**Документация к проекту «Создание маршрутов на интегрированной карте»**

Назначение приложения: веб-приложение для создания, просмотра и редактирования маршрутов на карте, с управлением точками и изображениями к этим точкам.

Технологии:

- Backend: Django + Django REST Framework (API, сериализаторы, валидация, админка).

- Frontend: React, интеграция с Yandex Maps, управление маршрутами/точками, загрузка изображений.

Ключевые возможности:

- CRUD для маршрутов и точек, привязка изображений к ним.

- Валидация изображений (размер, формат, MIME).

- Управление интерфейсом по режимам (просмотр, создание, редактирование).

- Интерактивная работа с картой, per-point операции, перетаскивание точек в списке.

Архитектура:

Общая схема:

Django REST API предоставляет эндпоинты для маршрутов, точек и загрузок изображений.

React-приложение рендерит карту (Yandex Maps), формы и списки, общается с API через Axios.

Валидации и ограничения выполняются на бэкенде и частично в фронтенде (константы/сообщения).

Безопасность и конфигурация:

Переменные окружения (секреты, DEBUG, ALLOWED\_HOSTS, CORS/CSRF).

Статика/медиа: раздельные директории, учет при деплойменте.

CSRF-токен берётся с cookie, CORS настроен на доверенные источники.

Директории и роли:

Django: settings, models (Route, Point, PointImage), serializers, views, urls, admin.

React: компоненты (YandexMap, RoutesList, RouteCreationForm, PointsSection, PointCreationForm, RouteItem, SVG-иконки), контекст/редьюсер/хуки, утилиты (в т.ч. обработка изображений), константы (LIMITS, LIMIT\_MESSAGES).

Backend (Django):

Модели:

- Route: сущность маршрута

- Point: относится к Route (один ко многим)

- PointImage: относится к Point (один ко многим), хранит изображения для точки

Route - основной маршрут

├── name (CharField, max 200 символов)

├── description (TextField, max 1000 символов)

├── created\_at (DateTimeField, автоматически)

└── updated\_at (DateTimeField, автоматически)

Point - точка маршрута

├── route (ForeignKey → Route)

├── name (CharField, max 200 символов)

├── description (TextField, max 1000 символов)

├── lat (DecimalField, -90 до 90)

├── lon (DecimalField, -180 до 180)

└── order (IntegerField, порядок в маршруте)

PointImage - изображение точки (max 4 на точку)

├── point (ForeignKey → Point)

└── image (ImageField, max 1 МБ)

Сериализаторы:

- Валидируют входящие данные, включая изображения.

- Включают проверку на размер и тип файлов.

Эндпоинт:

- Маршруты: операции получения списка, создание/удаление/редактирование

- Точки операции для точек привязанных к маршрутам

- Изображения загрузка и привязка изображений к точкам/

**Работа приложения**

Создание маршрута:

- Пользователь кликает "Добавить маршрут".

- Переход в режим CREATE\_ROUTE.

- Форма маршрута готова к редактированию.

- Добавление точек через карту или вручную.

Сохранение маршрута:

- POST запрос к /api/routes/.

- Backend создает Route и связанные Point объекты.

- Frontend обновляет список маршрутов.

- Загрузка изображений:

- В форме точки выбираются файлы.

- Валидация на фронтенде (размер, формат).

- Сжатие больших изображений (> 0.9 МБ).

- Конвертация в Base64 строку.

- Сохранение точки (POST/PUT к /api/points/).

- Загрузка изображений (POST к /api/points/{id}/upload\_image/).

- Backend валидирует каждый файл.

- Сохранение PointImage объектов в БД.

- Редактирование маршрута:

- Клик на маршрут в списке → режим EDIT\_ROUTE.

- Редактирование названия/описания.

- Редактирование точек (добавление, удаление, перемещение).

Сохранение:

- PUT запрос к /api/routes/{id}/.

- Backend обновляет Route и Point объекты.

- Удаляет удаленные точки и изображени.

**Компоненты**

App.js - главный компонент приложения

Структура:

- RouteProvider (контекст)

- AppContent (содержимое приложения)

- routes-container (боковая панель)

- map-container (карта)

- YandexMap.js - карта

Режимы отображения:

- MAIN\_LIST - превью маршрута при наведении (красные точки)

- VIEW\_ROUTE\_DETAILS - просмотр маршрута (синие точки)

- CREATE\_ROUTE/EDIT\_ROUTE - редактирование (зеленые точки)

- CREATE\_POINT - временная красная точка

Особенности:

- Кастомные хинты с информацией о точке

- Автоматическое масштабирование по маршруту

- Поддержка изображений в подсказках

RouteCreationForm.js - форма маршрута:

- Название и описание маршрута

- Добавление точек через карту или вручную

- Список точек с drag & drop

- Сохранение/отмена

PointCreationForm.js - форма точки:

- Название и описание точки

- Координаты (широта/долгота)

- Загрузка изображений (до 4)

- Сжатие больших изображений

RoutesList.js / RouteItem.js - список маршрутов:

- Отображение всех маршрутов

- Быстрые действия (редактирование, удаление)

- Раскрываемый список точек

- Иконка фотографии если есть изображения