

Бэкенд-часть продукта Easy Go

Используемый технологический стек:

- Node.js
- Express.js
- Express Validator
- JWT
- MongoDB + Mongoose(опционально)
- Geolocation API
- Generic Sensor API
- Twilio API
- AstraLinux

Пояснение выбора **тех. стека**:

Node.js имеет высокую производительность и позволяет обрабатывать большое кол-во одновременных запросов, что и нужно в нашем приложении. Далее, с ним легче будет взаимодействовать при написании Frontend части приложения. Так же Node.js поддерживает асинхронное программирование, а значит позволяет выполняться операциям параллельно.

Express.js нужен для возможности запуска веб сервера при помощи Node.js, упрощает создание серверной части, обеспечивает удобный интерфейс для работы с HTTP запросами.

Express Validator для валидации запросов с клиентской стороны

JWT для создания безопасных токенов аутентификации и авторизации

MongoDB + Mongoose – для реализации БД

Geolocation API используется для трэкинга местоположения

Generic Sensor API используется для получения данных со всех возможных датчиков, разрешенных для использования в браузере

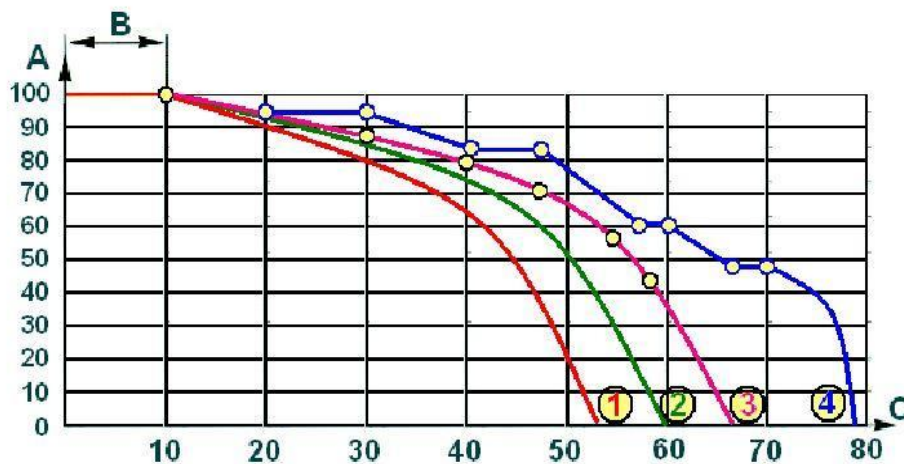
AstraLinux – ОС для сервера

Twilio API – для отправки смс при регистрации/авторизации и осуществления звонков

Как будет работать бэк часть:

Благодаря **Generic Sensor API** мы имеем доступ к LinearAcceleration sensor (датчику линейного ускорения).

Сравнительная величина тормозного пути при различных вариантах торможения.



- Торможение с применением ABS;
- Торможение с полной блокировкой колес;
- Дозированное торможение;
- Прерывистое торможение;
- Скорость движения автомобиля;
- Суммарный путь, пройденный автомобилем до начала эффективного торможения;
- Расстояние, пройденное автомобилем до полной остановки.

3

Как мы видим по графику, при экстренном торможении автомобиль начинает резко терять скорость, что означает сильное изменение линейного ускорения. Мы предлагаем сравнивать ускорения полученные какое-то определенное незначительное количество времени назад с текущим для определения резко снижения скорости. Точных значений для времени или какого-то экстренного значения ускорения мы назвать не можем, т.к. это требует проведения тестов.

Принцип действия программы от регистрации до вызова экстренных служб. Пользователь регистрируется -> его личные данные(паспорт, транспортные средства заносятся в БД) -> нажимает кнопку “Поехали” -> Сервер начинает следить за его датчиками -> при обнаружении резкого ускорения показывается окно “Зафиксировано падение” -> идет отсчёт времени -> по истечении времени формируется сообщение для звонка при помощи Geolocation API с местоположением и личными данными пользователя и его транспортного средства -> отправляется звонок на номер 103 с сообщением типа “Данный звонок сформирован роботом, по адресу г.Москва

ул.Коштянца д.6 произошло ДТП с Ивановым Иваном Ивановичем на автомобиле марки Mitsubishi синего цвета. -> Повторяет 3 раза, затем заканчивает звонок.