



Темы дипломных работ

Интернет-магазин	2
Новостной агрегатор.....	3
Телеграм-бот (пиццерия).....	4
Телеграм-бот (мемориальная архитектура).....	5
Файлообменник.....	6

Интернет-магазин

Необходимо написать веб-приложение, в котором можно просматривать список товаров с базовой информацией, а также детальную информацию о каждом из них конкретно, добавлять товары в корзину и осуществлять покупку (без подключения платежной системы). Добавлять и покупать товары могут только залогиненные пользователи. Добавление / удаление товаров в систему осуществляется только администратором.

Требования:

1. Использовать любой фреймворк из списка: Django, Flask, FastAPI, Aiohttp
2. Использовать любую СУБД из списка: MySQL, PostgreSQL, MongoDB
3. Реализовать регистрацию и вход в систему одним из следующих способов: Session, Access Token, JWT
4. Каждый исключительный случай должен возвращать ответ с соответствующим статусом.
5. Перед сохранением или обновлением данные от пользователя необходимо валидировать, и, в случае ошибок, вернуть ответ пользователю с соответствующими сообщениями по каждому из неверных полей.
6. Добавить тесты (достаточно будет unit- и интеграционных-тестов)

Новостной агрегатор

Необходимо написать веб-приложение, которое будет парсить и отображать список статей с существующих новостных ресурсов (минимум 2). У каждой новости должен быть постер, заголовок и краткое описание, а на странице конкретной новости можно будет получить всю необходимую информацию (текстовый контент, картинки, ссылки на источник и т/п). Для публикации собственных статей пользователю необходимо зарегистрироваться и, создав статью, отправить ее на публикацию. Администратор будет просматривать запросы на публикацию и выносить вердикт. В случае отказа, администратору необходимо написать причину.

Требования:

1. Использовать любой фреймворк из списка: Django, Flask, FastAPI, Aiohttp
2. Для парсинга использовать: selenium, scrapy, beautiful-soup
3. Для фоновых задач использовать: celery, cron, threading, BackgroundTasks
4. Использовать любую СУБД из списка: MySQL, PostgreSQL, MongoDB
5. Реализовать регистрацию и вход в систему одним из следующих способов: Session, Access Token, JWT
6. Каждый исключительный случай должен возвращать ответ с соответствующим статусом.
7. Перед сохранением или обновлением данные от пользователя необходимо валидировать, и, в случае ошибок, вернуть ответ пользователю с соответствующими сообщениями по каждому из неверных полей.
8. Добавить тесты (достаточно будет unit- и интеграционных-тестов)

Telegram-бот (пиццерия)

Необходимо написать телеграм-бот, который позволяет просматривать каталог пицц, добавлять их в корзину, указывая нужные размеры и вид, и оформлять заказ. Добавление и редактирование доступных вариантов, цен и т/п должно происходить в админке веб-приложения, к которой у покупателей нет доступа.

Требования:

1. Использовать любой фреймворк из списка: Django, Flask, FastAPI, Aiohttp
2. Использовать Telegram API: telebot, aiogram
3. Использовать любую СУБД из списка: MySQL, PostgreSQL, MongoDB
4. Реализовать регистрацию и вход в систему одним из следующих способов: Session, Access Token, JWT
5. Каждый исключительный случай должен возвращать ответ с соответствующим статусом.
6. Перед сохранением или обновлением данные от пользователя необходимо валидировать, и, в случае ошибок, вернуть ответ пользователю с соответствующими сообщениями по каждому из неверных полей.
7. Добавить тесты (достаточно будет unit- и интеграционных-тестов)

Telegram-бот (мемориальная архитектура)

Необходимо написать телеграм-бот, который позволяет просматривать каталог товаров, добавлять их в корзину и оформлять заказ, после этого администратору должно прийти оповещение и контакты для связи с покупателем. Добавление и редактирование доступных вариантов, цен и т/п должно происходить в админке веб-приложения, к которой у покупателей нет доступа.

Требования:

1. Использовать любой фреймворк из списка: Django, Flask, FastAPI, Aiohttp
2. Использовать Telegram API: telebot, aiogram
3. Использовать любую СУБД из списка: MySQL, PostgreSQL, MongoDB
4. Реализовать регистрацию и вход в систему одним из следующих способов: Session, Access Token, JWT
5. Каждый исключительный случай должен возвращать ответ с соответствующим статусом.
6. Перед сохранением или обновлением данные от пользователя необходимо валидировать, и, в случае ошибок, вернуть ответ пользователю с соответствующими сообщениями по каждому из неверных полей.
7. Добавить тесты (достаточно будет unit- и интеграционных-тестов)

Файлообменник

Необходимо написать веб-приложение, которое позволяет загружать файлы любых форматов, формировать временную ссылку на файл, по которой пользователи, имеющие эту ссылку, могут его скачать. Также необходимо добавить функционал, по которому можно разрешить скачивание файла определенному списку пользователей, а всем остальным запретить.

Требования:

1. Использовать любой фреймворк из списка: Django, Flask, FastAPI, Aiohttp
2. Использовать любую СУБД из списка: MySQL, PostgreSQL, MongoDB
3. Реализовать регистрацию и вход в систему одним из следующих способов: Session, Access Token, JWT
4. Каждый исключительный случай должен возвращать ответ с соответствующим статусом.
5. Перед сохранением или обновлением данные от пользователя необходимо валидировать, и, в случае ошибок, вернуть ответ пользователю с соответствующими сообщениями по каждому из неверных полей.
6. Добавить тесты (достаточно будет unit- и интеграционных-тестов)