



Работа над мини-проектом. Примеры проектов

Блок WebServer оснащён большим выбором тем и средств для реализации проекта. Обязательные требования к проекту:

- Использование базы данных (sqlite или любой другой) с применением ORM (sqlalchemy или любой другой).
- [продвинутый уровень] Разворачивание проекта на любом облачном сервисе (pythonanywhere, heroku, Яндекс.Облако).

Ваши ученики могут решить делать веб-приложение, REST API для какого-нибудь своего сервиса или чат-бот (навык).

Хороший проект по направлению веб-приложение отличается следующими характеристиками (проект не обязательно должен соответствовать всем пунктам):

- Регистрация и авторизация пользователей.
- Разделение страниц сайта на открытую часть (доступную всем) и закрытую часть (доступную только зарегистрированным пользователям).
- Базовая система разграничения доступа - зарегистрированный пользователь видит только свою информацию.
- Зарегистрированный пользователь может добавлять, редактировать, удалять сущности системы (новости, товары и т.п.).
- Понятный интерфейс пользователя - использование компонентов Bootstrap (или аналогичной библиотеки).
- Наличие некоторой функциональности доступной через API (выгрузка всех публичных новостей в виде json).
- [продвинутый уровень] Наличие большого количества связанных сущностей в базе данных (больше 3).
- [продвинутый уровень] Использование в сущности не только простых типов данных, но и изображений и/или файлов.
- [продвинутый уровень] Возможность базовой фильтрации и сортировки сущностей на серверной стороне.
- [продвинутый уровень] Базовая бизнес-логика (при добавлении товара в заказ, его количество на складе уменьшается).
- [продвинутый уровень] Наличие тестов для API.

Дизайн приложения не влияет на итоговую оценку.

Ученики могут реализовывать веб-приложение с использованием фреймворка отличного от Flask. Если проект будет построен на Django, то он должен быть объёмнее и сложнее того проекта, который выполнен на Flask или Bottle. Это обусловлено тем, что Django многое делает автоматически, без написания кода.

Хороший проект по направлению чат-бот/навык имеет следующие характеристики (проект не обязательно должен соответствовать всем пунктам):

- Навык (бот) должен поддерживать диалоги с пользователем, сохраняя контекст разговора. Иными словами, в работе навыка должна использоваться информация от пользователя в нескольких отдельных сообщениях;
- Навык (бот) должен использовать какие-либо внешние HTTP-API (Карты, Переводчик или любые другие доступные);

- Если во время работы навыка (бота) появляются ошибки (например, HTTP-ошибки при выполнении внешних запросов), он должен понятной форме сообщать о них пользователю и продолжать корректный диалог;
- [продвинутый уровень] Плюсом будет использование деталей API Алисы или чат-бота, MapsAPI или иных HTTP-API, которые не рассматривались на лекциях;
- [продвинутый уровень] (Для навыка) Навык должен как минимум быть отправлен на модерацию. Желательно, чтобы его работу можно было бы увидеть с любого устройства.

В качестве примеров проектов можно привести следующие

1. Сайт библиотеки, которая позволяет администратору загружать книги, а зарегистрированным пользователям скачивать их (в том числе через API).
2. Сайт агентства недвижимости, где риэлтор может добавлять объекты недвижимости, а пользователи могут их просматривать и фильтровать по некоторым параметрам.
3. Сайт магазина/аптеки/склада, который позволяет администратору создавать товары, а пользователям просматривать и добавлять их к заказу.
4. Навык, помогающий составить карту интересных мест для экскурсии. Итог диалога – ссылка на StaticAPI-карту. Это карта с расставленными на ней пронумерованными метками интересных для посещения мест и сопровождающий комментарий, описывающий эти метки. Диалог для карты строится из поисковых запросов, перебора ответов и команды «добавить найденный объект в карту» (и, возможно, комментария к нему).
5. Навык, сравнивающий информацию по районам города. Карта, магазины, аптеки, поликлиники, кинотеатры и т. п. Можно сделать сравнение районов по наличию/близости тех или иных объектов. Это полезно, например, при выборе места для аренды или покупки квартиры.
6. Географический тест. Навык содержит информацию по различным объектам (страны, города, моря и т. д.), показывает их спутниковые снимки с меткой над объектом, просит ввести название объекта и ведёт статистику правильных и неправильных ответов. По аналогии можно предварительно запустить обучение. Исходные данные для тестов содержатся в файле, можно иметь несколько тестов разной сложности.

Предложите ученикам перед началом работы над проектом сделать его краткое описание, а также схематично показать различные переходы между состояниями объектов в проекте. Это поможет им оценить масштаб работы, которую они собираются сделать, а также выбрать правильный путь для проектирования структуры приложения

Оптимальный размер группы для выполнения данного проекта - 2 человека. Однако с учётом того, что групповую работу во время дистанционных занятий организовать сложно, ученики могут делать индивидуальный проект. В таком случае попросите их уделить особое внимание качественному ведению репозитория с кодом проекта.

Пока без комментариев

Комментировать