



# Техническое описание дипломного проекта

Ваш дипломный проект — сайт «Продуктовый помощник». Это онлайн-сервис, где пользователи смогут публиковать рецепты, подписываться на публикации других пользователей, добавлять понравившиеся рецепты в список «Избранное», а перед походом в магазин скачивать сводный список продуктов, необходимых для приготовления одного или нескольких выбранных блюд.

## Исходники

Студенты веб-факультета Яндекс.Практикума подготовили для вас HTML-шаблоны и JS-код, они сохранены в репозитории `foodgram-project`. Склонируйте его и делайте проект в нём.

## HTML

Макеты, по которым сверстаны шаблоны, можно посмотреть на *Figma.com*:  
<https://www.figma.com/file/HNEJ68zF1bCa7Dx8ZsGxFh/Продуктовый-помощник-Final?node-id=0%3A>

Список шаблонов:

- Главная страница проекта для неавторизованного пользователя.
- Главная страница проекта для авторизованного пользователя.
- Персональная страница пользователя (рецепты одного автора).
- Страницы для регистрации/авторизации/сброса пароля/изменения пароля.
- Страница рецепта для автора

- Страница рецепта для авторизованного читателя
- Страница рецепта для неавторизованного читателя
- Страница создания рецепта.
- Страница редактирования/удаления рецепта.
- Страница подписок.
- Страница избранных рецептов.
- Страница списка покупок.
- Формы:
  - зарегистрироваться,
  - войти на сайт,
  - сбросить пароль,
  - изменить пароль.

## JavaScript

При определённых действиях пользователя с веб-страницы будут отправляться запросы к бэкенду. Это делает внедрённый в страницу код **JavaScript**, язык сценариев, работающих на стороне клиента.

JS будет отправлять запросы в таких случаях:

- при добавлении рецепта в список избранного и удалении рецепта из списка;
- при добавлении рецепта в список покупок и удалении рецепта из списка;
- при подписке на автора и отмене подписки;
- при вводе названия ингредиента в форме создания рецепта: на сервер отправляется введённый в поле фрагмент текста, возвращаются найденные совпадения, которые выводятся в подсказки. При выборе нужного ингредиента подставляются соответствующие единицы измерения: «Яйца — шт», «Молоко — мл», «Сахар — г»;
- при изменении количества рецептов в «Списке покупок»: в запросе отправляется ID добавленного/удалённого рецепта, после успешного

запроса в шапке сайта обновляется счётчик.

## Список ингредиентов

Мы подготовили для вас небольшой список ингредиентов с единицами измерения, чтобы вам не пришлось вручную заполнять базу. Список доступен в репозитории `foodgram-project` в форматах JSON и CSV.

## Рецепт

Рецепт должен описываться такими полями:

- Автор публикации (пользователь).
- Название.
- Картинка.
- Текстовое описание.
- Ингредиенты: продукты для приготовления блюда по рецепту. Множественное поле, выбор из предустановленного списка, с указанием количества и единицы измерения.
- Тег (можно установить несколько тегов на один рецепт, выбор из предустановленных): *завтрак, обед, ужин*.
- Время приготовления в минутах.
- Slug (уникальная часть URL для рецепта) — **по желанию**.

Все поля обязательны для заполнения.

## Ингредиент

Данные об ингредиентах хранятся в нескольких связанных таблицах. В результате на стороне пользователя ингредиент должен описываться такими полями:

- Название.
- Количество.
- Единицы измерения.

Все поля обязательны для заполнения.

## **Сервисы и страницы проекта**

Посмотрите HTML-шаблоны и дизайн-макеты, это поможет лучше ориентироваться в описании.

### **Главная страница**

Содержимое главной страницы — список рецептов, отсортированных по дате публикации (от новых к старым).

### **Страница рецепта**

На странице — полное описание рецепта, возможность добавить рецепт в избранное и в список покупок, возможность подписаться на автора рецепта.

### **Страница пользователя**

На странице — имя пользователя, все рецепты, опубликованные пользователем и возможность подписаться на пользователя.

### **Подписка на авторов**

Подписка на публикации доступна только авторизованному пользователю. Страница подписок доступна только владельцу.

Сценарий поведения пользователя:

1. Пользователь переходит на страницу другого пользователя или на страницу рецепта и подписывается на публикации автора кликом по кнопке «Подписаться на автора».
2. Пользователь переходит на страницу «Мои подписки» и просматривает список рецептов, опубликованных теми авторами, на которых он подписался. Сортировка записей — по дате публикации (от новых к старым).
3. При необходимости пользователь может отказаться от подписки на автора: переходит на страницу автора или на страницу его рецепта и нажимает «Отписаться от автора».

## Список избранного

Работа со списком избранного доступна только авторизованному пользователю. Список избранного может просматривать только его владелец.

Сценарий поведения пользователя:

1. Пользователь отмечает один или несколько рецептов кликом по кнопке «Добавить в избранное».
2. Пользователь переходит на страницу «Список избранного» и просматривает персональный список избранных рецептов.
3. При необходимости пользователь может удалить рецепт из избранного.

## Список покупок

Работа со списком покупок доступна авторизованным пользователям. Список покупок может просматривать только его владелец.

Сценарий поведения пользователя:

1. Пользователь отмечает один или несколько рецептов кликом по кнопке «Добавить в покупки».
2. Пользователь переходит на страницу **Список покупок**, там доступны все добавленные в список рецепты. Пользователь нажимает кнопку **Скачать список** и получает файл с суммированным перечнем и количеством необходимых ингредиентов для всех рецептов, сохранённых в «Списке покупок».
3. При необходимости пользователь может удалить рецепт из списка покупок.

Список покупок скачивается в формате *.txt* (или, **по желанию**, можно сделать выгрузку PDF).

При скачивании списка покупок ингредиенты в результирующем списке не должны дублироваться; если в двух рецептах есть сахар (в одном рецепте 5 г, в другом — 10 г), то в списке должен быть один пункт: *Сахар — 15 г*.

В результате список покупок может выглядеть так:

- Фарш (баранина и говядина) (г) — 600
- Сыр плавленый (г) — 200
- Лук репчатый (г) — 50
- Картофель (г) — 1000
- Молоко (мл) — 250
- Яйцо куриное (шт) — 5
- Соевый соус (ст. л.) — 8
- Сахар (г) — 230
- Растительное масло рафинированное (ст. л.) — 2
- Соль (по вкусу) — 4
- Перец черный (щепотка) — 3

**По желанию:** в список покупок можно вывести шапку и подвал (или что-то одно) с информацией о вашем проекте.

**По желанию:** можно настроить проект так, чтобы сервис «Список покупок» был доступен и неавторизованным пользователям. Работа с сессиями поможет решить эту задачу.

Документация по работе с сессиями в Django:

- Описание: <https://docs.djangoproject.com/en/3.1/topics/http/sessions/>
- Примеры:  
<https://docs.djangoproject.com/en/3.1/topics/http/sessions/#examples>

## Фильтрация по тегам

При нажатии на название тега выводится список рецептов, отмеченных этим тегом. Фильтрация может проводится по нескольким тегам в комбинации «или»: если выбраны несколько тегов — в результате должны быть показаны рецепты, которые отмечены хотя бы одним из этих тегов.

При фильтрации на странице пользователя должны фильтроваться только рецепты выбранного пользователя. Такой же принцип должен соблюдаться при фильтрации списка избранного.

## Регистрация и авторизация

В проекте должна быть доступна система регистрации и авторизации пользователей. Чтобы собрать весь код для управления пользователями воедино — создайте приложение **users**.

### Обязательные поля для пользователя:

- Логин
- Пароль
- Email

Валидация email при регистрации — **по желанию**.

### Уровни доступа пользователей:

- Гость (неавторизованный пользователь)
- Авторизованный пользователь
- Администратор

### Что могут делать неавторизованные пользователи

- Создать аккаунт.
- Просматривать рецепты на главной.
- Просматривать отдельные страницы рецептов.
- Просматривать страницы пользователей.
- Фильтровать рецепты по тегам.

### Что могут делать авторизованные пользователи

- Входить в систему под своим логином и паролем.
- Выходить из системы (разлогиниваться).
- Восстанавливать свой пароль.
- Менять свой пароль.
- Создавать/редактировать/удалять собственные рецепты

- Просматривать рецепты на главной.
- Просматривать страницы пользователей.
- Просматривать отдельные страницы рецептов.
- Фильтровать рецепты по тегам.
- Работать с персональным списком избранного: добавлять/удалять **чужие** рецепты, просматривать свою страницу избранных рецептов.
- Работать с персональным списком покупок: добавлять/удалять **любые** рецепты, выгружать файл со количеством необходимых ингредиентов для рецептов из списка покупок.
- Подписываться на публикации авторов рецептов и отменять подписку, просматривать свою страницу подписок.

## Что может делать администратор

Администратор обладает всеми правами авторизованного пользователя.

Плюс к этому он может:

- изменять пароль любого пользователя,
- создавать/блокировать/удалять аккаунты пользователей,
- редактировать/удалять **любые** рецепты,
- добавлять/удалять/редактировать ингредиенты.

## Настройки админки

В интерфейс админ-зоны нужно вывести необходимые поля моделей и настроить фильтры.

### Модели:

- Вывести все модели с возможностью редактирования и удаление записей.

### Модель пользователей:

- Добавить фильтр списка по email и имени пользователя.

### Модель рецептов:



- В списке рецептов вывести название и автора рецепта.
- Добавить фильтр по автору, названию рецепта, тегам.
- На странице рецепта вывести число добавлений этого рецепта в избранное.

#### **Модель ингредиентов:**

- В список вывести название ингредиента и единицы измерения.
- Добавить фильтр по названию.

### **Технические требования и инфраструктура**

- Проект должен использовать базу данных PostgreSQL.
- Код должен находиться в репозитории `foodgram-project`.
- В Django-проекте должен быть файл `requirements.txt` со всеми зависимостями.
- Проект нужно запустить в трёх контейнерах (nginx, PostgreSQL и Django) через `docker-compose` на вашем сервере в Яндекс.Облаке. Образ с проектом должен быть запущен на Docker Hub.