Постановки задач для итоговой аттестации

Создайте новый Django-проект и отдельный репозиторий для него. Произведите первоначальную конфигурацию проекта и предоставьте доступ к этому репозиторию вашему куратору.

Этап 1. Создание и проектирование

Задача 1. Разработка модели хранения данных

Внимательно изучите ТЗ на проект и разработайте модель хранения данных: спроектируйте схему таблиц в БД и связей между ними, а также укажите, какие модели должны быть в проекте.

Для проектирования таблиц стремитесь соблюдать язык UML. Используйте любой инструмент для визуального представления, который будет удобен, например Flowchart Maker & Online Diagram Software, DbDesigner.net: Free Database Designer, Diagramming Software & Team Collaboration Tools | Gliffy Diagram Apps.

Задача 2. Разработка структуры URL на сайте

Разработайте структуру URL-ссылок на вашем сайте. Придумайте, по каким адресам какие разделы сайта будут доступны, что должно быть вынесено в API, какие маршруты используются для обновления информации, какие для добавления и так далее.

Результат представьте в виде таблицы со следующими полями:

- Раздел.
- Страница.
- Описание.
- HTTP-метод.
- URL.
- Комментарий.

Например:

Каталог	Детальная страница	Добавление нового отзыва	POST	/somewhere/to/catalog/<id>/reviews	—

Задача 3. Разработка модели сервисов на сайте

Изучив ТЗ, вынесите отдельные сервисы, которые должны быть реализованы на сайте, придумайте им название (название класса и интерфейса) и зарегистрируйте пустые заглушки в сервисах, создайте также эти пустые интерфейсы и классы. Если для каких-то сервисов очевидны некоторые методы, то сразу вынесите их и создайте пустой метод-заглушку.

Обязательные сервисы:

- Получение административных настроек.
- Список просмотренных товаров.

- Добавление товара в корзину.
- Добавление отзыва к товару.
- Оплата.

Вы можете вынести и свои отдельные сервисы. Сервисом в данном ТЗ называется любой класс, не являющийся моделью или любой другой частью фреймворка (событие, уведомление и так далее), созданный для решения определённой задачи.

Задача 4. Интеграция вёрстки шаблона сайта

Интегрируйте вёрстку основного шаблона сайта в Django-проект. Разбейте основной макет на header и footer, создайте отдельные части Jinja-шаблона. Разметьте шаблон для возможности его дальнейшего расширения: управление блоком head, управление телом страницы, управление подключаемыми скриптами, управление заголовком страницы.

Задача 5. Подключение административной панели и БД

Подключите админ-панель Django к проекту.

Настройте подключение базы данных PostgreSQL (для разработки можно использовать SQLite).

Задача 6. Разработка верхнего меню и футера

Интегрируйте вёрстку верхнего меню и футера. Укажите все необходимые ссылки для пунктов меню. Реализуйте различные пункты меню для авторизованного и неавторизованного пользователя (сам процесс авторизации и регистрации не реализуется в рамках этой задачи).

Этап 2. Главная страница и заглушки

Задача 7. Создание модели категорий каталога

Создайте необходимые модели, связи, миграции и интеграции в админ-панель для категорий каталога.

Задача 7.1. Разработка меню категорий каталога

Интегрируйте вёрстку меню выбора категорий каталога. В это меню должны попадать только активные (отдельный boolean-признак) категории товаров, в которых есть активные товары. При этом категории должны быть отсортированы в порядке возрастания поля «индекс сортировки».

Задача 8. Создание модели ролей

Создайте необходимые модели, связи, миграции и интеграции в админ-панель для ролей пользователя на сайте.

Задача 9. Создание модели пользователей

Создайте необходимые модели, связи, миграции и интеграции в админ-панель для хранения информации о пользователе на сайте.

Задача 9.1. Разработка страниц авторизации/регистрации и восстановления пароля

Интегрируйте вёрстку в механизм стандартной авторизации/регистрации и восстановления пароля.

Задача 10. Разработка страницы управления настройками в административном разделе

Разработайте страницу управления настройками в административном разделе и на сайте. Реализуйте весь сервис получения настроек.

Список настроек и исходные значения должны браться из конфигурационного файла. Если тип исходного значения boolean, то параметр должен быть представлен в виде чекбокса, в другом случае — в виде строки для ввода.

Задача 11. Разработка заглушки сервиса добавления товара в корзину

Добавьте сервис добавления товара в корзину, создайте методы-заглушки для работы этого сервиса. Сервис должен позволять:

- добавить товар в корзину;
- убрать товар из корзины;
- изменить количество товара в корзине;
- получить список товаров в корзине;
- получить количество товаров в корзине.

Опишите эти методы в сервисе и в его интерфейсе, реализуйте возврат статичных данных этими методами.

Задача 12. Разработка заглушки сервиса добавления отзыва к товару

Добавьте сервис добавления отзыва к товару, создайте методы-заглушки для работы этого сервиса. Сервис должен позволять:

- добавить отзыв к товару;
- получить список отзывов к товару;
- получить скидку на корзину;
- получить количество отзывов для товара.

Опишите эти методы в сервисе и в его интерфейсе, реализуйте возврат статичных данных этими методами.

Задача 13. Разработка заглушки сервиса интеграции с сервисом оплаты

Добавьте сервис интеграции с сервисом оплаты, создайте методы-заглушки для работы этого сервиса. Сервис должен позволять:

- оплатить указанный заказ;
- получить статус оплаты заказа.

Опишите эти методы в сервисе и в его интерфейсе, реализуйте возврат статичных данных этими методами.

Задача 14. Создание моделей товаров с ценами

Создайте необходимые модели, связи, миграции и интеграции в админ-панель для товаров с цен на сайте.

Этап 3. Каталог

Задача 15. Разработка каталога товаров

Интегрируйте вёрстку каталога товаров в соответствии с ТЗ. На странице отображается список товаров из выбранной категории товаров. В список попадают только активные товары, подходящие под фильтр.

Задача 15.1. Разработка фильтра в каталоге товаров

Интегрируйте вёрстку и реализуйте фильтр в каталоге товаров в соответствии с ТЗ. Фильтр по цене должен представлять из себя два значения для ввода (от и до) и/или слайдер. Фильтрация по названию и другим текстовым полям должна осуществляться по вхождению указанной подстроки в товар или его описание.

При поиске по продавцу и производителю фильтр реализуется в виде списков со множественным выбором.

Для характеристик типа boolean реализуется возможность выбора из трёх вариантов: «да», «нет», «не учитывать». Для характеристик типа «список» реализуется группа чекбоксов: если ни один не выбран, то фильтр не применяется, если выбран один или несколько, то в список попадают товары, у которых значение этой характеристики попадает в список выбранных значений.

Остальные характеристики фильтруются как строка.

Задача 15.2. Разработка сортировки в каталоге товаров

Интегрируйте вёрстку и реализуйте сортировку в каталоге товаров в соответствии с ТЗ.

Задача 15.3. Разработка постраничной навигации в каталоге товаров

Интегрируйте вёрстку и реализуйте постраничную навигацию в каталоге товаров в соответствии с ТЗ.

Задача 15.4. Интеграция с сервисом добавления отзывов к товару

Для вывода количества отзывов о товаре интегрируйтесь с сервисом добавления отзывов к товару.

Задача 15.5. Интеграция с сервисом добавления товара в корзину

При нажатии на кнопку «Купить» выполните необходимый метод в сервисе добавления товаров в корзину и обработайте результат его работы.

Задача 16. Разработка каталога топ-товаров на главной странице

Интегрируйте вёрстку каталога топ-товаров на главной странице в соответствии с ТЗ.

Задача 17. Разработка детальной страницы товара

Интегрируйте вёрстку детальной страницы товара в соответствии с ТЗ.

Данные о товаре кешируются на сутки (параметр берётся из сервиса получения настроек), при изменении товара должен осуществляться сброс кеша на этой странице.

Задача 17.1. Интеграция с сервисом добавления товара в корзину

При нажатии на кнопку «Купить» выполните необходимый метод в сервисе добавления товаров в корзину и обработайте результат его работы.

Задача 17.2. Интеграция с сервисом добавления отзывов к товару

Для вывода отзывов о товаре интегрируйтесь с сервисом добавления отзывов к товару.

Задача 18. Создание модели отзывов

Создайте необходимые модели, связи, миграции и интеграции в админ-панель для отзывов к товарам на сайте.

Задача 18.1. Разработка сервиса добавления отзывов

Интегрируйте вёрстку блока отзывов на детальной странице товара, для получения отзывов используйте сервис добавления отзывов.

Также интегрируйте форму добавления отзыва, при добавлении отзыва вызывайте соответствующий метод сервиса. При возникновении ошибки валидации эти ошибки должны быть выведены над формой.

При нажатии на кнопку «Показать ещё» должны подгружаться отзывы. Кнопка должна скрыться, если уже загружены все отзывы.

Реализуйте сервис добавления отзывов и все доступные в нём методы:

- добавить отзыв к товару;
- получить список отзывов к товару;
- получить количество отзывов для товара.

Задача 19. Разработка части страницы личного кабинета

Интегрируйте вёрстку сводной страницы личного кабинета в соответствии с ТЗ. Интегрируйте вёрстку только блока информации о пользователе, остальные блоки оставьте в виде статики.

Задача 19.1. Разработка страницы профиля

Интегрируйте и реализуйте страницу профиля в соответствии с ТЗ. Для поля «Телефон» необходимо реализовать маску подстановки в виде: +7 *(___)* ___ - __ - __ или в похожем. При этом храниться номер должен в виде числа (10 цифр) без кода страны (цифры 7).

В случае возникновения ошибки валидации текст ошибки должен быть подписан отдельно к каждому полю либо над формой, если ошибка не относится к форме в целом.

После успешного изменения полей профиля пользователя информация должна обновиться на всём сайте.

Этап 4. Оформление заказов, админка, оплата

Задача 20. Создание модели корзины

Создайте необходимые модели, связи, миграции и интеграции в админ-панель для корзины пользователей на сайте.

Задача 20.1. Разработка сервиса добавления товара в корзину с реальными данными

Реализуйте сервис добавления товаров в корзину, реализуйте его методы:

- добавить товар в корзину;
- убрать товар из корзины;
- изменить количество товара в корзине;
- получить список товаров в корзине;
- получить количество товаров в корзине.

Вне зависимости от того, авторизован пользователь или нет, корзина должна быть привязана к нему. Если пользователь покинет сайт, а потом вернётся через какое-то время, корзина пропасть не должна. А после авторизации корзина неавторизованного пользователя должна попасть в корзину этого авторизованного. Если у него были до этого отложены товары в корзине, то должно произойти слияние корзин.

Метод добавления

Этот метод должен принимать два параметра: товар и количество для добавления (по умолчанию 1). Этот метод должен сам определить, есть ли указанный товар в корзине, и если есть, то самостоятельно «пробрасывать» себя в метод изменения количества товара в корзине, если нет, то добавлять товар с указанным количеством.

Метод изменения количества

Этот метод должен принимать два обязательных параметра: товар и на сколько изменить. При этом если изменение отрицательное, то количество должно уменьшаться вплоть до удаления товара из корзины.

Задача 20.2. Разработка корзины в шапке сайта

Интегрируйте вёрстку блока корзины в шапке сайта в соответствии с ТЗ, используя разработанный сервис работы с корзиной пользователя.

Задача 20.3. Разработка страницы корзины

Интегрируйте и разработайте корзину пользователя в соответствии с ТЗ.

Задача 21. Создание модели заказов

Создайте необходимые модели, связи, миграции и интеграции в админ-панель для заказов пользователей на сайте.

Задача 21.1. Разработка страницы оформления заказа

Интегрируйте и разработайте страницу оформления заказа в соответствии с ТЗ.

Задача 21.2. Оформление заказа: параметры пользователя

Интегрируйте шаг оформления заказа «Параметры пользователя» в соответствии с ТЗ. Так же, как и для профиля пользователя, для поля «Телефон» реализуйте маску подстановки в виде +7 *(___)* ___ - __ - __ или в похожем. При этом храниться номер должен в виде числа (10 цифр) без кода страны (цифры 7).

Задача 21.3. Оформление заказа: способ доставки

Интегрируйте шаг оформления заказа «Способ доставки» в соответствии с ТЗ. Все необходимые параметры берутся из сервиса получения настроек.

Задача 21.4. Оформление заказа: страница оплаты

Интегрируйте шаг оформления заказа «Страница оплаты» в соответствии с ТЗ.

Задача 21.5. Интеграция с сервисом интеграции с сервисом оплаты

Интегрируйтесь с сервисом интеграции с сервисом оплаты (в этом предложении нет опечаток) после подтверждения оформления заказа пользователем, вызвав необходимый метод отправки оплаты.

Задача 22. Разработка страницы «История заказов»

Интегрируйте вёрстку страницы «История заказов» в соответствии с ТЗ.

Задача 22.1. Разработка страницы истории заказов — детальная страница заказа

Интегрируйте вёрстку страницы «История заказов — детальная страница» в соответствии с ТЗ. Если заказ не был оплачен по какой-то причине, то интегрируйте блок выбора способа оплаты, как на странице оформления заказа, и вызовите соответствующий метод сервиса оплаты.

Задача 22.2. Разработка истории заказов на сводной странице личного кабинета

Интегрируйте вёрстку блока истории заказов на сводной странице личного кабинета пользователя.

Задача 23. Разработка сервиса оплаты

Разработайте отдельный метод API на своём сайте для обработки запросов оплаты в соответствии с ТЗ. API в качестве параметров принимает: номер заказа, номер карты, сумму к оплате.

Разработайте сервис интеграции с сервисом оплаты, реализовав его методы:

- оплатить указанный заказ;
- получить статус оплаты заказа.

Метод «оплатить заказ»

В качестве параметра принимает заказ и добавляет его на оплату в очередь.

Создайте команду с одним параметром — ID заказа, с помощью которой можно также добавить заказ на оплату в очередь.

Реализуйте обработчик очереди (job), который будет запрашивать оплату у сервиса «Фиктивной оплаты» посредством запроса к API (используя ваш API).

Реализуйте сервис «Фиктивной оплаты». Этот сервис должен содержать один метод — «оплатить заказ», который в качестве параметра принимает три значения: номер заказа, номер карты, сумма к оплате. Логика работы сервиса из ТЗ: «Если введённый номер чётный и не заканчивается на 0, то оплата подтверждается, если введённый номер чётный и заканчивается на 0, то сервис генерирует случайную ошибку оплаты».

После подтверждения или ошибки обработчик должен установить соответствующий статус оплаты заказу.

Этап 5. Отладка и сдача проекта

Произведите отладку и исправление ошибок на проекте.

Создайте отдельную фикступу с демонстрационным набором всех возможностей вашего проекта и его презентацией.