



Приложение № 1

к Договору № 041225 от «04» декабря 2025 г.

ТЕХНИЧЕСКОЕ ЗАДАНИЕ

на разработку веб-приложения «Мир Странствий» (MVP)

1. Общие сведения

1.1. Цель проекта

Создать минимально жизнеспособный продукт (MVP) веб-приложения «Мир Странствий» — социальной платформы для путешественников, которая позволяет:

- Просматривать карту с интересными местами.
- Создавать и сохранять маршруты путешествий.
- Оставлять отзывы и фотографии о местах.
- Подписываться на других пользователей и видеть их активность.
- Обмениваться сообщениями внутри платформы.

1.2. Технологический стек (MVP)

- Дизайн:** Figma (согласование внешнего вида приложения).
- Backend:** Django, Django REST Framework (DRF), Django ORM, Django Templates (для административной части).
- База данных:** PostgreSQL.
- Карта и маршрутизация:** JavaScript API Яндекс Карт (отображение карты, геокодирование, построение маршрутов по дорожной сети).
- Хранение медиа:** S3-совместимое объектное хранилище (например, Yandex Object Storage / VK Cloud).
- Фронтенд:** Next.js (React) для клиентской части, взаимодействующей с API (DRF), и Django Templates для административной панели.

- **Деплой:** облачная инфраструктура (Yandex/VK/другая по согласованию), серверное приложение под управлением gunicorn (или аналогичного WSGI/ASGI-сервера) за nginx.

Использование JavaScript API Яндекс Карт осуществляется в пределах условий бесплатного использования сервиса. В случае необходимости перехода на платный тариф (при превышении лимитов или изменении политики использования) Заказчик обеспечивает оформление и оплату соответствующей лицензии.

2. Функциональные требования

2.1. Модуль пользователей и аутентификации (User)

Функции:

1. Регистрация пользователя по e-mail и одноразовому паролю (OTP) / ссылке:

- ввод e-mail;
- отправка одноразового кода или ссылки на e-mail;
- завершение регистрации после подтверждения.

2. Авторизация и выход из системы:

- вход по e-mail + одноразовому коду/ссылке либо по e-mail/паролю (конкретный вариант фиксируется при реализации и описывается в документации);
- выход (инвалидирование сессии / токена).

3. Восстановление доступа:

- запрос восстановления по e-mail;
- отправка одноразовой ссылки/кода на e-mail.

4. Профиль пользователя:

- аватар (изображение);
- имя/ник;
- краткая информация о себе (био);
- страна/город (строковые поля);
- интересы (набор тегов / перечисление).

5. Настройки видимости профиля:

- «Публичный» — профиль и маршруты доступны всем;
- «Только для зарегистрированных пользователей».

Результат:

- Рабочий модуль регистрации/логина.
- Страницы/экраны просмотра профиля (своего и других пользователей).
- Страница/экран редактирования профиля.

2.2. Модуль карты и мест (Place)

Функции:

1. Отображение интерактивной карты на основе JavaScript API Яндекс Карт на отдельной странице/экране.
2. Вывод маркеров мест из БД на карте (placemark'и Яндекс Карт с данными из модели Place).

3. Модель Place:

- название;
- краткое описание;
- тип места (например: пляж, город, достопримечательность, треккинг и др.; выбор из предопределённого списка);
- координаты (широта/долгота — lat, lng);
- одно «главное» фото;
- дополнительно — список дополнительных фото/видео (в рамках MVP допускается хранение нескольких медиа-файлов).

4. Список мест рядом:

- по текущему виду карты (по ограничивающему прямоугольнику карты — bounding box);
- и/или по простому фильтру: страна/регион (текстовые поля в модели).

5. Страница/экран отдельного места:

- информация о месте (название, тип, описание, координаты);
- фото (основное и дополнительные);
- отзывы о месте;
- кнопка «Добавить в маршрут».

6. Поиск места:

- поиск по названию места/города по данным БД;
- при необходимости — использование геокодера Яндекс Карт для преобразования текстового запроса в координаты (без сохранения полученных от геокодера картографических данных, помимо координат, использованных для привязки пользовательского контента).

2.3. Модуль маршрутов (**Trip**)

Функции:

1. Модель **Trip**:

- владелец маршрута (пользователь);
- название маршрута;
- короткое описание (для отображения в ленте);
- полное описание (для детальной страницы маршрута);
- дата создания;
- флаг «публичный/приватный».

2. Модель **TripPoint**:

- привязка к маршруту;
- порядок следования (номер точки);
- координаты (lat/lng);
- опциональная текстовая заметка.

3. Создание маршрута:

- форма создания маршрута (название + описания);
- выбор точек маршрута через:
 - клики по карте Яндекс (установка/перетаскивание меток);
 - выбор существующих мест Place (подстановка их координат).

4. Визуализация маршрута:

- отображение точек маршрута на карте Яндекс;
- построение маршрута по дорожной сети (автомобильный или пешеходный — тип маршрута уточняется при согласовании) с использованием сервиса маршрутизации JavaScript API Яндекс Карт;

- автоматическое приближение карты к маршруту (установка границ по рассчитанной геометрии маршрута).

5. При этом:

- на стороне сервера хранятся только координаты точек маршрута (**TripPoint**);
- линия маршрута (геометрия) каждый раз рассчитывается на клиенте с помощью JavaScript API Яндекс Карт и не сохраняется в БД как собственные картографические данные.

6. Просмотр маршрута по публичной ссылке:

- для незарегистрированных пользователей доступен только просмотр маршрута и его описания (без возможности редактирования, копирования и комментирования).

7. Копирование маршрута:

- функция «Сохранить себе» — создаёт копию чужого публичного маршрута для текущего пользователя (с указанием исходного автора в метаданных).

8. Уведомление о пересечении маршрутов:

- в рамках MVP реализуется в упрощённом виде:
 - при добавлении или изменении маршрута проводится проверка на совпадение мест (**Place**) в маршрутах других пользователей;
 - при обнаружении совпадений может формироваться уведомление пользователю о том, что другие пользователи также имеют маршруты с теми же местами;
- геометрический анализ пересечения линий маршрута (по геометрии треков) в рамках MVP не реализуется.

Важно:

- В рамках MVP используется функциональность маршрутизации Яндекс Карт для построения маршрута по дорожной сети на клиентской стороне.
- На стороне сервера хранятся только пользовательские координаты и описание маршрута; данные о маршруте, возвращаемые Яндекс Картами (включая геометрию и служебные параметры), не сохраняются и не используются за пределами отображения маршрута в интерфейсе.
- Продвинутая логика (альтернативные маршруты, массовая оптимизация, сложные сценарии логистики и т.п.) не реализуется и может быть предметом отдельного технического задания.

2.4. Модуль отзывов и медиа (Review / Media)

Функции:

1. Модель **Review:**

- автор (пользователь);
- место (**Place**);
- оценка (1–5);
- текст отзыва;
- дата создания.

2. Возможность прикрепить к отзыву 1–3 фотографии (загрузка в S3-совместимое хранилище).

3. Отображение списка отзывов в карточке места:

- сортировка по дате (от новых к старым);
- отображение средней оценки по месту (вычисляемый показатель).

4. Ограничения и модерация:

- отзыв может редактировать или удалять только его автор (через интерфейс пользователя);
- администратор/модератор может скрыть или удалить любой отзыв через административную панель.

2.5. Модуль обмена сообщениями (Мессенджер)

Мессенджер реализуется как встроенный модуль на стороне сервера (Django), с хранением сообщений в БД и обменом через HTTP-запросы (без использования WebSocket в рамках MVP).

Функции:

1. Модель **Message:**

- отправитель (пользователь);
- получатель (пользователь) и/или идентификатор чата/комнаты;
- текст сообщения;
- дата и время отправки;
- дата и время прочтения (опционально);
- флаги «отправлено» / «получено» / «прочитано» (логическое или статусное поле).

2. Личные сообщения (1:1):

- список диалогов пользователя (по собеседникам);
- список сообщений в выбранном диалоге;
- отправка новых сообщений;
- пометка сообщений как прочитанных при открытии диалога.

3. Групповой «общий» чат (одна общая комната):

- отдельный канал/комната, в котором могут писать все авторизованные пользователи;
- отображение сообщений в общем канале;
- отправка сообщений в общий чат;
- обновление списка сообщений через перезагрузку страницы или периодические запросы (polling / AJAX).

4. В рамках MVP:

- не реализуются сложные статусы «онлайн/оффлайн», индикаторы набора текста, push-уведомления и иные функции, требующие real-time взаимодействия;
- логика мессенджера ограничивается отправкой и получением сообщений, их сохранением и отображением.

2.6. Социальные функции (подписки и лента активности)

Функции:

1. Подписки:

- пользователь может подписаться на другого пользователя;
- возможность отписаться;
- хранение связей подписчик → автор.

2. Лента активности:

- формирование списка событий от пользователей, на которых подписан текущий пользователь, в том числе:
 - создание новых маршрутов;
 - публикация новых отзывов;
- сортировка по дате добавления (новые события отображаются выше).

3. Страница/экран «Лента»:

- отображение компактного списка событий, например:
 - «Иван создал маршрут “Горы Синая”»;
 - «Анна оставила отзыв о месте “Naama Bay”»;
 - «Сергей добавил фото о месте “Аэропорт Хабаровск”
- переход к соответствующему маршруту/месту/профилю по клику.

4. Ограничения MVP:

- отсутствует сложная рекомендательная лента по интересам/геолокации;
- лента строится только на основе подписок и случайно подобранных контента (создание маршрута, отзыв).

2.7. Чат-бот ИИ (тур-гид)

Функции:

1. Интерфейс чат-бота:

- отдельная всплывающая форма чата с ИИ-ботом (тур-гидом);
- ввод текстового вопроса;
- отображение ответов бота в формате диалога.

2. Возможности бота (MVP):

- ответы на общие вопросы о местах (туристическая информация, типы мест, общие рекомендации);
- подсказки по возможным маршрутам (на уровне рекомендаций: какие типы мест/направления можно комбинировать, без автоматического построения маршрута в базе);
- ответы на общие «лайфхаки» по путешествиям (сбор багажа, базовая безопасность, общие советы).

3. Техническая реализация:

- использование внешнего сервиса LLM (например, OpenAI или аналог), взаимодействие через API;
- ключи доступа и оплата запросов к LLM-сервису обеспечиваются Заказчиком (либо отдельным соглашением Сторон).

4. Ограничения и ответственность:

- ответы чат-бота носят справочно-рекомендательный характер;
- бот не даёт юридических, медицинских, финансовых и иных специализированных консультаций;
- использование информации осуществляется пользователем на свой риск.

2.8. Административная панель и модерация

Функции:

1. Использование стандартной административной панели Django.
2. Доступные действия в админке:
 - управление пользователями (просмотр, блокировка/деактивация при необходимости);
 - управление местами (Place) — добавление/редактирование/удаление;
 - управление отзывами (Review) — удаление/скрытие при нарушениях;
 - управление маршрутами (Trip) — скрытие/удаление при нарушениях;
 - просмотр и обработка жалоб пользователей.
3. Жалобы пользователей:
 - форма «Пожаловаться» в пользовательском интерфейсе на:
 - место;
 - отзыв;
 - маршрут;
 - жалоба фиксируется в базе данных с указанием объекта жалобы, автора жалобы и причины (текст);
 - в админке — список жалоб с возможностью:
 - просмотреть объект жалобы;
 - снять жалобу (если нарушений нет);
 - скрыть/удалить содержимое при наличии нарушений.

3. Нефункциональные требования

3.1. Производительность

1. В рамках MVP обеспечивается корректная работа Продукта при одновременной нагрузке до **100 (ста) одновременно активных пользователей**.
2. Нагрузочное и стресс-тестирование, а также масштабирование под боевые нагрузки выше указанного значения могут быть предметом отдельного этапа работ и/или отдельного технического задания.

3.2. Безопасность

1. Использование стандартных механизмов безопасности Django и применяемых библиотек на фронтенде, в том числе:
 - защита от CSRF там, где это применимо;
 - защита от XSS (корректное экранирование пользовательского ввода и выводимых данных).
2. Хранение паролей пользователей:
 - пароли (если используются) хранятся исключительно в виде криптостойких хэшей с использованием стандартных механизмов Django (или аналогичных по стойкости).
3. Хранение других пользовательских данных:
 - по общему правилу данные хранятся в базе данных в открытом виде, за исключением полей, которые Стороны отдельно согласуют как требующие шифрования (например, при наличии специфических требований законодательства или внутренней политики безопасности Заказчика);
 - при необходимости на уровне инфраструктуры используется шифрование дисков и/или бэкапов, предоставляемое облачным провайдером.
4. Доступ к административной панели:
 - ограничен только для авторизованных администраторов/модераторов;
 - адрес и способы доступа (в т.ч. возможные IP-фильтры, VPN и т.п.) могут быть дополнительно согласованы Сторонами.

3.3. Кроссбраузерность и адаптивность

1. Поддержка актуальных стабильных версий следующих браузеров:

- Google Chrome;
- Mozilla Firefox;
- Safari;
- Microsoft Edge.

2. Адаптивный интерфейс:

- корректное отображение на настольных компьютерах и ноутбуках;
- корректное отображение на современных мобильных устройствах (смартфоны) в вертикальной ориентации.

3.4. Язык интерфейса

1. В рамках MVP интерфейс реализуется на русском языке.
2. Возможность дальнейшего добавления других языков учитывается при проектировании (i18n), но разработка и включение дополнительных языков не входит в объём работ по данному Техническому заданию и может быть оформлена отдельным этапом.

3.5. Использование внешних сервисов

1. Для работы карты и маршрутизации используется JavaScript API Яндекс Карт.
2. Для работы ИИ-бота используется внешний сервис LLM (например, OpenAI или аналогичный).
3. Ключи доступа, регистрация учётных записей и оплата услуг внешних сервисов (включая платные тарифы Яндекс Карт и LLM-сервиса) осуществляются Заказчиком или по отдельному письменному соглашению Сторон.

Настоящее Техническое задание является неотъемлемой частью Договора № 041225 от «04» декабря 2025 г.

ЗАКАЗЧИК:

ООО «Алгоритм»

М.П.

/Савин С. В./

ИСПОЛНИТЕЛЬ:

ООО «РЕМОНТ ПРЕМИУМ»

М.П.

/Клюкин А. Н./