

Лабораторная работа №3

Тематика:

1. Веб-программирование;
2. Веб-фреймворки: Django/Flask;
3. Telegram API

Общее описание

Варианты на выбор

1 Веб-сайт

Тема выбирается самостоятельно.

Основные задания:

- Продумать необходимые сущности для описания предметной области. Реализовать их в терминах моделей, используя подходящие типы данных и связи объектов.
- Реализовать CRUD (create, read, update, delete) операции;
- Реализовать админскую панель (в случае выбора фреймворка Flask);
- Добавить все модели в админскую панель. Отобразить связанные модели на одной странице;
- Реализовать механизмы авторизации/аутентификации;
- Добавить тесты на pytest. Использовать fixtures для создания юзеров и соединения к базе данных. Использовать parametrize;
- Добавить logging;
- Добавить тесты;

Требования:

- Проект представляет собой решение какой-либо бизнес задачи, а не просто набор несвязных модулей;
- Бекенд проекта должен быть написан либо на Django, либо на Flask;
- Использовать не sqlite базу данных;
- Добавить индексы для полей по которым идет выборка;
- Реализовать связи many-to-many, one-to-many, one-to-one;
- Реализовать Join запросы;
- Покрытие тестами кода на 80% и выше;
- (*) Покрытие тестами кода на 95% и выше;
- Приемлемый внешний вид. Использование css (например бутстрап), при желании реализация своих стилей (можете найти и скачать шаблоны в интернете);
- Валидация форм как на стороне сервера, так и на стороне клиента;
- Использовать мощный фреймворк VanillaJS. Например: реализация toaster в случае клика или ошибки при валидации;
- (*) Использовать JS фреймворки такие как: React JS, Angular2, Vue JS;
- Приложение должно считывать свою конфигурацию из конфигурационного файла;
- Поддержка разного уровня логирования (уровень логирования брать из конфигурации приложения);
- (*) Использование различного формата вывода логов для режима разработки и продакшн запуска;
- Ограничить использование API проекта для неавторизованных запросов.

2 Telegram бот

Тема выбирается самостоятельно. Основные задания практически идентичны заданию по Веб-сайту, однако здесь меняется интерфейс взаимодействия с бекендом с браузера на telegram.

Основные задания:

- Продумать необходимые сущности для описания предметной области. Реализовать их в терминах моделей, используя подходящие типы данных и связи объектов;
- Реализовать CRUD (create, read, update, delete) операции и также возможность их использовать через интерфейс telegram;
- Добавить возможность администрирования через интерфейс telegram;
- Реализовать механизм авторизации;
- Добавить уведомления от бота;
- Добавить тесты на pytest. Использовать fixtures для создания юзеров и соединения к базе данных. Использовать parametrize;
- Добавить logging;
- Добавить тесты;

Требования:

- Проект представляет собой решение какой-либо бизнес задачи, а не просто набор несвязных модулей;
- Бекенд проекта должен быть написан либо на Django, либо на Flask;
- Использовать не sqlite базу данных;
- Добавить индексы для полей по которым идет выборка;

- Реализовать связи many-to-many, one-to-many, one-to-one;
- Реализовать Join запросы;
- Покрытие тестами кода на 80% и выше;
- (*) Покрытие тестами кода на 95% и выше;
- Бот должен корректно реагировать на любой тип сообщений;
- Валидация ошибочных запросов боту;
- Приложение должно считывать свою конфигурацию из конфигурационного файла;
- Поддержка разного уровня логирования (уровень логирования брать из конфигурации приложения);
- (*) Использование различного формата вывода логов для режима разработки и продакшн запуска;
- Ограничить использование API проекта для неавторизованных запросов (частично либо полностью);

3 Любое задание

Вам предоставляется возможность предложить задание лабораторной (предложение актуально до 01.05.2021).

Для этого, вы должны:

- Составить техническое задание (ТЗ) для вашего проекта;
- Согласовать и скорректировать ТЗ с преподавателем по лабам.

Примечание: если проект будет не только в вашем гит репозитории, то результат лабораторных в этом семестре обнуляется.