Сервис обрабатывает запрос следующим образом:

1. Пользователь заходит на сайт, отвечает на вопросы. Указывает текущей адрес, описывает что он хочет от прогулки и сколько у него есть свободного времени. В свободной форме. Из опроса формируется запрос к ИИ.
2. С помощью ИИ из запроса пользователя извлекаются его адрес и время на прогулку, а также формируется упорядоченный список релевантных тегов. Например запрос “Я хочу сходить в музей и потом попить кофе, у меня примерно 3 часа, сейчас я около Ошарской 14.” будет преобразован в {'tags': [['Музей'], ['Кофейня'']], 'user_location': 'улица Ошарская, дом 14', 'time': 180}
3. Далее из базы данных объектов извлекаются все места имеющие соответствующие теги, их координаты передаются в блок нахождения оптимального маршрута. Также туда передаются дополнительные инструкции: требуемый порядок тегов в маршруте и метки для групп тегов (выбор одного из группы, включение нескольких объектов соответствующих группе в маршрут при возможности). Адрес пользователя нормализуется и выполняется нечеткий поиск по базе данных адресов и их географических координат.
4. Алгоритм нахождения маршрута отсекает нерелевантные из-за удаленности от пользователя точки. Для оставшихся строится матрица времен (время, которое нужно затратить чтобы пройти пешком от точки i к точке j). С помощью нее находится список маршрутов, который подходит по длительности (с учетом оценочного времени необходимого для посещения объектов). Из списка выбирается случайный элемент.
5. На основе базы данных объектов с помощью ИИ составляется описание маршрута

.

Аренда сервера для запуска на нем llm была признана экономически нецелесообразной, поэтому для доступа к ИИ используется API OpenRouter. Используемая модель Gemini 2.5 Flash. У нее высокая скорость ответа, хорошее понимание русскоязычных запросов (2 место на lmarena) и следование инструкциям. Для нормализации адресов используется libpostal. В качестве движка построения маршрутов используется osrm, он быстро работает, обладает достаточным функционалом и прост в использовании. Веб-интерфейс написан на VUE 3.

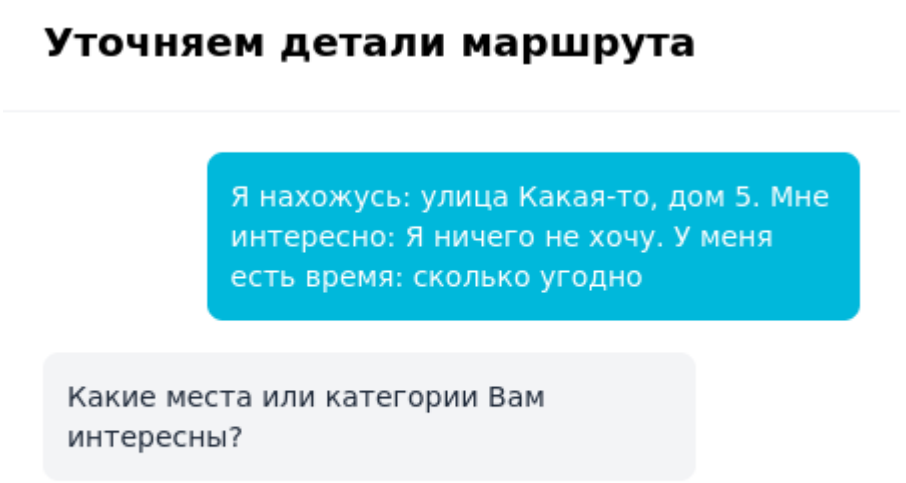
Приложение разбито на компоненты - docker контейнеры:

1. libpostal-service - готовый образ, предоставляющий REST API для библиотеки libpostal. Используется, так как библиотека очень тяжелая.
2. navigation-engine - компонент на базе osrm-backend. Предоставляет API OSRM, но с картой только по Нижнему Новгороду, что ускоряет работу.
3. engine-wrapper - обертка над navigation-engine, для упрощения взаимодействия. Написан на FastAPI
4. input_to_route - бизнес логика находиться здесь, написан на FastAPI. Получает данные от пользователя, ищет по точкам в базе, общается с ИИ.
5. web-ui - веб-интерфейс пользователя

Для хостинга используется платформа dockhost, которая позволяет гибко разворачивать docker контейнеры (российский аналог heroku).

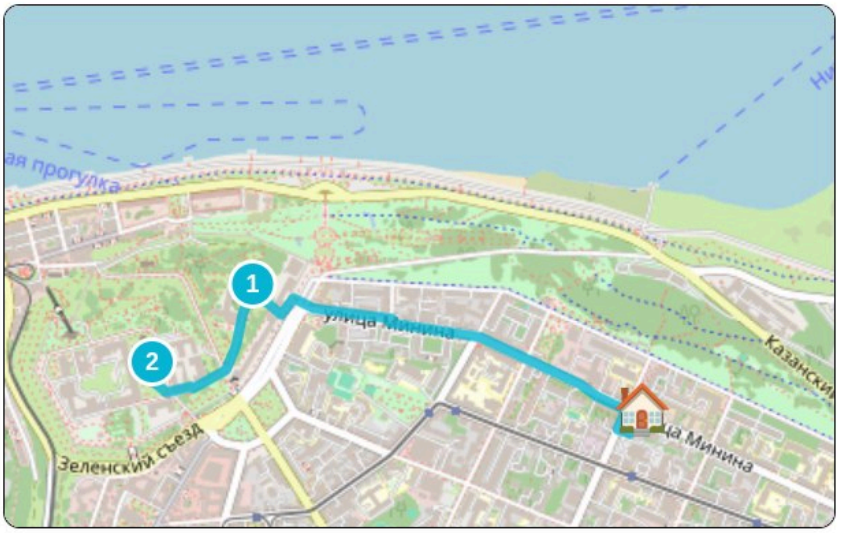
Для получения данных от пользователя используется опрос. На трех последовательных экранах пользователь отвечает на вопросы в свободной форме, что удобно для него. В любой момент можно вернуться на предыдущий экран для корректировки, нажав на стрелку назад вверху экрана.

Если пользователь ошибся при опросе, или указал не полные данные, система переходит к общению в диалоговом режиме. Диалог продолжается до получения всех данных от пользователя (на данный момент есть ограничение на 4 сообщения от пользователя, для избежания траты токенов)



Несмотря на преобразование запроса в набор тегов, наше решение строит маршрут максимально близко к исходной логике. Как пример рассмотрим несколько внешне похожих запросов с существенно разными пожеланиями пользователя.

1. ‘Хочу сходить в музей и попить кофе’

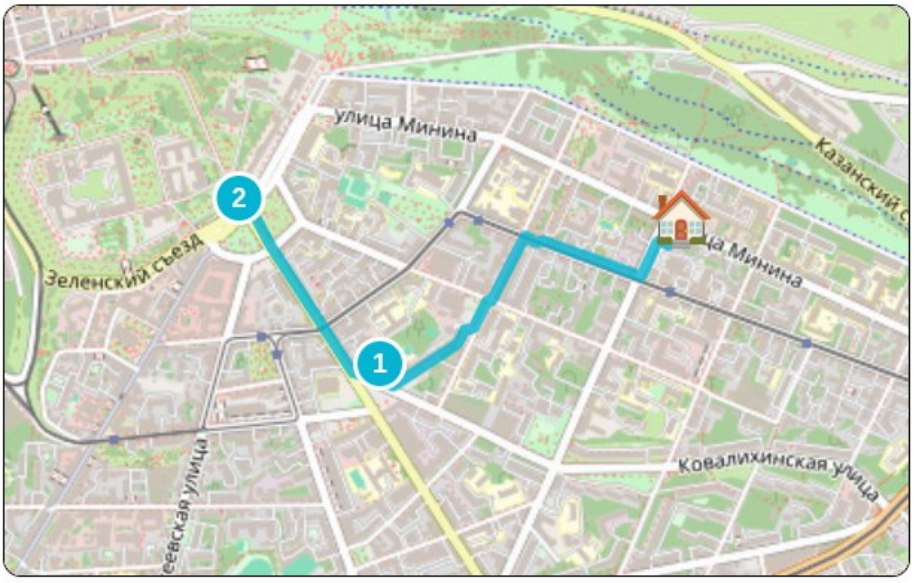


Отлично, вот маршрут, который подойдет для вашего времени и интересов:

Нижегородский государственный художественный музей Погрузитесь в мир искусства, где представлены коллекции русского, западноевропейского и восточного искусства, включая произведения национального и мирового значения.

Skuratov Coffee Насладитесь ароматным кофе и вкусными десертами, любуясь панорамным видом на кремль и Волгу.

2. ‘Хочу попить кофе, потом в музей’

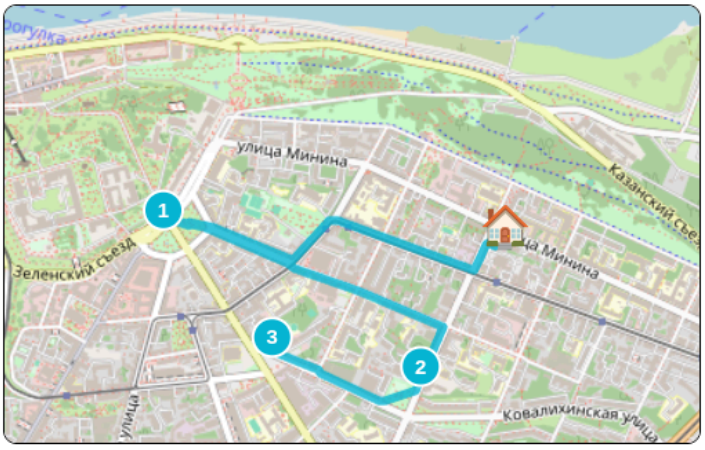


Вот ваш маршрут:

Skuratov Coffee: Насладитесь ароматным кофе и вкусными десертами в уютной кофейне.

Дмитриевская башня: Познакомьтесь с визитной карточкой Нижнего Новгорода, величественной башней, где когда-то располагался художественно-исторический музей.

3. ‘Хочу походить по музеям, потом зайти за кофе’



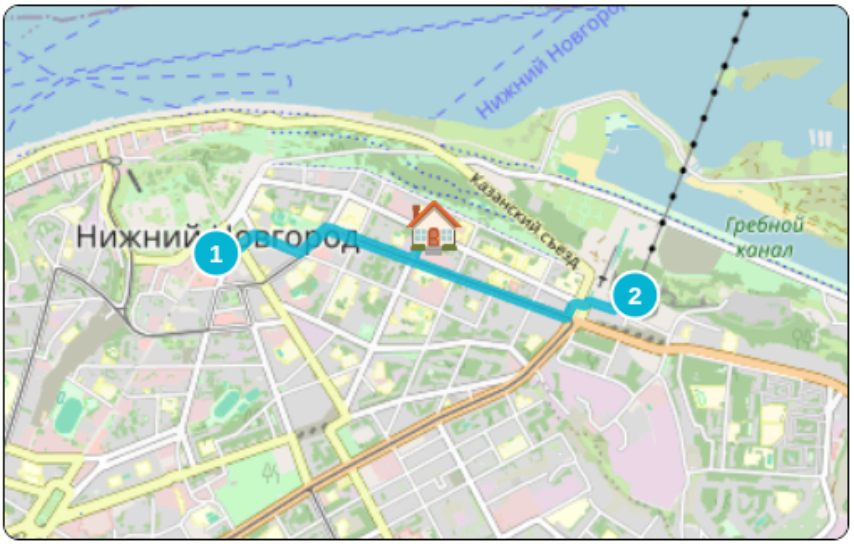
Вот ваш маршрут:

Дмитриевская башня Визитная карточка Нижнего Новгорода, эта башня занимает главенствующее место в Нижегородском кремле и получила современный облик в русском стиле в 1895 году. В ней был организован художественно-исторический музей, а на шпиль водружен позолоченный герб города.

Музей Горького Погрузитесь в атмосферу дома, где Максим Горький жил с 1902 по 1904 год, завершил пьесу «На дне» и принимал известных гостей, таких как Шаляпин.

Skuratov Coffee Завершите прогулку в уютной кофейне, где можно насладиться ароматным кофе и вкусными десертами.

4. ‘Хочу сходить в музей или попить кофе, потом прокатиться на канатной дороге’



Вот ваш маршрут:

Skuratov Coffee: Насладитесь ароматным кофе спешелти-класса с зерном собственной обжарки и вкусными десертами.

Канатная дорога: Прокатитесь по самой большой в Европе междугородной канатной дороге, откуда открываются захватывающие виды на Волгу.

Адрес и время можно вводить в довольно свободной форме. Название улицы в большинстве случаев удается распознать даже при наличии опечаток. При вводе времени вместо длительности прогулки можно написать ‘я свободен до 6 вечера.’