



AI CHALLENGE ОТБОРОЧНЫЙ КЕЙС

В компанию КРИТ обратился заказчик — владелец средней логистической компании. Он просит нас разработать ИТ решение для автоматизации процесса перевозки автомобилей. Он изложил свои требования в следующем виде.

У меня есть компания по перевозке дорогих люкс-автомобилей ООО «КИБЕРТРАКС». Перевозка автомобилей выполняется закрытыми автовозами, в каждый из которых помещается 1 автомобиль. Пользуется повышенным спросом перевозка автомобилей из Бишкека в Москву, а также из Москвы в Сочи.

Раньше я работал на крупные сетевые автосалоны, но в связи с падением маржи решил выйти на рынок частных перевозок и предлагать услуги физлицам.

Я хочу **сделать веб-сайт**, на котором любой человек мог бы зарегистрироваться, указав фео, телефон и заказать перевозку своего автомобиля из Города А в Город Б.

Сайт должен быть **красивый, в современном дизайне!** На сайте он может выбрать (или ввести) марку и модель своего автомобиля: допускается свободный ввод или небольшой встроенный список. Далее он должен указать город начала перевозки, дату начала перевозки и город окончания перевозки.

Система рассчитывает цену перевозки автоматически:

- По тарифу **150 рублей за 1 км** при расстоянии между городами **до 1000 км**, и **100 руб/км** при расстоянии между городами **больше 1000 км**
- Стоимость за 1км зависит от месяца начала перевозки и настраивается мной в отдельном окне, так как перевозка в зимние месяцы стоит дороже
- Стоимость перевозки по маршруту Москва — Сочи фиксирована и составляет 200000 рублей, вне зависимости от сезона
- Стоимость перевозки Бишкек — Москва составляет 350000 рублей. Также система считает стоимость страховки, которая составляет 10% от стоимости перевозки
- Система показывает время перевозки в часах и днях, дату прибытия в пункт назначения, исходя из норматива что за 24 часа автовоз проезжает 1000 км

После того как пользователь указал все данные, система выводит итоговый экран заказа.

После подтверждения заказа открывается окно оплаты-симулятора, на ней требуется реализовать форма ввода данных карты.

- Необходимо определять платежную систему по первой цифре номера карты
- Есть одна кнопка «Оплатить», по нажатию которой заказ помечается как успешно оплаченный
- Мы пока не подключили процессинг, поэтому в первой версии все оплаты считаются успешными

Мои автобазы пока еще есть не во всех крупных городах, поэтому я должен иметь возможность **настраивать списки городов из/в, которые возможны перевозки**. Автовозов много, а если мои все заняты, то я могу арендовать их у партнеров. Поэтому при наличии заказа я всегда смогу найти автовоз и проверять доступность автовозов в городе назначения не требуется.

В любой момент времени я могу посмотреть реестр всех заказов на перевозку на любую выбранную дату. Я могу **фильтровать заказы по периодам, городам, статусу оплаты** (мне важно видеть какие заказы не оплачены, чтобы обзванивать этих клиентов). Я также могу сам изменить статус оплаты, так как некоторые заказы оплачиваются не через банк. Я бы хотел иметь возможность **сортировки заказов по датам и стоимости**.

- Если заказ еще не уехал (дата начала перевозки позже выбранной даты), то он подсвечивается **зеленым**
- Если заказа в пути (выбранная дата находится между датами начала и завершения перевозки) – **желтым**
- Если заказ прибыл (выбранная дата > даты завершения перевозки) – **синим**

Все мои автовозы прибывают и отбывают в плановые сроки - я за этим строго слежу. Если я **нажму на заказ**, то я вижу по нему **всю информацию** - фамилия клиента, автомобиль и т.д.. ч

Я бы хотел получить гибкое современное решение, которое потом смогу развивать силами своего отдела ит! Поэтому мне также нужна документация по решению, чтобы мои айтишники быстро разобрались (не знаю, что им нужно, но важно чтобы они потом могли сами все развивать).

Уточнение Архитектора по итогам общения с Заказчиком: Заказчик сказал, что если что-то непонятно, то мы можем сами подумать, как будет лучше и сделать как мы считаем правильным, но помнить про сжатые сроки!

УТОЧНЕНИЯ ДЛЯ ОТБОРОЧНОГО ЭТАПА

- Источник расстояний - разрешается использовать внешний API маршрутизации с бесплатным тарифом или офлайн-матрица JSON
- **Марка/модель авто:** свободный ввод текста или выбор из небольшого встроенного списка; интеграции с внешними каталогами не требуются
- **Оплата:** симуляция без ввода карточных данных. Кнопка «Оплатить» устанавливает заказ в оплаченный. Все оплаты успешны
- Реализация уведомлений пользователю не обязательна
- Администрирование (города, тарифы, реестр): простая авторизация без внешних провайдеров (без Keycloak). Достаточно фиксированного логина/пароля в коде для доступа к админ-разделу
- Писать код только с LLM (рекомендуется IDE Cursor)
- **Бэкенд:** Самостоятельный выбор языка
- **Веб-клиент:** обязателен (JS + React)
- «Красоту и современность» сделает LLM, достаточно черновых макетов

ПРЕДЛАГАЕМАЯ МОДЕЛЬ ДАННЫХ

- **Пользователь:** ФИО, телефон
- **Заказ:** ссылка на пользователя; автомобиль (бренд/модель текстом); город отправления; город прибытия; дата начала перевозки; рассчитанное расстояние (км); применённый тариф (цена за км); признак фикс-маршрута; стоимость перевозки; стоимость страховки (10% от стоимости перевозки); расчёт длительности и ETA; статус оплаты (оплачено/не оплачено/установлено вручную)
- **Тарифы по месяцам:** для каждого месяца — цена за км при расстоянии ≤ 1000 км и > 1000 км; страницы CRUD в админке
- **Города:** справочник доступных городов (из/в которые возможны перевозки), CRUD в админке

АРТЕФАКТЫ ДЛЯ СДАЧИ

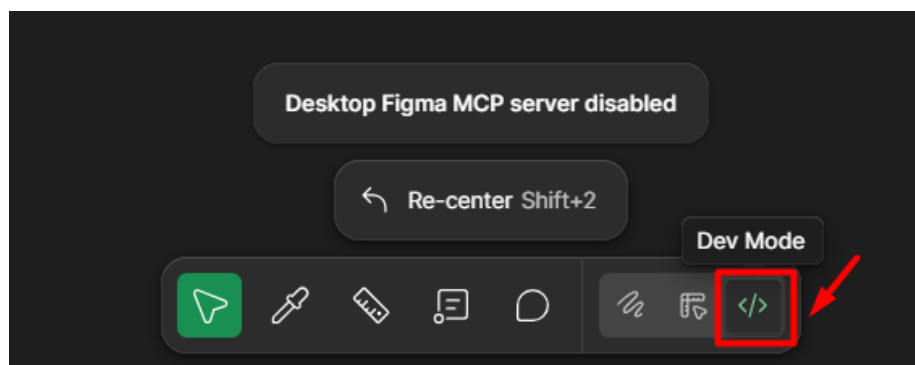
- Направлять на hr@krit.pro с темой письма «AI-challenge_Фамилия»

- Черновые макеты ключевых экранов FIGMA в PDF
- Исходный код в Git: репозиторий в предоставленной группе
- Схема БД
- User-story: пользовательских сценариев
- Swagger

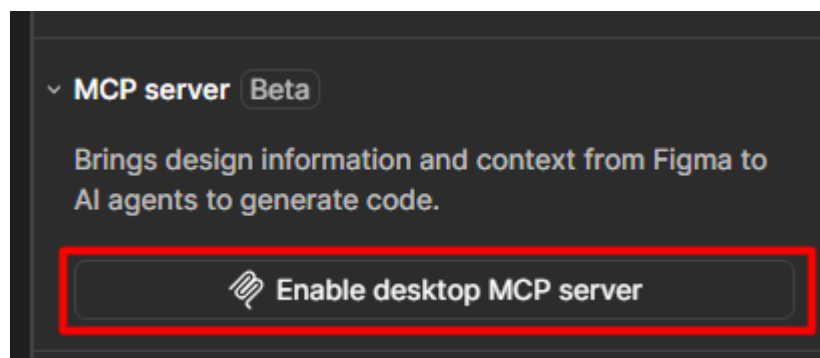
ИСПОЛЬЗУЕМЫЕ ИСТОЧНИКИ

- [Cursor](#) – скачивание по гиперссылке
- FIGMA MCP

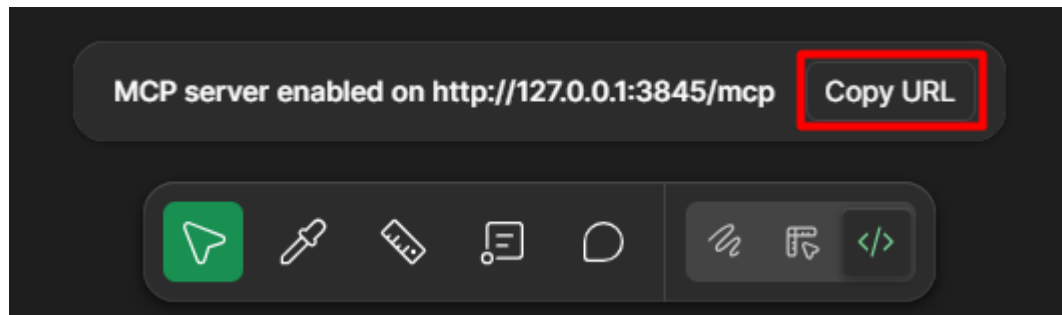
1. Для подключения Figma к Cursor необходимо активировать dev-mode (УЗ фигмы будет предоставлена участвующим в отборе)



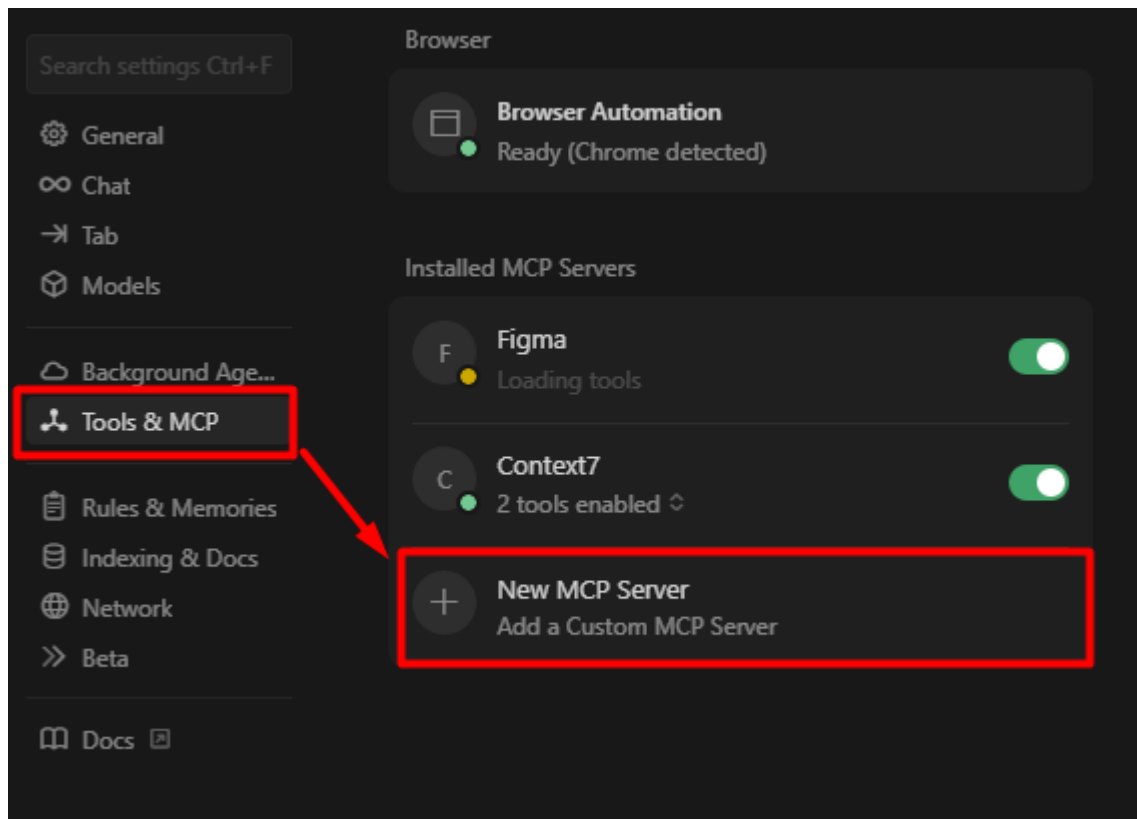
2. Активировать функцию «Enable desktop MCP server»



3. После чего скопировать URL



4. Зайти в настройки Cursor, нажать «New MCP Server»



5. Вставить значение в текстовый файл mcp.json

```
{  
  "mcpServers": {  
    "Figma": {  
      "url": "Вставить URL"  
    },  
  },  
}
```