**Техническая документация**

**1.Введение:**

Название проекта: TaskTraker

TaskTraker - решение для автоматизации и оптимизации, планирования и маршрутизации задач. На основе алгоритмов маршрутизации и анализа данных, TaskTraker позволяет рационально распределять задачи между сотрудниками, минимизируя время и ресурсы. Сервис предоставляет API для взаимодействия с бэкендом, а также интуитивно понятный веб-интерфейс для управления и мониторинга задач.

**2.Ссылки:**

- [ссылка на frontend сайта](https://github.com/kate-savinkova/entrivia)

- [ссылка на flutter](https://app.flutterflow.io/project/sovk-k2ymet?tab=uiBuilder&page=HomePage)

- [ссылка на дизайн мобильного приложения](https://www.figma.com/proto/8RF6os2AWqDDk4YcgTUPl3/%D0%A1%D0%BE%D0%B2%D0%BE%D0%BA?page-id=0%3A1&type=design&node-id=250-13042&viewport=2333%2C-786%2C0.21&t=0LUvbKFHY8d4ZEIy-1&scaling=scale-down&starting-point-node-id=29%3A589&show-proto-sidebar=1&mode=design)

-[ссылка на дизайн сайта менеджера](https://www.figma.com/proto/8RF6os2AWqDDk4YcgTUPl3/%D0%A1%D0%BE%D0%B2%D0%BE%D0%BA?page-id=342%3A18826&type=design&node-id=342-18886&viewport=3619%2C-2513%2C1&t=dCIPxlZ2m6OAPDI0-1&scaling=scale-down&starting-point-node-id=342%3A18886&mode=design)

-[ссылка на backend](https://github.com/entrivia/backend_hack)

**3.Технологии:**

1.База Данных:

* SQL Сервер на cloud.ru
* Реляционные таблицы для хранения данных.

2.Серверное администрирование:

* Сервер для работы с данными на cloud.ru

3.Бэкенд:

* Язык программирования GO для написания основного кода бэкенда.

4.Фронтенд для менеджера:

* Сайт, основанный на React, Bootstap, предоставляющий интерфейс для менеджеров.

5.Мобильное приложение для сотрудников:

* Приложение разработано на Flutter для обеспечения мобильной функциональности

6.Docker

* Мы используем контейнеризацию Docker для упаковки и доставки нашего приложения, обеспечивая тем самым надежность и удобство в развертывании в различных средах

**4.Описание работы алгоритма:**

Для построения оптимального маршрута между точками для выездного сотрудника мы используем:

1) Собрать данные:

• Приоритет задачи.

• Опыт сотрудника.

• Продолжительность рабочего дня.

• Место работы сотрудника.

• Адреса агентских точек.

• Общие временные затраты на дорогу.

2)Определить веса:

• Назначить вес каждому критерию в соответствии с его важностью для определения оптимальности маршрута.

3)Назначить баллы:

• По каждому критерию присвоить баллы каждой задаче или точке на маршруте.

4)Нормализация баллов:

• Нормализовать баллы, чтобы они лежали в одном диапазоне.

5)Вычислить общий балл:

• Суммировать нормализованные баллы для каждой задачи или точки.

6)Составить маршрут:

• Использовать жадный алгоритм чтобы выбрать порядок посещения точек на основе общего балла.

7)Оптимизация маршрута:

• Применить алгоритмы оптимизации TSP (Traveling Salesman Problem), чтобы дополнительно улучшить маршрут.

8)Вывод маршрута:

• Вернуть оптимальный маршрут для выездного сотрудника.

**5.Контакты:**

Капитан: Игорь Тараев

Telegram: @k3k13