Возможные варианты стека для реализации backend:

* **Node.js (Express.js1)**

Node.js обеспечивает высокую производительность благодаря асинхронной модели выполнения и широкому выбору библиотек. Он хорошо подходит для создания API и обработки запросов. Express.js, в свою, очередь легкий и гибкий фреймворк для создания RESTful API с хорошей документацией и активным сообществом.

* **Node.js (Gatsby)**

Отлично подходит для создания статических сайтов и одностраничных приложений (SPA) с использованием React.

* **Django**

Высокоуровневый и полноценный фреймворк, который предоставляет множество готовых решений.

* **Flask**

Легковесный и гибкий фреймворк для создания RESTful API. Он не навязывает структуру проекта и позволяет выбрать нужные библиотеки.

* **ASP.NET**

Мощный фреймворк на C# для создания веб-приложений, обеспечивающий высокую производительность и безопасность.

Возможные варианты БД:

* **PostgreSQL**

Мощная, надежная и безопасная реляционная база данных с богатыми возможностями запросов и поддержкой многих типов данных.

* **MongoDB**

Гибкая NoSQL база данных, хорошо масштабируется и позволяет хранить неструктурированные данные. Быстрые операции чтения и записи.

Был выбран стек Node.js (Express.js) + PostgreSQL по нескольким причинам:

1. **Производительность:**

Node.js — это среда выполнения JavaScript, которая обеспечивает высокую производительность благодаря асинхронной модели выполнения. Это означает, что наше приложение сможет обрабатывать множество запросов одновременно.

1. **Широкий выбор библиотек и модулей:**

Node.js имеет активное сообщество разработчиков и богатую экосистему библиотек, а Express.js предоставляет базовые инструменты для создания API, не навязывая при этом жестких структур, что делает разработку более эффективной.

1. **Поддержка WebSocket:**

Node.js с легкостью поддерживает протокол WebSocket, необходимый для реализации real-time обновления событий и задач.

1. **PostgreSQL:**

PostgreSQL — это мощная и надежная реляционная база данных. Она предоставляет ACID-совместимость, что означает, что данные будут целостными и надежными. Также PostgreSQL поддерживает многие расширения и типы данных, что может быть полезно при моделировании структуры данных для TODO-листа. Использование NoSQL баз данных не является преимущественным, поскольку все данные TODO-листа строго структурированы.

1. **Кроссплатформенность:**

Node.js поддерживается на разных операционных системах, что позволяет развертывать ваше приложение на различных хостинговых платформах.

1. **Преимущества перед другими опциями:**

Express.js, в отличии от Gatsby.js, не ограничен спецификой одностраничных приложений, а также лучше взаимодействует с PostgreSQL. Поскольку он существует большее время, то обладает лучшей документацией и более обширным сообществом.

В итоге, выбор Express.js и PostgreSQL для создания TODO-List обоснован их высокой производительностью, широким инструментарием, простотой в разработке и кроссплатформенностью.

1. Особенности работы и внутреннего устройства express.js // Хабр URL: <https://habr.com/ru/companies/ruvds/articles/414079/> (дата обращения: 17.09.2023).
2. Подойдет ли PostgreSQL вообще всем проектам или нужны альтернативы // Хабр URL: <https://habr.com/ru/companies/selectel/articles/740680/> (дата обращения: 17.09.2023).