**Техническое описание дипломного** проекта

Ваш дипломный проект — сайт Foodgram, «Продуктовый помощник». Вы напишете онлайн-сервис и API для него. На этом сервисе 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**

После того, как вы прочтёте уроки этой темы — вам станет доступен репозиторий foodgram-project-react; в нём подготовлен фронтенд и структура приложения. Склонируйте этот репозиторий и делайте проект в нём.

В репозитории есть папки *frontend*, *backend*, *infra,* *data и docs*.

* В папке *frontend* находятся файлы, необходимые для сборки фронтенда приложения.
* В папке *infra* — заготовка инфраструктуры проекта: конфигурационный файл nginx и *docker-compose.yml*.
* В папке *backend* пусто, там вы будете с нуля разрабатывать бэкенд продуктового помощника.
* В папке *data* подготовлен список ингредиентов с единицами измерения. Список сохранён в форматах JSON и CSV: данные из списка будет необходимо загрузить в базу.
* В папке *docs —* файлы спецификации API.

В репозитории нет ни базы данных, ни бекенда, однако сразу после клонирования репозитория вы можете запустить проект и увидеть спецификацию API. По этой спецификации вам предстоит написать API для проекта Foodgram.

**Запуск проекта**

В папке infra выполните команду docker-compose up.

При выполнении этой команде сервис frontend, описанный в *docker-compose.yml* подготовит файлы, необходимые для работы фронтенд-приложения, а затем прекратит свою работу.

Проект запустится на адресе [http://localhost](http://localhost/), увидеть спецификацию API вы сможете по адресу [http://localhost/api/docs/](http://localhost/api/docs/redoc.html)

Как будет выглядеть ваше приложение, можно посмотреть на [Figma.com](https://www.figma.com/file/HHEJ68zF1bCa7Dx8ZsGxFh/%D0%9F%D1%80%D0%BE%D0%B4%D1%83%D0%BA%D1%82%D0%BE%D0%B2%D1%8B%D0%B9-%D0%BF%D0%BE%D0%BC%D0%BE%D1%89%D0%BD%D0%B8%D0%BA-Final?node-id=0%3A)

**Базовые модели проекта**

Более подробно с базовыми моделями можно ознакомиться в спецификации API.

**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).
* Время приготовления в минутах.

Все поля обязательны для заполнения.

**Тег**

Тег должен описываться такими полями:

* Название.
* Цветовой HEX-код (например, #49B64E).
* Slug.

Все поля обязательны для заполнения и уникальны.

**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 [дизайн-макеты](https://www.figma.com/file/HHEJ68zF1bCa7Dx8ZsGxFh/%D0%9F%D1%80%D0%BE%D0%B4%D1%83%D0%BA%D1%82%D0%BE%D0%B2%D1%8B%D0%B9-%D0%BF%D0%BE%D0%BC%D0%BE%D1%89%D0%BD%D0%B8%D0%BA-Final?node-id=0%3A) проекта, это поможет лучше ориентироваться в описании.

**Главная страница**

Содержимое главной страницы — список первых шести рецептов, отсортированных по дате публикации (от новых к старым). Остальные рецепты доступны на следующих страницах: внизу страницы есть пагинация.

**Страница рецепта**

На странице — полное описание рецепта. Для авторизованных пользователей — 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.

**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
* Имя
* Фамилия

**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).
* 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 в него рецепты или удалять их, просматривать свою страницу избранных рецептов.
* Работать с персональным списком покупок: добавлять/удалять **любые** рецепты, выгружать файл со количеством необходимых ингридиентов 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.
* добавлять/удалять/редактировать теги.

Все эти функции нужно реализовать в стандартной админ-панели Django.

**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 фильтры по автору, названию рецепта, тегам.
* На странице рецепта вывести общее 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-react.
* В Django-проекте должен быть файл *requirements.txt* со всеми зависимостями.
* Проект нужно запустить в трёх контейнерах (nginx, PostgreSQL и Django) (контейнер frontend используется лишь для подготовки файлов) через docker-compose на вашем сервере в Яндекс.Облаке. Образ с проектом должен быть запушен на Docker Hub.