

«Национальный исследовательский университет  
«Высшая школа экономики»  
Лицей

Индивидуальная выпускная работа

**ИТ-ПРОЕКТ: Teach&Learn**

<https://github.com/DNick/ivr>

*Выполнил: Денисов Николай Дмитриевич*

*Группа: 11u5*

Москва – 2023

# 1. Введение

Для меня программирование — это отдельный интересный вид творчества. С одной стороны, не нужно писать много текстов, «складывать слова в предложения», с другой — всё-таки это творческий процесс, который подразумевает создание нового продукта, который имеет свою сложную структуру и может помочь обществу.

К моменту начала разработки проекта мой опыт включал в себя прохождение курсов по программированию (например, курсы по промышленной разработке на Python, программированию на C++, олимпиадному программированию и компьютерным сетям) и участие в нескольких проектах (создание чат-ботов, веб-приложений на Django, Flask).

Моё обучение в сфере IT во многом состояло из онлайн-курсов, благодаря которым я получил ценные навыки и знания. Цель данного проекта - сделать доступными онлайн курсы для более широкой аудитории с разной профессиональной ориентацией и по различным направлениям. Мне показалась эта сфера востребованной и важной. Онлайн курсы часто бывают полезны, поэтому я хочу дать возможность другим людям создавать свои, чтобы популяризировать это направление.

## 2. Проблемное поле

Со времен пандемии много людей в силу обстоятельств перешли на онлайн образование. Однако Платформы, предоставляющие возможность создать с нуля или записаться на онлайн-курс, имеют ряд существенных недостатков:

- Сложно начать - Многоступенчатая регистрация пользователя на новой платформе, необходимость скачать приложение платформы на телефон, при желании проходить курс с мобильного устройства (браузерная версия не всегда хорошо адаптирована).
- Сложно создать – Многие сайты не подразумевают создание курсов с мобильных устройств, так как интерфейс сложно адаптировать под маленький экран. Более того, некоторые платформы не предоставляют возможности создавать свой собственный контент.
- Сложно разобраться – Пользователю необходимо разбираться в новой системе, изучать её возможности и ограничения.

## 3. Образ продукта

Продукт представляет собой Telegram бота, непосредственно из которого можно создать свой курс (платный/бесплатный) по рисованию, программированию, пению и т. д. Курс представляет собой набор уроков, в каждом уроке могут быть представлены аудио, видео, ссылки, и/или текстовые материалы. Чтобы курс вышел в интернет, он должен сначала пройти валидацию от модератора/администратора. Создатель курса может получать обратную связь от пользователей. Клиенты же могут найти себе подходящий курс по теме, при этом можно сортировать курсы в теме по цене, оценке и дате добавления.

## 4. Сделанный продукт

В целом, большинство сценариев было реализовано с небольшими изменениями.

- "Сценарий 1" был реализован полностью, но с изменениями. Пользователь может просматривать курсы в web интерфейсе.
- "Сценарий 2" был реализован полностью за исключением проверки на нецензурную лексику. Оказалось неважным из-за ручной проверки. Можно создать курс и загрузить в него материалы.
- "Сценарий 3" был реализован полностью за исключением сортировки списка курсов. Оказалось ненужным, потому что у одного человека скорее всего не будет много разных курсов. Можно просматривать прежде созданные курсы и редактировать их.
- "Сценарий 4" был реализован частично. Решил не делать курсы с регулярной отправкой сообщений, так как непонятно, зачем это нужно. Пользователь может пройти курс.
- "Сценарий 5" был реализован не так, как в нём написано. Пользователь может оставить отзыв, при этом указав кому он должен прийти: создателю курса или администратору.
- "Сценарий 6" был реализован в упрощённом виде. Если создатель курса загрузил ссылку на обсуждение, то пользователям всегда предлагается в него зайти.
- "Сценарий 7" был реализован полностью. Курс после его создания проходит модерацию от администратора.

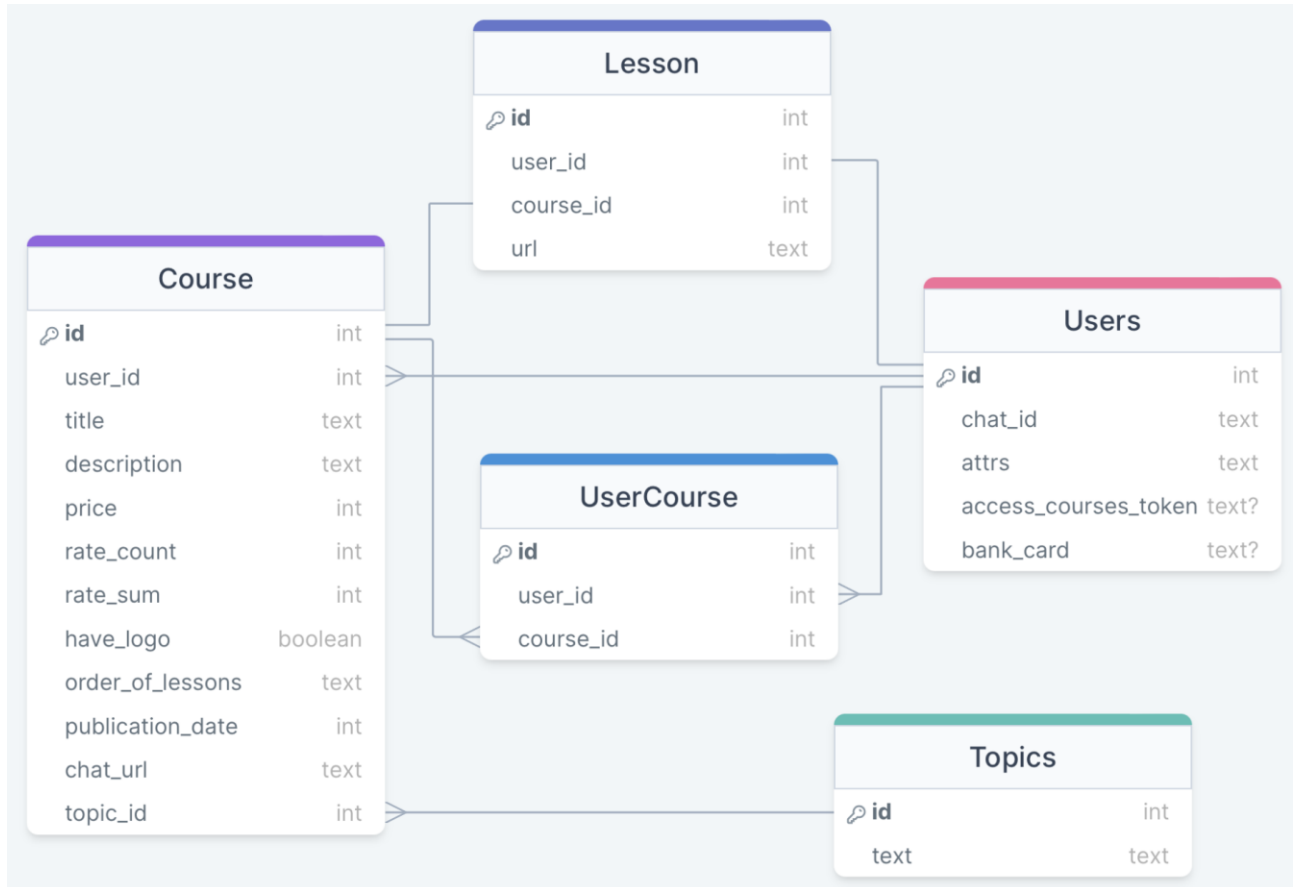
## 5. Бэкенд

Использовалась база данных PostgreSQL, расположенная на Linux сервере. База данных имеет 5 таблиц:

- Users – Хранит всех зарегистрированных пользователей
- Course – Хранит созданные курсы
- Lesson – Хранит добавленные в курсы уроки

- Topics – Хранит сферы, по которым создаются курсы
- UserCourse – Хранит записи пользователями на курс

Ознакомиться со связями между таблицами, полями таблиц предлагаю на изображении:



## 6. Средства разработки

Для проекта был выбран язык Python, но в проекте также встречаются HTML файлы для отображения каталога курсов. Чтобы реализовать продукт, было использовано достаточно большое количество библиотек и фреймворков, но я приведу самые масштабные из них:

- pyTelegramBotAPI – библиотека для создания Telegram ботов (наиболее функциональна среди аналогов)
- peewee – ORM для взаимодействия с базой данных (легковесная и удобная)
- telegraph – библиотека для взаимодействия с одноимённым сервисом (удобный инструмент для создания уроков в рамках курса)
- boto3 – библиотека для работы с удалённым хранилищем медиафайлов Yandex Cloud (современное, надёжное российское решение для хранения файлов)
- Flask - веб-фреймворк (оптимальное решение для задачи рендеринга HTML страниц)

- YooMoney – библиотека-прослойка между API ЮMoney и разработчиком. ЮMoney - сервис электронных платежей, который я использовал для приёма платежей от пользователей за платные курсы (удобный сам по себе, даёт возможность принимать платежи на кошелёк физического лица)

## 7. Этапы работы

Этапы работы над продуктом были таковы:

- Продумывание проекта в целом. То, что я хочу видеть в итоге.
- Разработка пользовательских сценариев. Было сделано ещё в 10 классе.
- Проектирование базы данных. Этот этап я выполнял в ходе всего времени работы над продуктом, так как приходилось постоянно добавлять поля в таблицы.
- Разработка функций создателей курсов.
- Разработка функций клиентов.
- Тестирование, отладка. Написал unit тесты к трём функциям, а также просто использовал бота для проверки на ошибки.

## 8. Рефлексия

Проблемы встречались часто. Приходилось использовать технологии, неизвестные мне ранее. Приведу несколько примеров:

- Приём платежей. Изначально не представлял себе, как это можно реализовать. Пришлось рассматривать разные варианты, искать подходящие. В итоге остановился на сервисе ЮMoney.
- Хранение фотографий/логотипов курсов в удалённой базе. Также не знал, как разработчики это обычно реализуют, но в ходе поиска нашёл подходящую мне библиотеку и хранилище
- Дизайн HTML страниц. Так как я сам не могу реализовать красивый дизайн, использовал сервис Bootstrap с готовыми CSS классами.

В качестве того, что в будущем можно было бы добавить в продукт, хочу отметить следующее:

- Использование ML модели для ранжирования курсов в каталоге в соответствии с интересами пользователя;

- Возможность написания текстовых отзывов о курсе (для доступа к ним всех пользователей);
- Другие способы оплаты;
- Возможность добавления демо-ролика курса;
- Возможность предоставления по решению создателя частичного доступа к курсу в качестве ознакомления;
- Выдача сертификатов о прохождении курсов;
- Добавление автоматической модерации курсов (проверка на мат, осмысленность, содержательность).

В ходе работы я научился быстро искать нужную для меня информацию в интернете и применять её на практике (приходилось изучать новые технологии для их реализации в продукте).

Были риски не реализовать все заявленные функции. В целом, риски не осуществились. Единственное, что не получилось сделать — это выложить проект на Heroku, так как этот сервис сделал все тарифы платными и не разрешает оплатить услуги картой, выпущенной в России.

## 9. Заключение

Работать над таким полезным и интересным продуктом мне понравилось. Надеюсь, что в дальнейшем у меня будет возможность и свободное время, чтобы улучшать и совершенствовать бота. Я получил новый опыт и теперь могу дополнить своё портфолио этим проектом. Создание подобных ботов именно на базе Telegram, динамично развивающейся платформы, – актуальная задача в настоящее время.