

Национальный исследовательский университет
«Высшая школа экономики»
Лицей

Индивидуальная выпускная работа
ИТ-ПРОЕКТ: Сервис для работы с текстом Spectre
<https://github.com/aijdwr127/ivr>

Выполнил: Хасанов Айдар Рифкатович
Группа: ИИ2

Москва – 2023

1. Введение

Я начал заниматься разработкой веб-приложений на Node.js относительно недавно, примерно год назад. Сначала это были простые TODO-приложения, на которых я учился работать с различными базами данных (например, PostgreSQL, MongoDB), потом проекты становились сложнее, во многих из них уже требовалось реализовывать безопасную регистрацию и авторизацию, работу с cookie-файлами и различными сторонними библиотеками.

Проект изначально задумывался как вспомогательное средство при работе с разными текстами. Мысль о создании подобного сервиса мне пришла еще в 9 классе на уроке литературы. Я обнаружил, что чтение книги параллельно с прослушиванием этой же аудиокниги в разы повышает количество запоминаемой информации. Это и стало отправной точкой. Кроме того, лично мне всегда были интересны методы повышения своей продуктивности, в частности, использование полезных программ и приложений, поэтому Spectre я делал для себя и своих единомышленников.

2. Проблемное поле

Тщательно исследовав рынок аналогичных продуктов, я пришел к выводу, что приложения, реализующего те же функции, что и Spectre, нет. Перечислю некоторые из проблем данных сервисов:

1. Малое количество поддерживаемых языков
2. Крайне небольшой лимит на количество слов / символов в документе
3. Невозможность добавления заметок
4. Отсутствие собственной библиотеки документов
5. Отсутствие всякого взаимодействия с документами и другими пользователями
6. Невозможность изменения скорости или громкости прослушивания документа
7. Медленная обработка текстов и аудиофайлов
8. Невозможность добавления документов в виде файлов
9. Старомодный дизайн, который сильно затрудняет использование сервиса
10. Недоступность в России

3. Образ продукта

Программный продукт предоставляет следующие возможности:

1. Регистрация нового пользователя (*решение проблемы 5*)
2. Добавление названия документа

3. Добавление текста документа в виде файла форматов “.doc”, “.docx” или непосредственно в форму (*решение проблемы 8*)
4. Выбор различных языков (английский, русский, французский, немецкий), необходимый для спецификации документа и задания параметров прослушивания (*решение проблемы 1*)
5. Возможность регулировать воспроизведение аудиофайла документа: увеличить или уменьшить громкость, увеличить или уменьшить скорость воспроизведения, возможность поставить на паузу или запустить аудиофайл снова (*решение проблемы 6*)
6. Возможность добавления или удаления заметок для какого-то определенного документа (*решение проблемы 3*)
7. Возможность добавления различных документов из общей библиотеки сервиса в свою собственную (*решение проблем 4, 5*)
8. Возможность поиска документов из общей библиотеки сервиса по заданным параметрам. Возможен поиск по автору или по подстроке в названии документа (*решение проблем 4, 5*)

4. Описание продукта

1. Регистрация пользователя

Классический сценарий во многих современных приложениях. При его реализации соблюдались все меры для обеспечения безопасности информации пользователей: пароли и уникальные идентификаторы пользователей хешировались, а передача информации из формы в базу данных осуществлялась посредством post-запроса.

2. Авторизация пользователя

Данный сценарий практически идентичен предыдущему. После проверки на валидность введенных данных пользователь начинал новую сессию (чтобы не терять сохраненный прогресс), что осуществлялось посредством cookie-файлов с захешированным “секретом”.

3. Добавление записи

С помощью данной функции пользователь может добавить свой документ в общую библиотеку сервиса. Автором записи, что логично, будет отмечен сам пользователь. Можно выбрать один из 4 доступных языков, этот выбор впоследствии повлияет на аудиозапись документа. Как уже было упомянуто выше, в данном сценарии реализована в том числе работа с файлами формата “.doc”, “.docx”.

4. Поиск по записям

В рамках данного сценария пользователь может найти любые записи по названию или никнейму (логину) автора. Все полученные результаты выводятся на экран.

5. Загрузка записи из библиотеки

Является продолжением предыдущего сценария. Можно добавить или, если запись уже была добавлена пользователем, удалить ее из собственной библиотеки.

6. Просмотр добавленных записей

Пользователь может увидеть весь список добавленных им в свою библиотеку записей.

7. Работа с документом

В рамках этого сценария пользователь совершает основные действия с документом: он может прочитать его, прослушать или добавить к нему какие-то заметки. Также пользователь может выбрать различные опции по прослушиванию документа. Полный список доступных функций озвучен в демонстрации пользовательских сценариев.

8. Редактирование заметок

Пользователь может редактировать заметки в рамках конкретного документа: добавить, удалить или прочитать их.

5. Детали реализации бэкенда

Бэкенд целиком и полностью реализован на **MongoDB**. Основные преимущества MongoDB:

- ☐ Схожий с JSON формат документов, что упрощает процесс интеграции с JS
- ☐ Удобный модуль `mongoose.js`, позволяющий быстро создавать схемы для объектов в БД
- ☐ Простота использования
- ☐ Высокая производительность

6. Средства разработки

Языки, использованные при разработке:

- ☐ HTML
- ☐ CSS
- ☐ Javascript

Средой выполнения была избрана Node.js. Вся бэкенд логика была написана в ней.

Некоторые плагины и фреймворки, использованные при разработке:

- ☐ `handlebars` (для рендера html-компонентов на страницу)
- ☐ `express.js` (для настройки сервера)
- ☐ `mongoose.js` (для работы с БД)

- ☐ bcrypt.js (для хеширования)
- ☐ gtts.js (для работы с технологией text-to-speech)

Инструменты разработки:

- ☐ Sublime Text 3 (редактор кода)
- ☐ Visual Studio Code (редактор кода)
- ☐ MongoDB Compass (графический интерфейс БД)
- ☐ Postman (для тестирования бэкенда)

7. Этапы работы

1. Разработка пользовательских сценариев

В дедлайн уложиться не удалось. В числе первых были добавлены сценарии регистрации и авторизации. По мере разработки добавлялись новые сценарии. Некоторые нереалистичные сценарии были удалены.

2. Проектирование интерфейса

Интерфейс был спроектирован сразу после реализации бэкенда. Основной фишкой дизайна приложения является использование инструментария bootstrap. Благодаря этому можно было не заикливаться на прописывании стилей и сосредоточиться на реализации.

3. Тестирование, отладка

Тестирование осуществлялось сразу же после добавления каждой функции. Окончательное тестирование приложения было проведено в самом конце, после добавления всего функционала приложения. В процессе тестирования было выявлено несколько багов в работе функций сохранения документов, которые впоследствии были исправлены.

4. Подготовка проекта к защите

Проходит сейчас. Итоговая презентация продукта в процессе написания.

8. Рефлексия

Проблемы:

- ☐ Изначально в проекте предполагалось использование модуля [Google Text-to-Speech](#). В связи с некоторыми политическими событиями, его пришлось заменить на более слабый модуль gtts.js.

- В первой версии плана сервиса была возможность добавления и конвертации “.erub” и “.fb2” файлов. К сожалению, файлы этих форматов довольно сложно парсить без потери важной информации, поэтому эта функция была упразднена.

Пути развития:

- Создание нативного приложения (для мобильных и десктопных устройств).
- Реализация парсера для файлов различных форматов с целью повышения гибкости сервиса.
- Добавление верификации по почте; функции восстановления пароля.

Приобретенные компетентности:

- Умение работать с аудио- и текстовыми файлами посредством Node.js.
- Умение работать с БД.
- Понимание некоторых аспектов безопасности веб-приложения.

Данные умения помогут мне в будущем быстро создавать продвинутое веб-приложение и легко масштабировать уже имеющиеся проекты.

Риски:

Многие риски действительно осуществились: острая нехватка времени для реализации некоторых функций, невозможность использования API очень удобных сервисов и постоянно поджимающие дедлайны. Впрочем, это не помешало проекту перейти в гавань завершенных.

9. Заключение

Время, проведенное за проектом, было потрачено не зря и с большим удовольствием!