# **TD. Backend. Тестовое задание**

## Основная задача

**Разработать только Backend. (Не нужно создавать фронтенд, пользователей, админ-панель)**

Требуется создать API для витрины автомобилей, на которой можно оформить заявку на кредит

Исходники приложения выложить на github.com

После выполнения всех этапов скинуть ссылку на репозиторий для проверки

Написать сколько примерно времени заняло выполнение задания

## Описание приложения

На фронтенде есть список автомобилей. На каждом автомобиле находится кнопка “Рассчитать кредит”

При нажатии на кнопку открывается страница с детализацией авто с возможностью рассчитать кредит

Расчет кредита зависит от Кредитной программы, Первоначального взноса и срока кредита

Как только пользователь выберет кредитную программу, он нажимает на кнопку “Отправить заявку”

## Требования

Создать API для Frontend команды

API имеет следующие endpoint’s:

1. Получение всего списка авто
2. Получение одного авто с детализированной информацией
3. Расчет кредита по указанным параметрам из кредитной формы
4. Сохранение заявки с указанными параметрами из кредитной формы

## Описание endpoint’s

1. Список автомобилей:

У каждого автомобиля должны быть следующие данные: бренд, фото, цена

|  |
| --- |
| GET /api/v1/cars  ANSWER: [  {  "id": int,  "brand":  {  "id": int,  "name": str  },  "photo": str,  "price": int  },  ...  ] |

2. Получение одного авто с детализированной информацией

У каждого автомобиля должны быть следующие данные: бренд, модель, фото, цена

|  |
| --- |
| GET /api/v1/cars/{id}  ANSWER: {  "id": int,  "brand":  {  "id": int,  "name": str  },  "model":  {  "id": int,  "name": str  },  "photo": str,  "price": int  } |

3. Расчет кредита по указанным параметрам из кредитной формы

|  |
| --- |
| GET /api/v1/credit/calculate?price={price}&initialPayment={initialPayment}&loanTerm={loanTerm}&monthlyLoanPayment={monthlyLoanPayment}  ANSWER: {  "programId": int,  "interestRate": int,  "monthlyPayment": int,  "title": str  } |

Входные данные в запросе:

* **price**: Цена автомобиля (int, required). Пример: 1401000
* **initialPayment**: Первоначальный взнос за кредит. В запросе отдаются рубли с копейками. (float: точность до десятых долей, required). Пример: 200000,56
* **loanTerm**: Срок кредита в месяцах (int, required). Пример: 64
* **monthlyLoanPayment —** ежемесячный платёж по кредиту, выбираемый пользователем (см. скрин «готов платить в месяц» )

Ответ запроса на расчет ежемесячного платежа

* **programId**: Уникальный id кредитной программы
* **interestRate**: Процентная ставка кредитной программы (float, точность до десятых долей). Пример: 12.3
* **monthlyPayment**: Ежемесячный платёж (int). Пример: 24276
* **title**: Название кредитной программы (str). Пример: “Alfa Energy”

**Алгоритм расчета кредитных на Ваше усмотрение**

Пример: если первоначальный взнос более 200000 р., платеж в месяц до 10000 р., срок кредита менее 5 лет – выводить программу с процентной ставкой 12.3, платеж в месяц от 9800 р., иначе любую другую кредитную программу

Кредитные программы должны храниться в БД

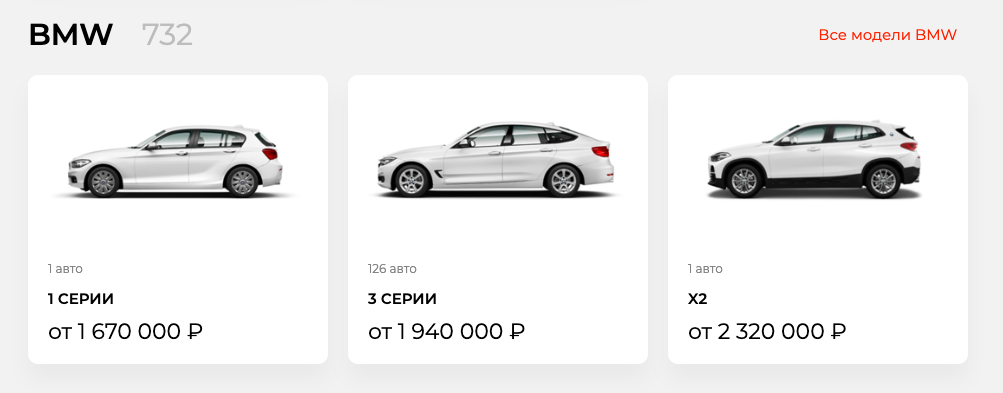
4. Сохранение заявки с указанными параметрами из кредитной формы

|  |
| --- |
| POST /api/v1/request  BODY: {  "carId": int,  "programId": int,  "initialPayment": int,  "loanTerm": int  }  ANSWER: {“success”: bool} |

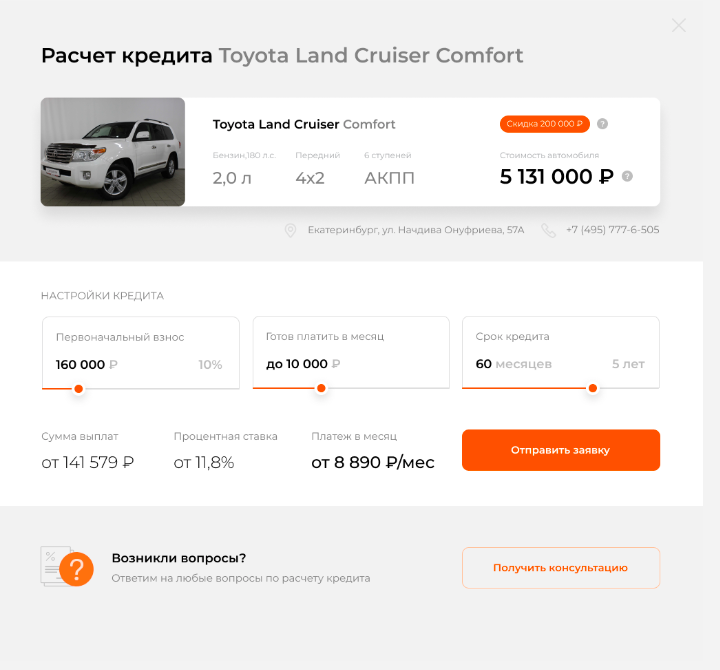
## Стек технологий

1. PHP 7.+
2. Framework: Symfony (Желательно), Laravel, YI2
3. Обернуть в Docker (Желательно)
4. БД: Mysql, MariaDB (Желательно), Postgresql

## Макеты для визуального понимания приложения

1. Список автомобилей  


2. Подробная страница с автомобилем



3. Уведомление об успешной отправки заявки

