаимодействие с telegram-ботом**
Рядом с каждым дайджестом есть специальная кнопка, нажав на которую можно поставить напоминание в telegram-боте для прочтения дайджеста.

Можно выбрать удобный интервал времени: через минуту, через час, утром, вечером, через день, через неделю.

7. Сохранение дайджеста как расширение Chrome

С помощью расширения Chrome пользователь может сохранить картинку или ссылку для будущего использования в дайджесте.

5) Backend

Серверная часть моего приложения была реализована с помощью фреймворка Django на языке Python. Так как мною было принято решение использовать современный фреймворк React для разработки web-приложения, необходимо было чтоб бэкенд обладал API для взаимодействия с ним. API реализовано благодаря надстройке django-rest-framework, для аутентификации использовались knox токены, а для работы на одном хосте django CORS headers. Всё вышеперечисленное позволило реализовать эффективный API с высокой безопасностью при авторизации.

Django удобно своей ORM для работы с сохраненными данными на сервере, с помощью базы данных Sqlite. Это реализовано с помощью сущности Models. В моем проекте были созданы Models для каждого блока информации, который возникает при работе моего приложения:

- Profile - для хранения информации о пользователе.
- Topics - для хранения темы дайджеста.
- Image Digest - для хранения самого дайджеста из картинок, который содержит информацию о создателе, название, тему, вступление, заключение, настройки приватности, а также список пользователей, кто сохранил дайджест.
- Link Digest - аналогичная модель, но для хранения дайджеста из ссылок
- Digest Image - модель для хранения картинки, которая содержит ссылку на картинку в памяти, описание, а также указание на дайджест которому принадлежит она.
- Digest Image - модель для хранения ссылки, которая содержит ссылку, описание, а также указание на дайджест которому принадлежит она
- Comments - модель для хранения комментария. Она содержит ссылку на пользователя, текст комментария, время создания, а также ссылку на дайджест, где был оставлен комментарий.

6) Средства разработки

Backend:

Для реализации backend-части своего проекта я использовала библиотеку Django. Основные ее плюсы заключаются в поддержке ORM (Object-Relational Mapping), которая предоставляет гибкий ресурс для работы с базами данных через ООП-подход python'a. Также одним из главных плюсов данного фреймворка является широкий спектр задач, которые с помощью него можно решать. И

возможность быстро и удобно масштабировать проект, под любую нагрузку. Именно поэтому я решила адаптировать его технологии под нужды своего проекта.

API:

API было реализовано при помощи фреймворка Django Rest Framework (DRF). Это один из самых мощных и гибких фреймворков, работающих с API на django, основное преимущество заключается в упрощенной работе с сериализацией. Из коробки фреймворк предоставляет довольно обширную систему для работы с различными видами запросов, но при необходимости, более тонкие моменты можно настроить вручную. Я воспользовалась этой опцией при работе над своим проектом, когда столкнулась с необходимостью реализовать верификацию почты и восстановление пароля. Я вручную прописывала post- и put- запросы, которые были необходимы в каждом из этих случаев. Стоит отдельно отметить удобство работы с пагинацией у данного фреймворка, что также послужило плюсом в выборе используемой библиотеки.

Frontend:

Frontend-часть проекта была реализована на библиотеке React (JavaScript). На данный момент это одна из самых передовых библиотек для работы с web-приложениями, она обеспечивает более быстрый рендеринг, основывается на принципах декларативного программирования, позволяет повторно использовать компоненты, как и DRF, ориентируясь на принципы DRY (Don't Repeat Yourself). Также стоит отметить, что виртуальный DOM (Объектная модель документа) обеспечивает два серьезных преимущества: эффективность и производительность. Данная библиотека не самая простая в освоении, как минимум, потому, что использует собственный формат синтаксиса jsx. Несмотря на все сложности освоения данной библиотеки, мне удалось реализовать web-интерфейс именно на ней, используя подход переиспользования компонент и шаблонов. Особое внимание стоит также уделить библиотеке axios, которая отвечает за запросы между клиентом и сервером. Она более гибкая, нежели встроенная функция fetch в React, но требует дополнительных усилий для того, чтобы разобраться как правильно передать и принимать запросы с сервера. Тем не менее, данная библиотека действительно помогла мне при работе над проектом, так как данные, которые я отправляла и получала с сервера были совершенно разные (картинки, текст, json-формат), но не смотря на это получилось установить процесс обмена данными между backend и frontend частями проекта.

Среды разработки:

При реализации серверной части моего проекта и бота я пользовалась IDE Pycharm, а также разобралась с работой встроенных debug утилит. При написании фронтенда моей основной средой разработки был использован VSCode из-за большого количества расширений, который помогают быстрее работать с React и языком разметки.

Git:

Во время выполнения ИВР я регулярно пользовалась системой контроля версий git, а также репозиторием на github. Что позволило избежать внезапную потерю кода и дало возможность откатывать к более ранним версиям, при каких-либо поломках.

7) Этапы работы

1. Изучение Django (май-июнь)

Изучение Django прошло достаточно успешно, мной был выбран курс на Udemu, который полностью покрывал тему разработки на современной версии фреймворка. Однако по DRF было мало структурированных материалов, поэтому пришлось самостоятельно находить примеры использования в сети и оттачивать от них.

2. Реализация системы регистрации и аутентификации по API (июль)

Основной сложностью в системе аутентификации было отправление писем с подтверждением почты. Однако все остальные части в том числе принципы rest api и системы токенов аутентификации были успешно реализованы и я уложились в дедлайн.

3. Реализация системы публикации дайджестов и взаимодействия с ними (август)

Эта часть вызвала больше всего вопросов при написании бэкенда, так как из-за большого количества различных сущностей было сложно продумать связи между всеми объектами, для суммаризации видения проекта очень помогли UML и объектные диаграммы. Отдельной сложностью была реализация загрузки картинок на сервер. Однако и эту часть я выполнила вовремя.

4. Изучение React (начало сентября)

Изучение React проходило достаточно тяжело так как, реакт в первую очередь основывается на принципах декларативного программирования, до этого проекта незнакомых для меня. Также в открытом доступе оказалось крайне мало материалов в открытом доступе, позволяющих освоить React.

5. Настройка обмена данными между React и Django (конец сентября)

Данный этап оказался самым болезненным для моего проекта. Во многом потому, что крайне мало информации по этой теме, и лишь ресурсы

DigitalOcean предоставляли обучающие tutorиалы по объединению Django и React. Также мне пришлось переписать часть бэкенда, для приведения проекта в соответствие с принципами CORS (Cross Origin Resource Sharing) и особенностями аутентификации в React.

6. Реализация логики отправки и получения запросов на бэкенд (начало октября)

Во многом данный этап основывался на методе проб и ошибок по вышеперечисленным причинам. Однако у меня получилось реализовать, по моему мнению, элегантную систему передачи информации на фронтенд.

7. Дизайн интерфейса web-страниц (конец октября)

Данный этап был реализован в срок и достаточно хорошо.

8) Рефлексия

- В ходе работы над проектом возникла проблема в том, что не всегда быстро получалось найти хороший материал с описанием той или иной технологии, также иногда встречались трудности при конфликте версий различных библиотек.
- На мой взгляд основной путь развития продукта состоит в постепенном добавлении и других форматов дайджестов помимо картинок и ссылок (например, довольно интересная идея делать дайджесты-путеводители на картах).
- В ходе работы над продуктом я приобрела много компетенций. Сейчас я лучше понимаю, как устроены web-приложения, как взаимосвязаны между собой различные их части. Лучше научилась понимать структуру баз данных и принцип взаимодействия таблиц, могу подобрать оптимальный способ для хранения той или иной информации. Все эти навыки мне пригодятся и в дальнейшем обучении, они заложили некоторую основу для развития дальнейших компетенций.
- В начале работы были риски не найти информацию или образовательные ресурсы, которые бы содержали исчерпывающую информацию по изучаемой теме. На этапе изучения React это было особо ощутимо, однако в серьезную проблему это не переросло.

9) Заключение

Данный проект оказался действительно большой во всех пониманиях задачей. По сути, с нуля реализована полноценная социальная сеть с возможностью создания постов (дайджестов), лайков (сохранений) и комментариев. На данный момент это самый масштабный проект, который я когда-либо реализовывала.

Мне стало понятнее какими именно областями в компьютерных науках мне бы хотелось заниматься. А также моя ИВР безусловно будет большим плюсом в резюме, так как удалось поработать с самыми передовыми технологиями в web-разработке на данный момент.