



Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ _____ Информатика и системы управления _____

КАФЕДРА _____ Программное обеспечение ЭВМ и информационные технологии _____

ОТЧЕТ ПО УЧЕБНОМУ ПРАКТИКУМУ

Студент	_____	<u>Подпужников Н.В.</u>
	<i>подпись, дата</i>	<i>фамилия, и.о.</i>
Студент	_____	<u>Сучкова Т.М.</u>
	<i>подпись, дата</i>	<i>фамилия, и.о.</i>
Студент	_____	<u>Параскун С.Д.</u>
	<i>подпись, дата</i>	<i>фамилия, и.о.</i>
Студент	_____	<u>Платонович Е.В.</u>
	<i>подпись, дата</i>	<i>фамилия, и.о.</i>
Ментор команды	_____	<u>Хетагуров П.К.</u>
	<i>подпись, дата</i>	<i>фамилия, и.о.</i>
Руководитель практики	_____	<u>Оленев А.А.</u>
	<i>подпись, дата</i>	<i>фамилия, и.о.</i>

2020 г.

Введение

Генерация идеи

Идея реализовать данный todo-list пришла после того, как мы поняли, что есть удобные варианты, однако многие из них имеют ряд недостатков, таких как небольшая вложенность, невозможность расшарить задачу на несколько пользователей. Мы решили исправить эти недостатки и сделать свой todo-list, реализующий наши потребности.

Разработка концепции и ее проверка

На первых этапах шла разработка концепции продукта. Также был придуман дизайн сайта, который бы позволял пользователям концентрировать внимание на самой сути.

Нашей командой был опрошен некоторый круг лиц, которые поддержали нашу идею, что помогло утвердить ее.

Предполагаемые целевые пользователи: нуждающиеся в организации своего времени и ищущие сайты, которые будут удобны, просты в использовании и обладают нужным функционалом.

Цель: Сервис формирования пользователем индивидуальных задач, позволяющий отслеживать стадию их выполнения.

Задача: Создание сайта, реализующего поставленную цель.

Роли в команде:

- Подпружников Никита – тимлид + frontend-разработчик
- Татьяна Сучкова – дизайнер + frontend-разработчик
- Параскун София – backend-разработчик
- Платонович Егор – тестировщик

Аналитический раздел

Анализ существующих решений

Рассмотренные существующие решения, реализующие аналогичные или смежные функции:

- Google Tasks – Список дел от гугла (Малая вложенность, нет расшаривания)
- Evernote – продвинутый блокнот (Малая вложенность, нет расшаривания)
- DropBoxPaper – продвинутый блокнот (Малая вложенность, нет расшаривания)
- Jira – борд (Целевая аудитория программисты, обычные пользователи не используют)

Их достоинства и недостатки

Каждое из решений реализует фрагментарный набор функций todo-list`а, многие не имеют достаточной вложенности или нацелены на другую аудиторию.

Постановка задачи

Написать сайт покрывающего, по возможности, все вышеупомянутые функции.

Конструкторский раздел

Декомпозиция задачи

- Разработка интерфейса – Никита, Таня
 - Разработка графического интерфейса
 - Разработка модальных окон и кнопок для взаимодействия с пользователем
- Реализация серверной части - София
 - Настройка сервера
 - Связь с базой данных
 - Формировка ответа клиенту при получении запроса
- Связь клиентской и серверной части - Никита
 - Добавление групп
 - Удаление задач
 - Добавление задач

Компоненты продукта

- Клиентская часть – организация взаимодействия с пользователем;
- Серверная часть – предоставление доступа к данным из БД клиентской части;
- База Данных – хранение всех данных о группах, задачах и пользователях.

Технологический раздел

Технические средства

Был выбран JS в качестве языка разработки. Выбор был основан на том, что это не сложный язык, на котором можно писать, как серверную, там и клиентскую часть. Так же предоставляет удобные библиотеки для тестирования продукт

Использованные библиотеки, фреймворки и сторонние модули:

- MERN – стек библиотек и технологий для написания сайта.
- Eslint – библиотека для автоматической проверки code style
- Axios – запросы на сервер из клиентской части
- React-router – переход между страницами
- Prop-types – валидация типов передаваемых из компонента в компонент данных

Модель разработки

Нами была выбрана итерационная модель разработки, так как она позволяет легко внедрять новые фичи в проект и так же быстро исправлять ошибки в только что добавленных.

Реализация программного продукта

Клиентская часть была разбита на 2 страницы. Главная страница, на которой пользователь видит список всех задач у него и так же может добавлять задачи и группы. И так же страница задачи, на которой пользователь видит задачу и все подзадачи у неё. Все задачи и группы выводятся из базы данных. Все страницы были разбиты на компоненты. Например, общий компонент Navigation который отображается слева на обеих страницах. Это компонент, в котором выведен список групп, которые есть у пользователя.

В качестве backend основы на выбор было два варианта: MySQL и MongoDB, который мы и выбрали. Почему - так как noSQL базы являются более гибкими в использовании, позволяя хранить каждую запись как отдельный объект, а не элемент таблицы. Также MongoDB входит в состав MERN-стека, что позволяет упростить связь фронта и бэка.

Помимо этого, стоял еще один вопрос - хранить базу данных локально или удаленно. Так как нам была важна работа сайта в автономном режиме, мы воспользовались предложением Amazon`а, предоставляющим бесплатную версию на 512MB.

Так же был выбран nodeJS, так как он хорошо сочетается с выбранными для frontend`a React и Express и легок в понимании.

Как было написано выше, были реализованы все CRUD-методы для созданных коллекций Users, Groups, Tasks, Undertasks. Была продумана взаимосвязь этих компонентов, корректный вывод ответа.

Тестирование

Для линтера использован code formatter Prettier и Eslint linter. Плюсы - Eslint позволяет проводить анализ качества вашего кода, написанного на любом выбранном стандарте JavaScript. Умеет автоматически исправлять многие из найденных проблем, интегрируется с Prettier.

Был настроен gitlab-ci, в качестве executor был выбран shell (в связи с недостаточным пониманием о том, как настраивается docker.)

Плюсы — простота.

Минусы - чтобы такая система работала как подразумевает continuous integration моя машина должна быть все время включена, плюс может случиться то, что случилось у меня на второй попытке настройки gitlab-runner, так как все настройки происходят на локальной машине.

Была поставлена задача написать некоторые модульные тесты, выполнить которую не удалось в связи с нехваткой времени.

Развёртывание программного продукта

Развертывание было произведено на бесплатном хостинге heroku.

Заключение

В ходе выполнения проектной работы были некоторые трудности, в связи с которыми не удалось реализовать часть функционала. В какой-то момент нас покинул один из участников нашей команды. Он перестал выходить на связь, и из-за этого нам пришлось часть его работы делать в дополнение к нашей, из-за чего дедлайны сдвинулись и некоторые задачи реализовать не удалось.

В ходе реализации проекта была получена масса опыта. Например, опыт командной разработки, опыт тестирования, опыт разработки с использованием системы GIT и так далее. Получили опыт работы с JavaScript и разобрались со многими популярными библиотеками, например React, Axios, mongoose и так далее.