

ЛАБОРАТОРНАЯ РАБОТА №3

РЕАЛИЗАЦИЯ СЕРВЕРНОЙ ЧАСТИ ПРИЛОЖЕНИЯ СРЕДСТВАМИ DJANGO И DJANGORESTFRAMEWORK

Цель: овладеть практическими навыками и умениями реализации web-сервисов средствами Django.

Оборудование: компьютерный класс.

Программное обеспечение: Python 3.6+, Django 3, Django REST Framework (DRF), PostgreSQL *.

Практическое задание:

Реализовать сайт, используя фреймворк Django 3, Django REST Framework, Djoser и СУБД PostgreSQL *, в соответствии с вариантом задания лабораторной работы.

План выполнения работы:

1. Выполнить практическую работу 3.1
(https://docs.google.com/document/d/1PkpwxCUYQ2_Pi8Fpcgno6te3oCQHZfkh03Zxt6DhHSw/edit)
2. Выбрать вариант или предложить свой, есть 3 способа:
 - 2.1. Предложить свой вариант.
 - 2.2. Использовать вариант из дисциплины «Адаптивный веб-дизайн».
 - 2.3. Выбрать вариант из вариантов по курсу «Основы баз данных»
(https://drive.google.com/file/d/174gPjJ7AOHfzteYcobPY0x7sFBTkN1Xx/view?usp=s_haring).
 - 2.4. Выбрать бонусный вариант от Давида или Максима Валерьевича Хлопотова (см. пункт **Бонусные варианты**).

По любому из способов функционал нужно согласовать с преподавателем на консультации.
3. Реализовать модель базы данных средствами DjangoORM (согласовать с преподавателем на консультации).
4. Реализовать логику работу API средствами Django REST Framework (используя методы сериализации).
5. Подключить регистрацию / авторизацию по токенам / вывод информации о текущем пользователе средствами Djoser.
6. Выполнить практическую работу 3.1 по оформлению документации (в процессе разработки)
7. Реализовать документацию, описывающую работу всех используемых endpoint-ов из пункта 3 и 4 средствами Read the Docs или MkDocs.

Работа выполняется индивидуально.

Код практический и лабораторной части должен быть загружен в репозиторий курса, в соответствии с инструкциями [тут](#).

Бонусные варианты

Работу необходимо защитить на консультации или прислать видео с описанием проделанной работы.

Бонусные варианты

От Давида:

Система для проведения хакатонов

Есть несколько сущностей: жюри хакатона, участники, главный администратор, кураторы задач. У участников есть возможность выбрать одну из задач (регистрируется и имеет доступ к системе только капитан команды), после выбора задачи капитан может предложить решение, скачать какие-то файлы, которые ему предложены, посмотреть на ссылки, которые есть в задаче.

Ссылки и файлы к задачам добавляют кураторы задач через отдельный админский интерфейс, кроме того у них есть доступ к решениям, как и у членов жюри. Куратор может назначаться только на одну задачу и проводить консультации (например, в зуме, ссылку на консультацию он крепит к самой задаче и это выводится у команды в лк). Жюри может оценивать решения участников, с комментариями, сортировать решения по дате публикации.

Капитан при регистрации заполняет только свои учётные данные, после в кабинете команды — он может заполнить данные по каждому участнику, название команды и какой-нибудь условный девиз/описание.

У главного админа есть доступ ко всему, но он не может добавлять команды и редактировать их решения. Так же, не имеет права оценивать решения участников. Только просматривать. Ещё он может создавать задачи, которые потом будут дополнять кураторы. Ну и назначать кураторов на задачи, разумеется.

От Максима Валерьевича Хлопотова:

Сервис для peer review

Преподаватель создаёт задания, критерии оценок и варианты задания.

Студенты выполняют задание, другие студенты проверяют задание по этим критериям.