

“INCITY” ПЛАНИРОВЩИК МАРШРУТА ДЛЯ T-ПУТЕШЕСТВИЙ

Route planner for T-Travel

Серверная часть

Командный программный проект. Исполнители:

Кудрявцев Георгий Александрович БПИ-245

Осина Дарья Евгеньевна БПИ-245



Руководитель проекта:

Зорин Арсений Леонидович,
руководитель команды Trains, Т-Банк

2

ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

Планировщик маршрутов для Т-Путешествий InCity разрабатывается с целью помочь туристам в составлении маршрута их прогулок по незнакомому городу.

Целевая аудитория сервиса - туристы, путешествующие по городам России, люди, находящиеся в командировках, а также местные жители, которые хотели бы узнать что-то новое о своем городе.

Область применения InCity - маршрутизация и навигация

В современных реалиях путешествия по городам России становятся все более актуальной темой. Однако в процессе путешествий у туристов могут возникнуть определенные проблемы, например:

- сложно добраться до точки назначения
- разрозненность данных об интересных местах
- дефицит временного ресурса для поиска достопримечательностей

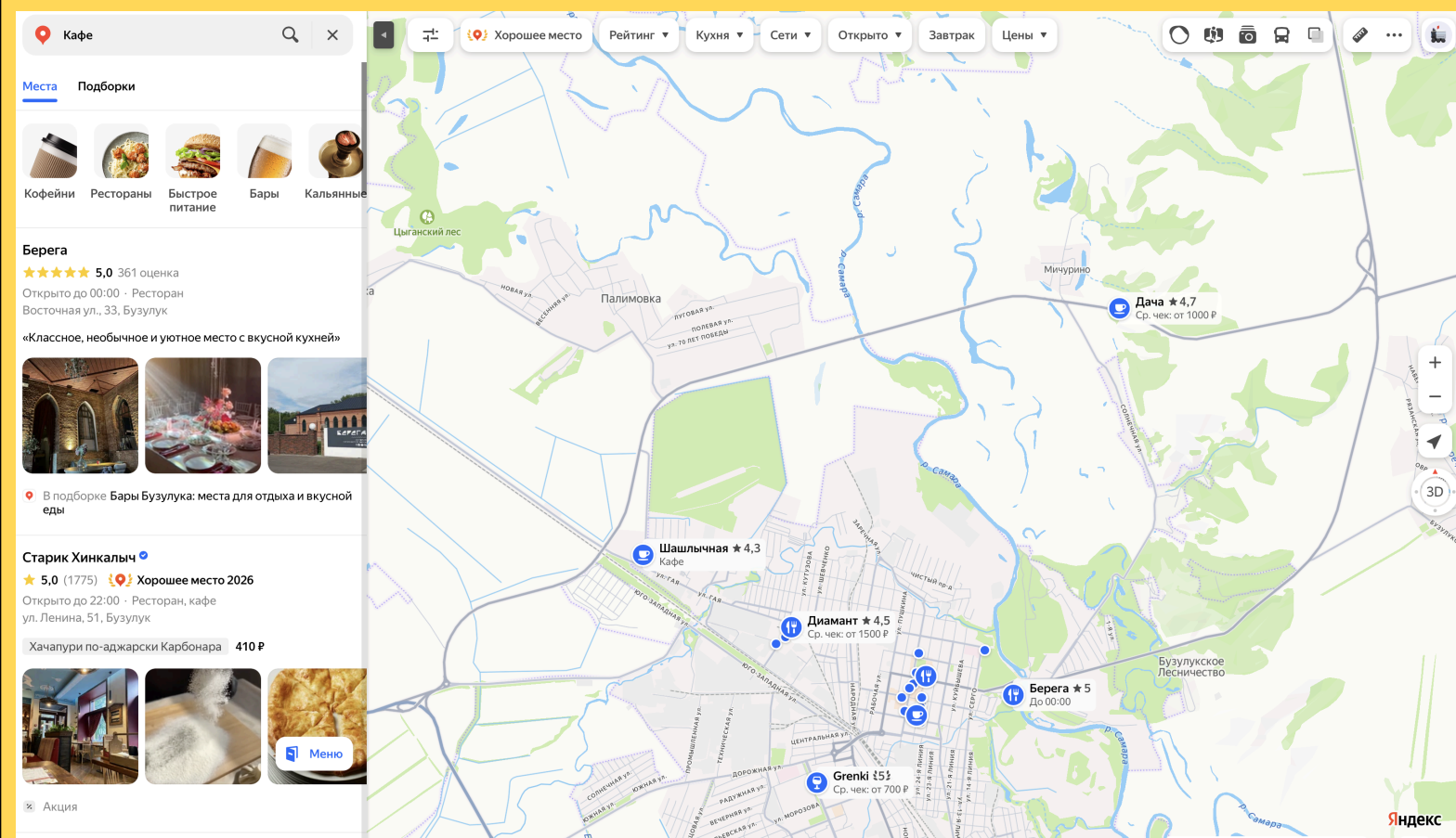
Веб-сервис “InCity” призван помочь пользователям легко и эффективно находить достопримечательности и события, а также предоставлять связный маршрут до них

Цель проекта - создать веб-приложение для поиска достопримечательностей и событий, интересных для посещения, а также для построения маршрута до них с использованием общественного транспорта

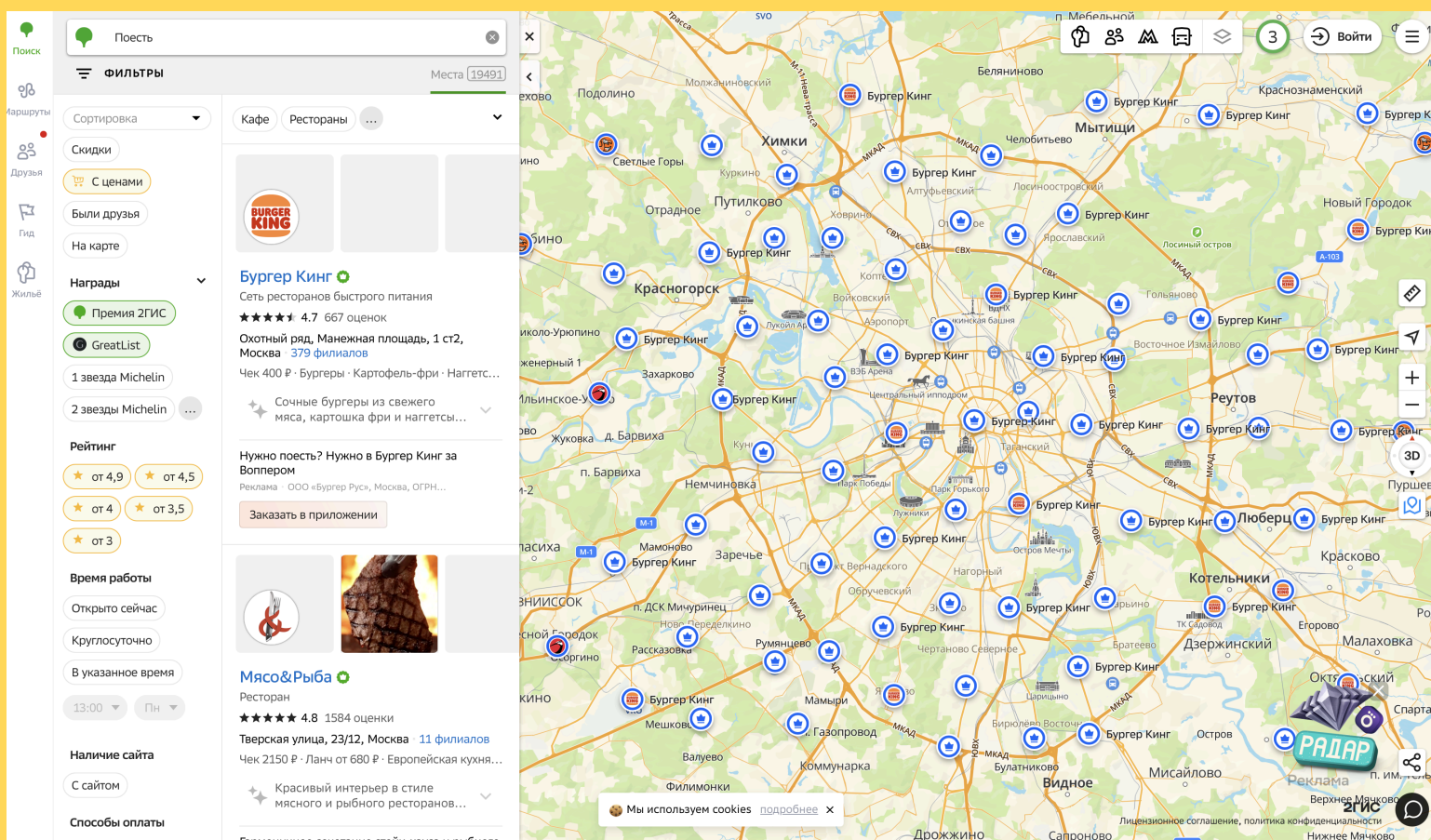
Задачи проекта:

- 1 ПРОАНАЛИЗИРОВАТЬ СУЩЕСТВУЮЩИЕ АНАЛОГИ
- 2 СПРОЕКТИРОВАТЬ АРХИТЕКТУРУ СИСТЕМЫ
- 3 РАЗРАБОТАТЬ СЕРВЕРНУЮ И КЛИЕНТСКУЮ ЧАСТИ ПРИЛОЖЕНИЯ
- 4 ИНТЕГРИРОВАТЬ КАРТОГРАФИЧЕСКИЕ АР
- 5 ИНТЕГРИРОВАТЬ АР ПОИСКА СОБЫТИЙ
- 6 РЕАЛИЗОВАТЬ АЛГОРИТМ ПОСТРОЕНИЯ МАРШРУТОВ
- 7 РАЗРАБОТАТЬ АДАПТИВНЫЙ ИНТЕРФЕЙС
- 8 ПРОВЕСТИ ТЕСТИРОВАНИЕ СИСТЕМЫ

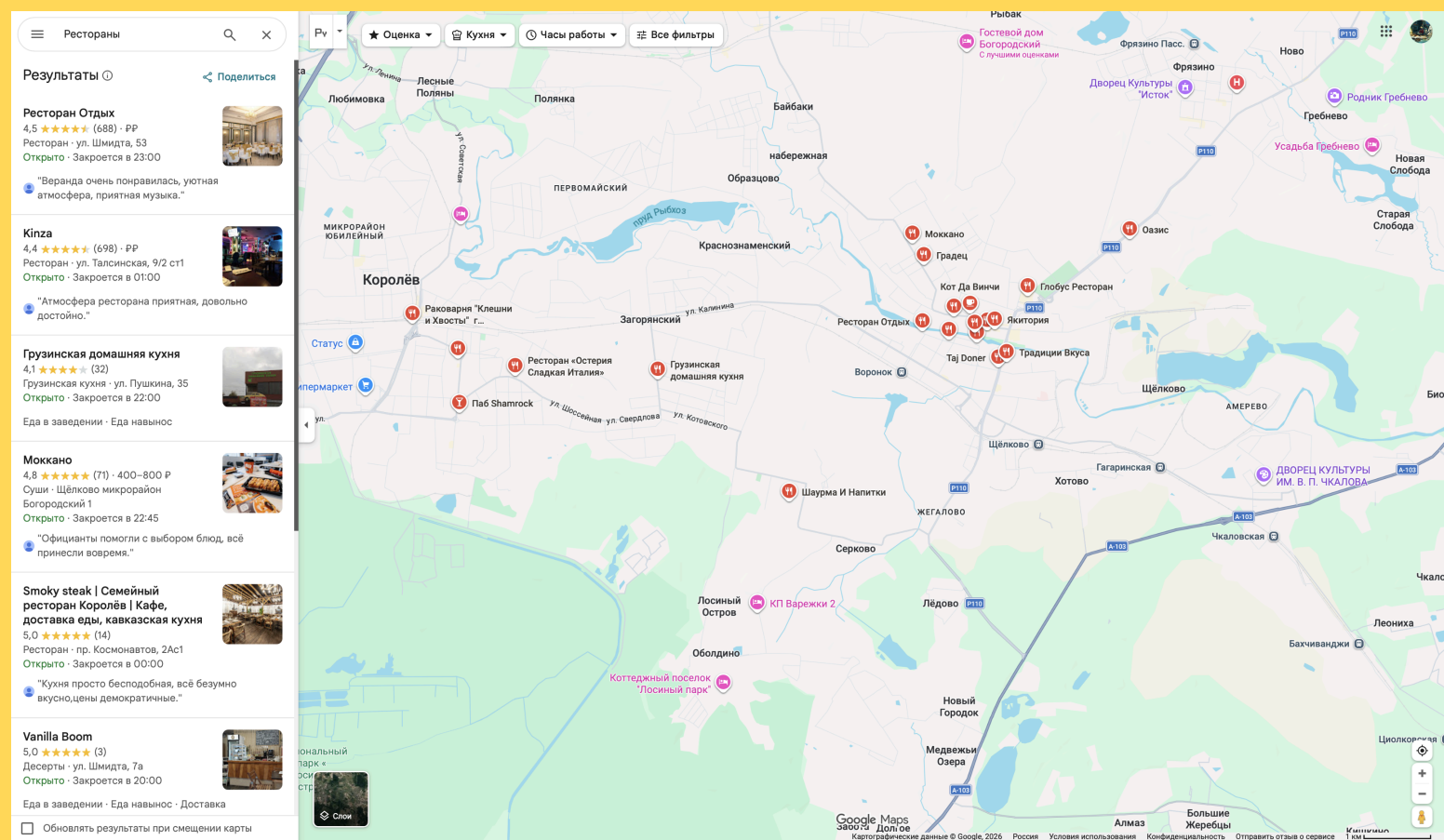
АНАЛИЗ КОНКУРЕНТОВ



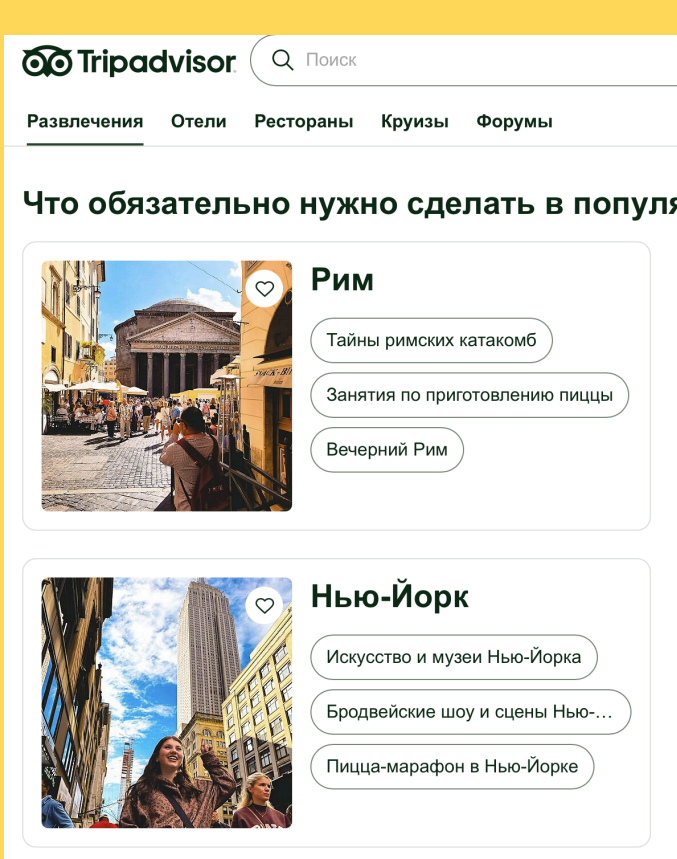
Яндекс Карты



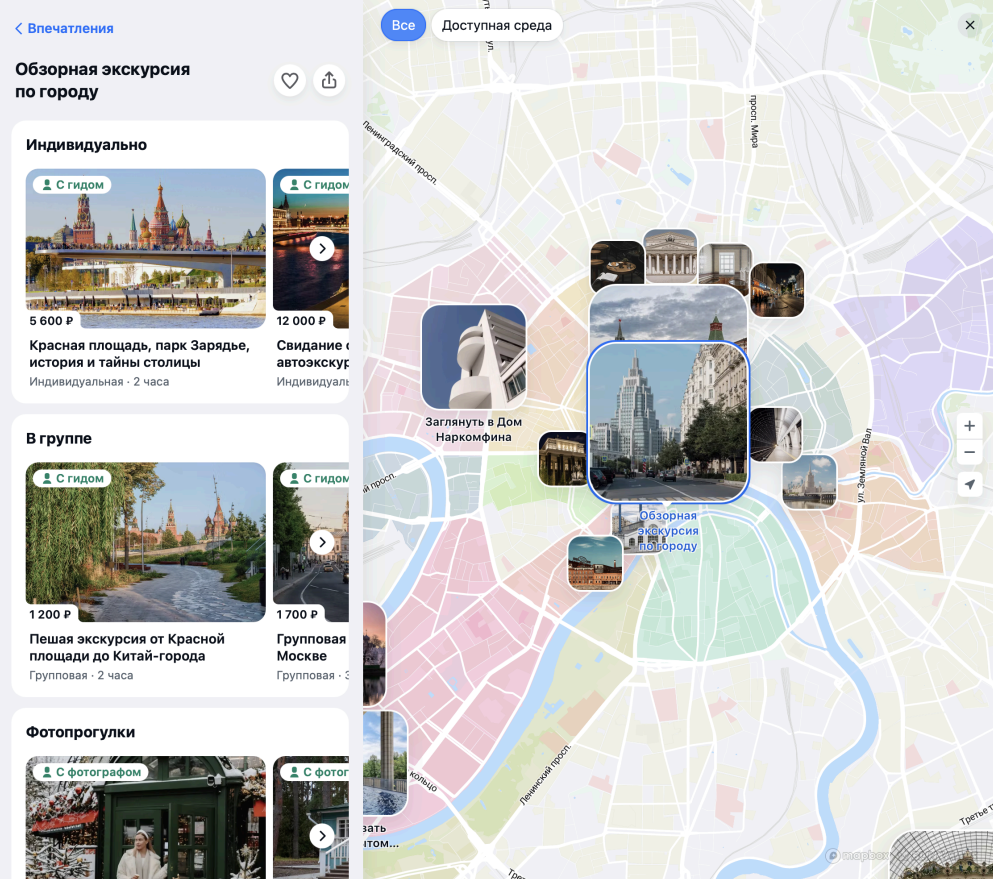
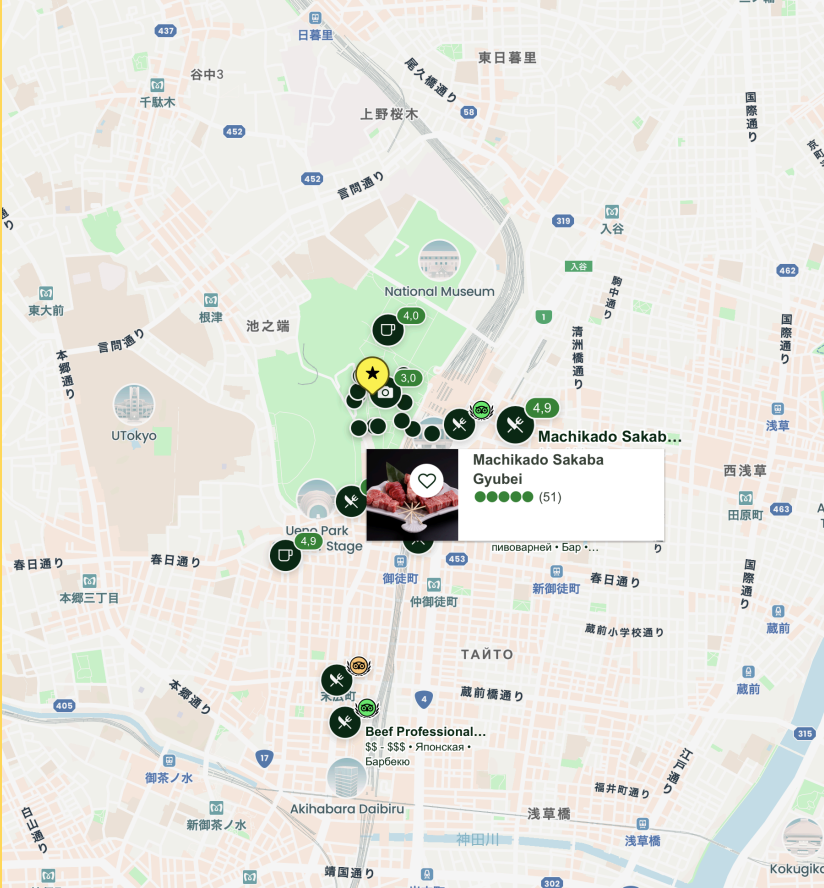
2ГИС



Google Maps



Tripadvisor



“Погнали” от Aviasales

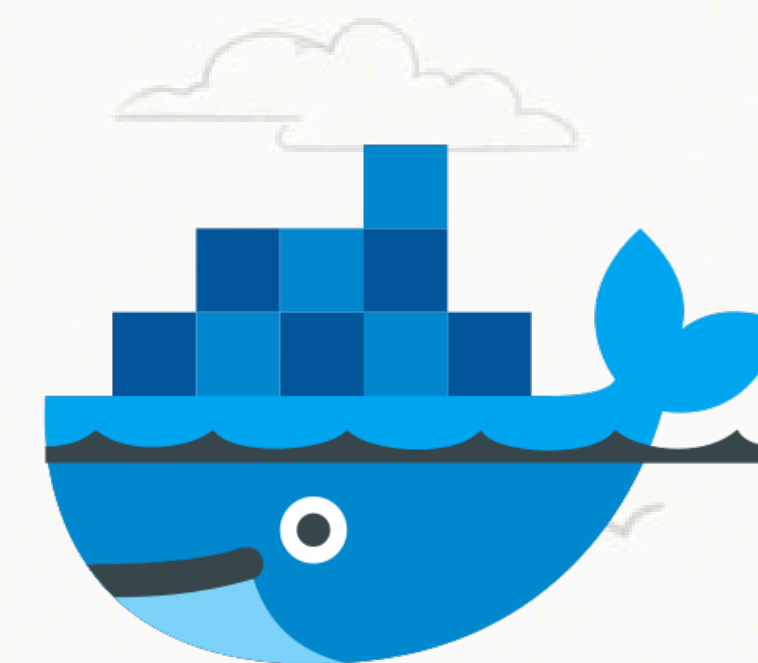
ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ СЕРВЕРНОЙ ЧАСТИ

- Принятие Rest-запроса от клиентской части
- Получение данных о транспортных узлах и маршрутах из базы данных
- Получение данных о происходящих в городе мероприятиях с помощью KudaGo API
- Формирование списка достопримечательностей и мероприятий, выбранных к посещению (список обхода)
- Построение маршрута между объектами, включенными в список обхода.
- Отправка ответов на Rest-запросы в клиентскую часть приложения

```
porting to image
exporting layers
exporting manifest sha256:303fa8032
exporting config sha256:6eb8ae3c143
exporting attestation manifest sha2
exporting manifest list sha256:7d27
naming to docker.io/library/backend
unpacking to docker.io/library/back
solving provenance for metadata file
nning 2/2
kend-backend Built
tainer route_planner_api Recreated
ing to route_planner_api, route_plan
planner_db |
planner_db | PostgreSQL Database di
planner_db |
planner_db | 2026-05-20 09:18:58.54
.0-19) 14.2.0, 64-bit
planner_db | 2026-05-20 09:18:58.54
planner_db | 2026-05-20 09:18:58.54
planner_db | 2026-05-20 09:18:58.54
planner_db | 2026-05-20 09:18:58.55
planner_db | 2026-05-20 09:18:58.57
planner_api | Waiting for PostgreSQ
planner_api | PostgreSQL is ready.
planner_api | Database already cont
planner_api | warn: Microsoft.AspNe
planner_api | Overriding HTTP
planner_api | info: Microsoft.Hosti
planner_api | Now listening o
planner_api | info: Microsoft.Hosti
planner_api | Application sta
planner_api | info: Microsoft.Hosti
planner_api | Hosting environ
planner_api | Content root pa
planner_api | info: Microsoft.AspNe
planner_api | Request startin
rt=metro,bus,tram,trolleybus&attract
s= - - -
planner_api | info: Microsoft.AspNe
planner_api | CORS policy exe
planner_api | info: Microsoft.AspNe
```

7

СТЕК ТЕХНОЛОГИЙ

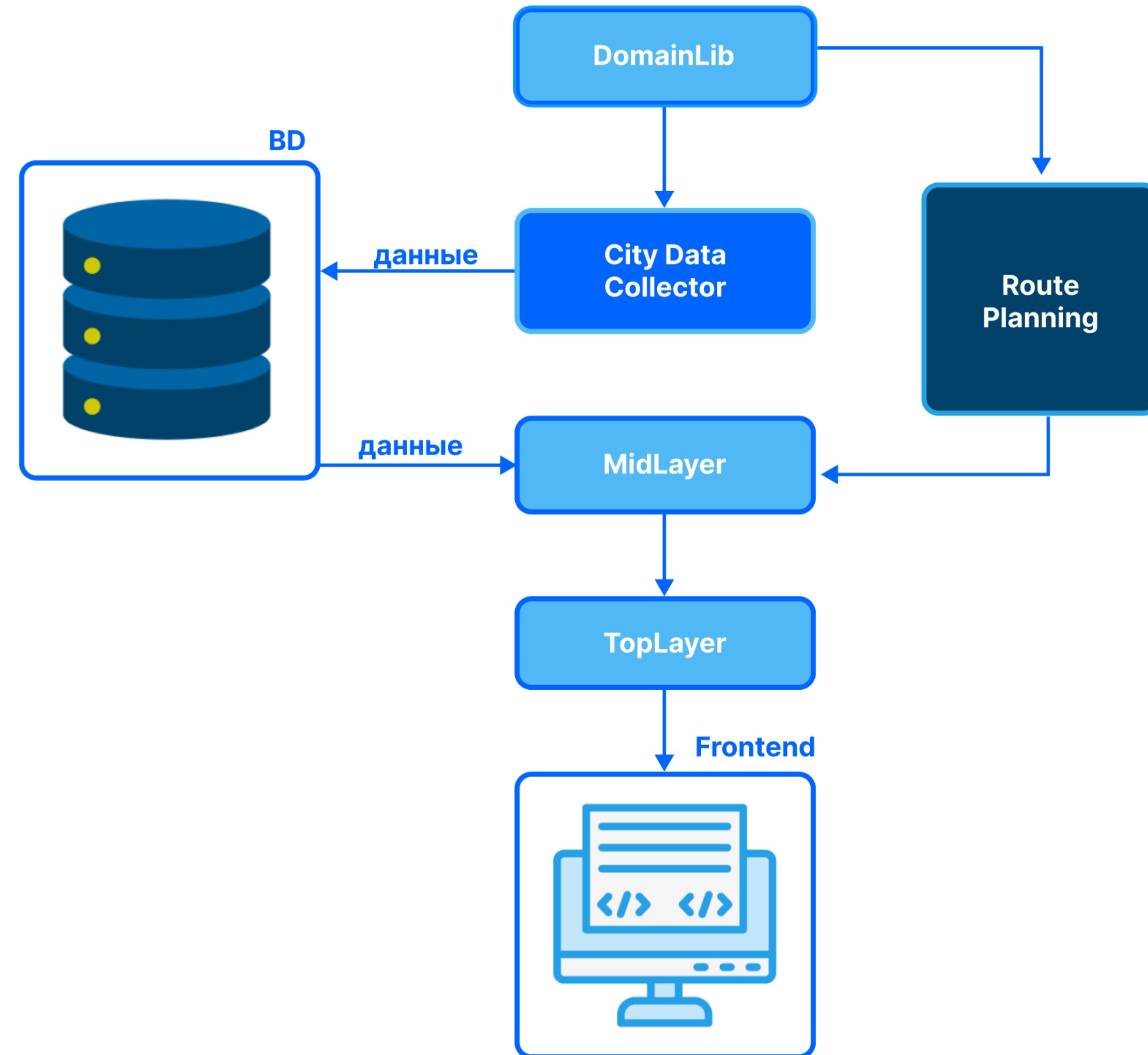


Dapper



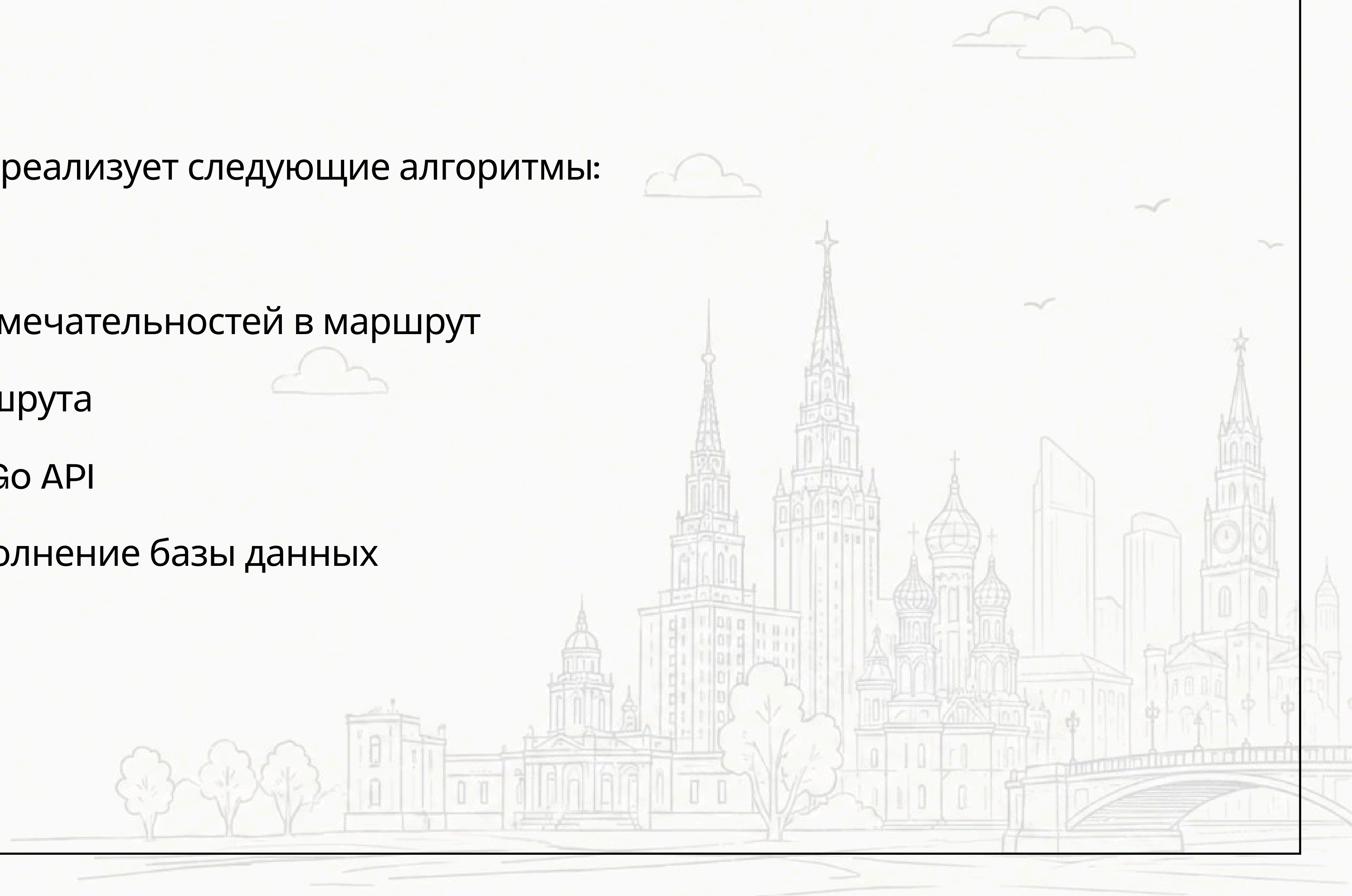
Open
Street
Map

BACKEND



Серверная часть приложения реализует следующие алгоритмы:

1. Алгоритм кластеризации
2. Алгоритм отбора достопримечательностей в маршрут
3. Алгоритм построения маршрута
4. Парсинг объектов из KudaGo API
5. Сбор данных из OSM и заполнение базы данных



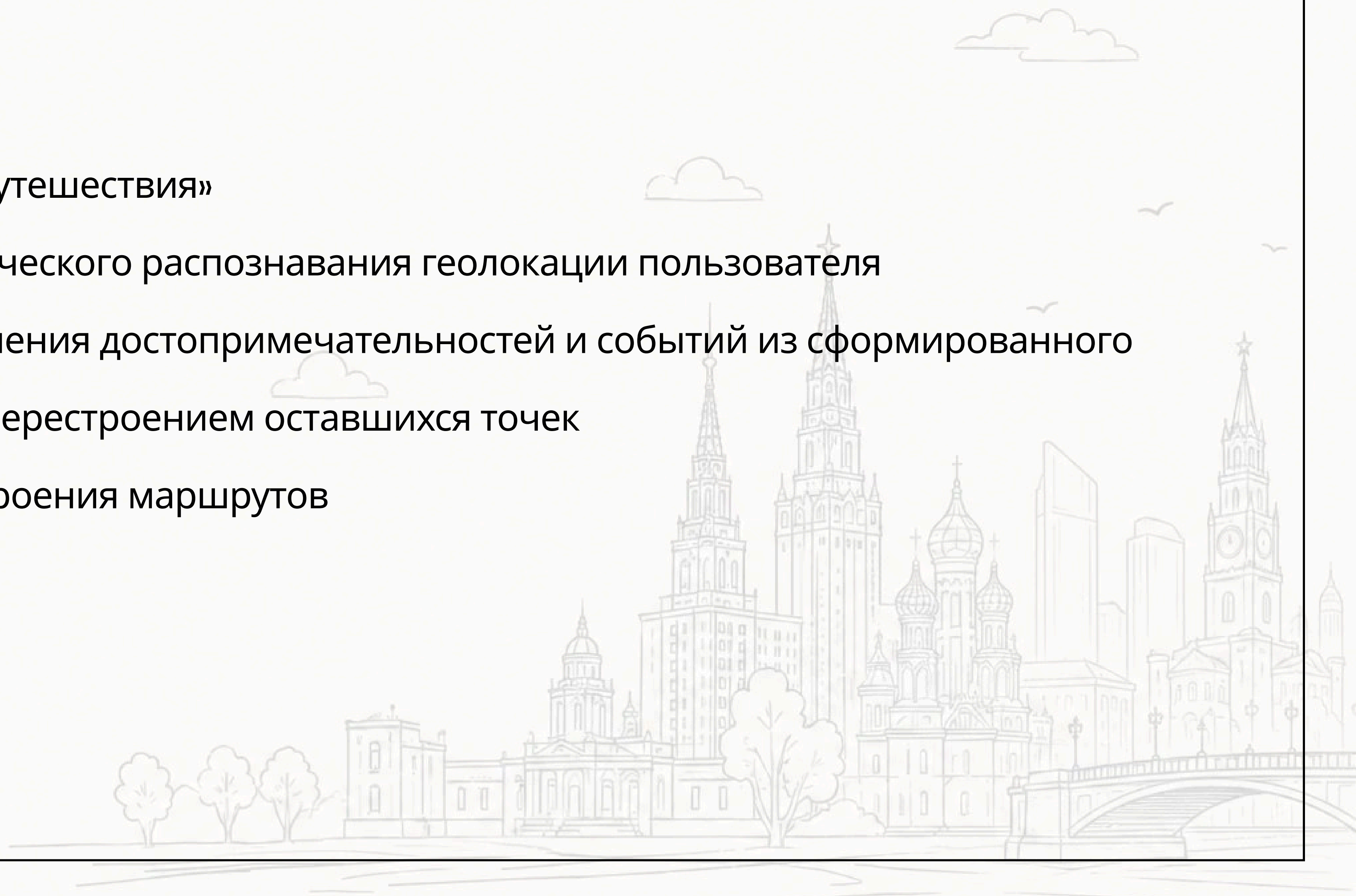
10

РЕЗУЛЬТАТЫ И ВЫВОДЫ

- 1 Реализация алгоритма автоматического построения кольцевых маршрутов
- 2 Интеграция с внешними источниками данных
- 3 Разработаны алгоритма парсинга данных из БД и сторонних API
- 4 Поддержка 50 городов
- 5 Реализация алгоритма сборщика данных

НАПРАВЛЕНИЯ ДАЛЬНЕЙШЕЙ РАЗРАБОТКИ

- Интеграция с API сервиса «Т-Путешествия»
- Внедрение функции автоматического распознавания геолокации пользователя
- Реализация возможности удаления достопримечательностей и событий из сформированного маршрута с автоматическим перестроением оставшихся точек
- Оптимизация алгоритма построения маршрутов
- Оффлайн-режим работы
- ML-рекомендации




```
=> => exporting attestation manifest sha256:d49087a9307b0125ac55e418e62d5deeb31edeb76fe22b6fa6d1bec128ec6a82 0.0s
=> => exporting manifest list sha256:7d270e461fbe9a7e7ee844bb59f45298a916d8c62b2a198e5d2deb4c94ccd488 0.0s
=> => naming to docker.io/library/backend-backend:latest 0.0s
=> => unpacking to docker.io/library/backend-backend:latest 1.0s
=> resolving provenance for metadata file 0.0s
[+] Running 2/2
  ✓ backend-backend Built 0.0s
  ✓ Container route_planner_api Recreated 0.5s
Attaching to route_planner_api, route_planner_db
route_planner_db | PostgreSQL Database directory appears to contain a database; Skipping initialization
route_planner_db | 2026-05-20 09:18:58.542 UTC [1] LOG: starting PostgreSQL 16.14 (Debian 16.14-1.pgdg13+1) on x86_64-pc
n 14.2.0-19) 14.2.0, 64-bit
route_planner_db | 2026-05-20 09:18:58.543 UTC [1] LOG: listening on IPv4 address "0.0.0.0", port 5432
route_planner_db | 2026-05-20 09:18:58.543 UTC [1] LOG: listening on IPv6 address ":::", port 5432
route_planner_db | 2026-05-20 09:18:58.548 UTC [1] LOG: listening on Unix socket "/var/run/postgresql/.s.PGSQL.5432"
route_planner_db | 2026-05-20 09:18:58.556 UTC [29] LOG: database system was shut down at 2026-05-20 09:17:10 UTC
route_planner_db | 2026-05-20 09:18:58.575 UTC [1] LOG: database system is ready to accept connections
route_planner_api | Waiting for PostgreSQL to be ready...
route_planner_api | PostgreSQL is ready. Checking if seed data needs to be loaded...
route_planner_api | Database already contains data (city count = 50). Skipping seed.
route_planner_api | warn: Microsoft.AspNetCore.Hosting.Diagnostics[15]
route_planner_api | Overriding HTTP_PORTS '8080' and HTTPS_PORTS ''. Binding to values defined by URLs instead 'htt
route_planner_api | info: Microsoft.Hosting.Lifetime[14]
route_planner_api | Now listening on: http://[::]:5000
route_planner_api | info: Microsoft.Hosting.Lifetime[0]
route_planner_api | Application started. Press Ctrl+C to shut down.
route_planner_api | info: Microsoft.Hosting.Lifetime[0]
route_planner_api | Hosting environment: Production
route_planner_api | info: Microsoft.Hosting.Lifetime[0]
route_planner_api | Content root path: /app
route_planner_api | info: Microsoft.AspNetCore.Hosting.Diagnostics[1]
route_planner_api | Request starting HTTP/1.1 GET http://localhost:5001/route-from-point?lat=55.69519643041895&lng=
ransport=metro,bus,tram,trolleybus&attractions=park-and-garden,monument,gastronomy&subattractions=urban,sculpture,memorial
route_planner_api | 2026-05-20 09:18:58.575 UTC [1] LOG: database system is ready to accept connections
```



- Figma | Создание презентаций // 2025. URL: <https://www.figma.com/> (дата обращения: 17.05.2026).
- QRCode.ru | Генератор QR-кодов онлайн // 2025. URL: <http://qrcoder.ru/> (дата обращения: 20.05.2026).
- Яндекс Диск | Облачное хранилище файлов // 2025. URL: <https://disk.yandex.ru/> (дата обращения: 20.05.2026).
- Miro | Онлайн-доска // 2025. URL: <https://miro.com/> (дата обращения: 08.05.2026).



Miro - доска
задач



сайт

демонстрация



репозиторий



документы

