# Постановка задач для дипломного проекта

Создайте новый Django-проект, создайте отдельный репозиторий для него. Произведите первоначальную конфигурацию проекта и предоставьте доступ к этому репозиторию всем участникам проекта и вашему преподавателю.

## Этап 1. Создание и проектирование

### Задача 1. Разработка модели хранения данных

Всей командой внимательно изучите ТЗ на проект и разработайте модель хранения данных, то есть 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, но главное, чтобы схема была понятна всем участникам проекта. Используйте любой инструмент для визуального представления, который будет удобен всей команде: [draw.io](https://www.draw.io/), [dbdesigner.net](https://app.dbdesigner.net/), [gliffy.com](https://www.gliffy.com/) или любой другой.

### 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 таблички 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

Изучая ТЗ, вынесите отдельные сервисы, которые должны быть реализованы на сайте, придумайте им названия (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 их вынесите и создайте пустой метод-заглушку. Список обязательных сервисов: «Получение административных настроек», «Список просмотренных товаров», «Скидки на товар», «Добавление товара в корзину», «Добавление отзыва к товару», «Список сравниваемых товаров», «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-панели 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».

Меню категорий должно кешироваться на сутки (параметр берётся из сервиса получения настроек). При этом кеш должен сбрасываться при изменении полей категорий, добавлении или удалении категории. Или при сбросе кеша в административном интерфейсе.

### Задача 8. Создание модели баннеров

Создайте необходимые модели, связи, миграции и интеграции в админ-панель для баннеров на главной странице.

#### Задача 8.1. Разработка баннера на главной странице

Интегрируйте вёрстку блока с баннерами на главной странице. В блоке с баннерами выводятся три случайных активных баннера из списка доступных. Баннеры представляют собой слайдер с прокруткой.

Выбранные баннеры должны быть закешированы на десять минут (параметр берётся из сервиса получения настроек). При изменении полей баннеров, добавлении или удалении баннера кеш этого блока должен быть сброшен.

### Задача 9. Создание модели ролей

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

### Задача 10. Создание модели пользователей

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

#### Задача 10.1. Разработка страниц авторизации / регистрации / восстановления пароля

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

### Задача 11. 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.

Также на странице настроек должен быть предусмотрен отдельный блок для сброса кеша различных разделов на сайте. Создайте отдельную кнопку для сброса всего кеша на сайте.

### Задача 12. Разработка заглушки сервиса добавления в список просмотренных товаров

Добавьте сервис добавления в список просмотренных товаров, создайте методы-заглушки для работы этого сервиса. Сервис должен позволять:

* добавлять товар в список сравнения;
* убирать товар из этого списка;
* получать список товаров, добавленных к сравнению (с возможностью ограничить количество, по умолчанию максимум ― три первых добавленных);
* получать количество товаров в этом списке.

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

### Задача 13. Разработка заглушки сервиса получения скидок на товары

Добавьте сервис получения скидок на товары, создайте методы-заглушки для работы этого сервиса. Сервис должен позволять:

* получить все скидки на указанный список товаров или на один товар;
* получить приоритетную скидку на указанный список товаров или на один товар;
* рассчитать цену со скидкой на товар с дополнительным необязательным параметром «Цена товара».

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

### Задача 14. 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 15. 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 16. Разработка заглушки сервиса добавления в список просмотренных товаров

Добавьте сервис добавления в список просмотренных товаров, создайте методы-заглушки для работы этого сервиса. Сервис должен позволять:

* добавить товар к списку просмотренных товаров;
* удалить товар из списка просмотренных товаров;
* узнать, есть ли товар уже в списке просмотренных;
* получить список просмотренных товаров;
* получить количество просмотренных товаров.

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

### Задача 17. 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 18. Создание модели продавцов

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

#### Задача 18.1. Разработка детальной страницы продавца

Разработайте и интегрируйте вёрстку детальной страницы продавца в соответствии с ТЗ. Данные о продавце должны быть закешированы на сутки, а топ его товаров должен быть закеширован на один час (параметр берётся из сервиса получения настроек). При этом при изменении продавца должен быть сброшен кеш на этой странице.

### Задача 19. Создание моделей товаров и цен

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

## Этап 3. Каталог (часть 1)

### Задача 20. 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.

Список товаров кешируется на сутки (параметр берётся из сервиса получения настроек), при изменении товаров, их добавлении или изменении должен осуществляться сброс кеша. В административном разделе должна быть возможность сбросить кеш каталога.

#### Задача 20.1. 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 фильтрация 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 20.2. Разработка сортировки в каталоге товаров

Интегрируйте вёрстку и реализуйте сортировку в каталоге товаров в соответствии с ТЗ. Сортировка также должна учитываться при формировании кеша каталога.

#### Задача 20.3. Разработка постраничной навигации в каталоге товаров

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

#### Задача 20.4. Интеграция с сервисом добавления отзывов к товару

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

#### Задача 20.5. 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 20.6. Интеграция с сервисом списка просмотренных товаров

При выводе иконки «Добавить в список просмотренных» учитывайте данные из сервиса списка просмотренных товаров. При нажатии на иконку выполните необходимый метод в сервисе.

#### Задача 20.7. Интеграция с сервисом скидок на товар

Примените сервис получения скидок на товары для корректного вывода цены на товар с учётом скидки.

### Задача 21. Разработка каталога топ-товаров на главной странице

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

Список товаров кешируется на сутки (параметр берётся из сервиса получения настроек), при изменении товаров, их добавлении или изменении должен осуществляться сброс кеша. В административном разделе должна быть возможность сбросить кеш каталога.

### Задача 22. 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 22.1. Разработка блока «Продавцы» на детальной странице товара

Интегрируйте вёрстку блока продавцов на детальной странице товара в соответствии с ТЗ.

#### Задача 22.2. 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 22.3. Интеграция с сервисом списка просмотренных товаров

При заходе пользователя на эту страницу выполните необходимый метод сервиса списка просмотренных товаров.

#### Задача 22.4. Интеграция с сервисом скидок на товар

Примените сервис получения скидок на товар для корректного вывода цены на товар с учётом скидки.

### Задача 23. Создание модели отзывов

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

#### Задача 23.1. Разработка сервиса добавления отзывов

Интегрируйте вёрстку блока «Отзывы» 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 ссылку «Показать ещё» должна быть осуществлена подгрузка отзывов. Кнопка должна скрыться, если уже загружены все отзывы.

Реализуйте сервис добавления отзывов и все методы, доступные в нём:

* добавить отзыв к товару,
* получить список отзывов к товару,
* получить количество отзывов для товара.

### Задача 24. Разработка сервиса добавления товаров в список сравнения

Реализуйте сервис добавления товаров в список сравнения, опишите все методы, доступные в нём:

* добавлять товар в список сравнения;
* убирать товар из этого списка;
* получать список товаров, добавленных к сравнению (с возможностью ограничить количество, по умолчанию максимум ― три первых добавленных);
* получать количество товаров в этом списке.

##### Добавление товара в список сравнения

Метод должен проверять, если ли указанный товар в списке сравнения, и добавлять его в список, если такового нет.

##### Список товаров, добавленных к сравнению

Метод принимает один необязательный параметр ― сколько товаров возвращать. По умолчанию ― три. Метод возвращает список из указанного количества в порядке добавления товаров в список просмотренных.

#### Задача 24.1. Разработка страницы сравнения товаров

Интегрируйте вёрстку страницы сравнения товаров в соответствии с ТЗ, используя разработанный сервис.

#### Задача 24.2. Разработка блока сравнения товаров в шапке сайта

Интегрируйте вёрстку блока сравнения товара в шапке сайта в соответствии с ТЗ, используя разработанный сервис.

### Задача 25. 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 25.1. 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 виде. При этом храниться номер должен в виде числа (десять цифр) без кода страны (без 7-ки).

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. Скидки, оформление заказов, админка, импорт, оплата

### Задача 26. Создание модели корзины

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

#### Задача 26.1. 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.

При этом 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 отрицательно, то количество должно уменьшаться вплоть до удаления товара из корзины.

#### Задача 26.2. Разработка корзины в шапке сайта

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

#### Задача 26.3. Разработка страницы корзины

Интегрируйте и разработайте корзину пользователя в соответствии с ТЗ.

### Задача 27. Создание модели заказов

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

#### Задача 27.1. Разработка страницы оформления заказа

Интегрируйте и разработайте страницу оформления заказа в соответствии с ТЗ.

#### Задача 27.2. Оформление заказа — параметры пользователя

Интегрируйте шаг оформления заказа «Параметры пользователя» в соответствии с ТЗ. Так же как и для профиля пользователя, для поля «Телефон» необходимо реализовать маску подстановки в виде «*+7 (\_\_\_) \_\_\_ - \_\_ - \_\_*» или в похожем виде. При этом храниться номер должен в виде числа (десять цифр) без кода страны (без 7-ки).

#### Задача 27.3. Оформление заказа — способ доставки

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

#### Задача 27.4. Оформление заказа — страница оплаты

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

#### Задача 27.5. 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 28. Разработка страницы «История заказов»

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

#### Задача 28.1. Разработка страницы истории заказов ― детальная страница заказа

Интегрируйте вёрстку страницы «История заказов ― детальная страница» в соответствии с ТЗ. В случае 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 28.2. Разработка истории заказов на сводной странице личного кабинета

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

### Задача 29. Создание модели истории просмотров

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

#### Задача 29.1. Разработка сервиса добавления в список просмотренных товаров

Разработайте сервис добавления в список просмотренных товаров, реализовав его методы:

* добавить товар к списку просмотренных товаров;
* удалить товар из списка просмотренных товаров;
* узнать, есть ли товар уже в списке просмотренных;
* получить список просмотренных товаров;
* получить количество просмотренных товаров.

##### Добавление товара в список просмотренных

При добавлении нового товара, если этот товар есть в списке просмотренных, он со своего места перемещается в самое начало списка. То есть в этом списке не может быть двух одинаковых товаров. Если этот товар и есть уже на последнем месте, то ничего не происходит (например, пользователь нажал F5).

##### Получение списка просмотренных товаров

Метод принимает необязательный параметр, сколько последних товаров возвратить (по умолчанию ― 20).

#### Задача 29.2. Разработка страницы «История просмотров»

Интегрируйте вёрстку страницы «История просмотров» в личном кабинете в соответствии с ТЗ.

#### Задача 29.3. Разработка блока «История просмотров» на сводной странице личного кабинета

Интегрируйте вёрстку блока «История просмотров» на сводной странице личного кабинета в соответствии с ТЗ.

### Задача 30. L10n и i18n в админке

-

### Задача 31. L10n и i18n в вёрстке

На всех страницах

### Задача 32. L10n и i18n в бэкенде

-

### Задача 33. Создание модели скидок

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

#### Задача 33.1. Разработка страницы списка скидок

Интегрируйте вёрстку страницы списка скидок на сайте в соответствии с ТЗ. В списке выводятся все активные скидки, для которых включено отображение на странице списка скидок.

#### Задача 33.2. Разработка детальной страницы скидки

Интегрируйте вёрстку детальной страницы скидки на сайте в соответствии с ТЗ.

### Задача 34. Разработка страницы списка скидок в административном разделе

Разработайте раздел со списком скидок в административном интерфейсе в соответствии с ТЗ.

#### Задача 34.1. Разработка страницы создания и редактирования скидки в административном разделе

Разработайте страницы создания и редактирования скидки в соответсвии с ТЗ.

#### Задача 34.2. Интеграция сервиса скидок с реальными данными по скидкам

Разработайте сервис скидки в соответствии с ТЗ, реализовав его методы:

* получить все скидки на указанный список товаров или на один товар;
* получить приоритетную скидку на указанный список товаров или на один товар;
* получить скидку на корзину;
* рассчитать цену со скидкой на товар с дополнительным необязательным параметром ― цена товара.

##### Метод расчёта цены

Принимает два параметра: «Товар» и необязательный параметр «Исходная цена». Если исходная цена товара не указана, то берётся максимальная стоимость товара у продавцов и расчёт скидки ведётся относительно неё. Метод должен, в соответствии с правилами скидки, вернуть корректную цену на товар.

##### Метод получения скидки на корзину

Метод получает в качестве параметра коллекцию позиций корзины вида «товар и исходная цена». И на их основе пытается применить самую приоритетную скидку на корзину или на набор. Если такая скидка есть, то применяется она; если нет, то тогда метод проверяет каждый товар в скидках на товар и применяет к каждому товару подошедшую приоритетную скидку. В качестве результата работы метод возвращает позиции корзины вида «товар, исходная цена, цена со скидкой, применена ли скидка».

### Задача 35. Разработка страницы проведения импорта

Интегрируйте страницу проведения импорта в соответствии с ТЗ. На странице должна выводиться информация о запущенном импорте. Если такой есть и он ещё не выполнен, то возможности запустить ещё один импорт не должно быть.

#### Задача 35.1. Разработка импорта

Реализуйте команду для запуска импорта, реализуйте отдельный Job, который и будет производить импорт в соответсвии с ТЗ. Импорт должен выполняться в фоновом режиме с использованием механизма очередей. При этом должна быть возможность получить статус выполнения импорта («Выполнен», «В процессе выполнения», «Завершён с ошибкой»).

Импорт должен вести лог своей работы в отдельных файлах, каждый файл должен быть помечен датой и временем начала выполнения импорта.

Если запущен другой процесс импорта, то импорт не должен запускаться, а должен сразу возвращать ошибку: «Предыдущий импорт ещё не выполнен. Пожалуйста, дождитесь его окончания».

### Задача 36. 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 «Фиктивная оплата» посредством запроса к API (используя ваш API).

Реализуйте сервис фиктивной оплаты. Этот сервис должен содержать один метод ― «Оплатить заказ». 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 нечётный 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 фикстуру с набором данных для демонстрации всех возможностей вашего проекта и его презентации.