**Компьютерная академия «ШАГ»**

**Днепропетровский филиал**

**Кафедра Разработки программного обеспечения**

**ДИПЛОМНЫЙ ПРОЕКТ**

**«BookTr – читалка книг с заучиванием слов»**

Группа: FD18

Студенты:

Загляда И.М.

Научный руководитель:

???

Кривой Рог 2019

Аннотация

Разработан сайт «Перевод книг FB2 с функцией заучивания слов». Сайт предназначен для людей, изучающих иностранные языки. В первую очередь, это конечно же изучение английского языка русскоязычными пользователями. Ключевыми требованием при разработке являлось:

1. *Переложить максимум нагрузки с сервера на сторону клиента (браузер), для обслуживания максимально возможного количества пользователей с минимальными затратами для владельца ресурса.*
2. *Сделать SPA-сайт с одной точкой фхода, после чего не перезагружать страницу.*

Как всем известно, изучать слова скушно и неинтересно, что ведет к демотивации учащихся. Совсем другое дело, когда изучение слов происходит в процессе чтения любимых книг. Именно этой цели и посвящена данная дипломная работа.

Сайт встречает пользователя страницей выбора родного и изучаемого языков (раздел по умолчанию «Выбор языка»). После выбора языков, происходит перелокализация приложения без перезагрузки страницы (AJAX).

Далее пользователь попадает в раздел «Библиотека». В этом разделе предоставляется возможность выбрать книгу формата FB2 с локального диска. Выбранная книга преобразовывается во внутренний формат, сохраняется во внутренней БД (IndexedDB), выводится в компонент «Чтение» куда и перенаправляется пользователь .

Находясь в разделе «Чтение», пользователь может читать книгу. Клик на слове выводит модальное окно с переводом и предложением изучить слово, либо добавить его в список «Изученные» . Изученные слова имеют прозрачный фон, выбранные для изучения помечаются желным цвепом, неизвестные светло-кирпичным цветом. Чем больше слов знает пользователей, тем меньше будет помеченных слов.

В любой момент пользователь может перейти в раздел «изучение слов» кликнув анимированное меню в правом верхнем меню экрана. Тут его встретит грид-компонент, содержащий все встреченные слова. Компонент обладает функцией постраничного отображения списка с возможностью быстрого перехода на любую страницу. Однако главной функцией раздела является изучение слов, перейти к которому можно нажав на кнопку «Начать изучение». Слова изучаются в прямом и обратном направлениях, после чего сохраняются в БД и перемаркировывается раздел «Чтение». Статистика всегда отображается в верхнем левом углу страницы.

Содержание

1. Введение 4
2. Техническое задание 5
3. Выбор технологии реализации проекта 7
4. Разработка сайта ... 8
   1. Разработка структуры системы 8
   2. Разработка базы данных 10
   3. Структура программного проекта 12
   4. Разработка интерфейса 13
   5. Руководство пользователя 15
5. Выводы 26
6. Список использованной литературы 27
7. Введение

**Зачем изучать иностранные языки?**

* Если говорить об английском языке, то он де-факто является международным языком, на котором Вас поймут в любой стране Европы, в Австралиии Америке. Во многих странах Азии и Африки. Вы можете объехать полмира владея одним лишь языком. Особенно ценен он для сферы IT, ведь вся самая свежая информация доступна на английском языке, самые денежные заказчики общаются на английском, да и просто промониторив рынок труда можно убедиться, что без английского языка никак не обойтись.
* Если говорить о других языках, то тут у каждого своя мотивция. Кто-то мечиает об эммиграции в Германию, либо Италию, кто-то желает прочитать книгу на языке оригинала, а кто-то просто расширяет свой кругозор.

2. Техническое задание

**Описание:**

«BookTr – перевод книг» – сайт для широкой аудитории желающих пополнить свой словарный запас. Пользователи читают книги, которые сами же загружают в свою библиотеку. По мере чтения добавляют слова в список «Для изучения». В любой удобный момент пользователь может перейти к заучиванию слов.

Перед разработчиком ставятся следующие технические требования:

1. Минимальная цена разработки, развертывания и содержания.
   1. Сайт должен вызывать минимальную нагрузку на сервер и быть неприхотливым к хостингу.
   2. Вся работа с контентом (книги) должна производиться на стороне пользователя. Не допускается загрузка книг на сервер. При этом после перезагрузки браузера не должна теряться библиотека пользователя и список заученных слов.
2. Сайт должен работать без перезагрузки страниц, т.е. быть классическим SPA-приложением.
3. Требования к браузеру не предъявляются. Допустимо использовать функционал самых современных браузеров Mozilla Firefox и Google Chrome.

**Список действий:**

1. Создать и клонировать репозиторий на GitHub.
2. Проектирования базы данных.
3. Создание базы данных.
4. Вывод информации на макет.
5. Глобальное тестирование и завершение всего проекта.

3. Выбор технологии для реализации проекта

Для создания сайта были выбраны следующие технологии:

* язык JavaScript
* язык разметки гипертекста HTML (HTML5)
* каскадные таблицы стилей (CSS3)
* технология обращения к серверу без перезагрузки страницы Ajax
* Google-translate – сервис, позволяющий получить перевод слов.
* IndexedDB – БД для хранения книг на стороне браузера.
* Promises – способ организации асинхронного кода — объект, представляющий результат успешного или неудачного завершения асинхронной операции.

Детальнее о выборе технологий:

* Язык JavaScript.   
  Это родной для всех браузеров скриптовый язык. Исходя из п.1 технического задания, пользуемся принципом «Быстро пишем прототим, а если взлетит, то переписываем прототип на более подходящем языке, типа TypeScript».
* HTML5, CSS3 – без них данный сайт не напишешь.
* Ajax – диктуется п.2 технического задания и необходимостью работать с Google Translate API.
* Google-translate – сервис, позволяющий получить перевод слов. Избавляет нас от необходимости составлять и поддерживать собственные словари, что весьма накладно. Перенос запросов на сторону браузера позволяет также издавиться от затрат, связанных с политикой Google на тарификацию за переводы.  
  Goole позволяет получить перевод фразы вызывав следующий запрос. https://translate.googleapis.com/translate\_a/single?client=gtx&sl=auto&dt=t&dt=bd&dj=1&text={{some%20tex}}t&tl=ru
* IndexedDB – IndexedDB – это современное NoSQL хранилище данные в формате JSON внутри браузера.   
  Проанализировав технологии, доступные на стороне браузера, приходим к выводу что альтернативы нет. Каждая книга формата FB22 занимает от 2 до 20МБ (в зависимости от количества вложенных изображений ). Это исключает возможность истользования Cookies (до 4х КБ) и Local Storage (обычно до 5ти МБ).   
  WebSQL использовать удобнее IndexedDB, но консорциум W3C прекратил работу над спецификацией в ноябре 2010 года, а это означает что не исключен отказ от поддержки технологии разработчиками тех браузеров, которые успели ее внедрить.
* Promises – Это буквально панацея от callback hall, который особенно ярко выражен при работе с IndexedDB. Без них код очень быстро становится запутанным и трудно поддерживаемым.

4. Разработка сайта

4.1. Разработка структуры системы

При разработке принято решение весь код организовать в виде псевдостатических классов JavaScript, которые относительно легко могут быть преобразованы в стандартные классы(если возникнет такая необходимость). Програмный код разделен на две крупных части, каждая из которых делится на мелкие составляющие. Составляющие приложения выстроены в иерархическую структуру, в которой крупные компоненты являются контейнерами бля меньших(см. Рис.1).

1. App – дерево статических классов, которые обслуживают непосредственно приложение. Это сам класс App – центральная точка входа и такие компоненты, как навигация, модалные окна, таблицы, адаптер БД, классы исключений, адаптер локализации.
2. Helper – дерево статических классов, которые являются библиотеками-помошниками. Это автономные единицы, которые могут без изменений переиспользоваться в других проектах. Сюда входят такие модули, как “Hash” для выполнения хеширования текста; “Google” помошник для работы с сервисами Google; “Xml” – для работы с данными формата XML; и т.д.

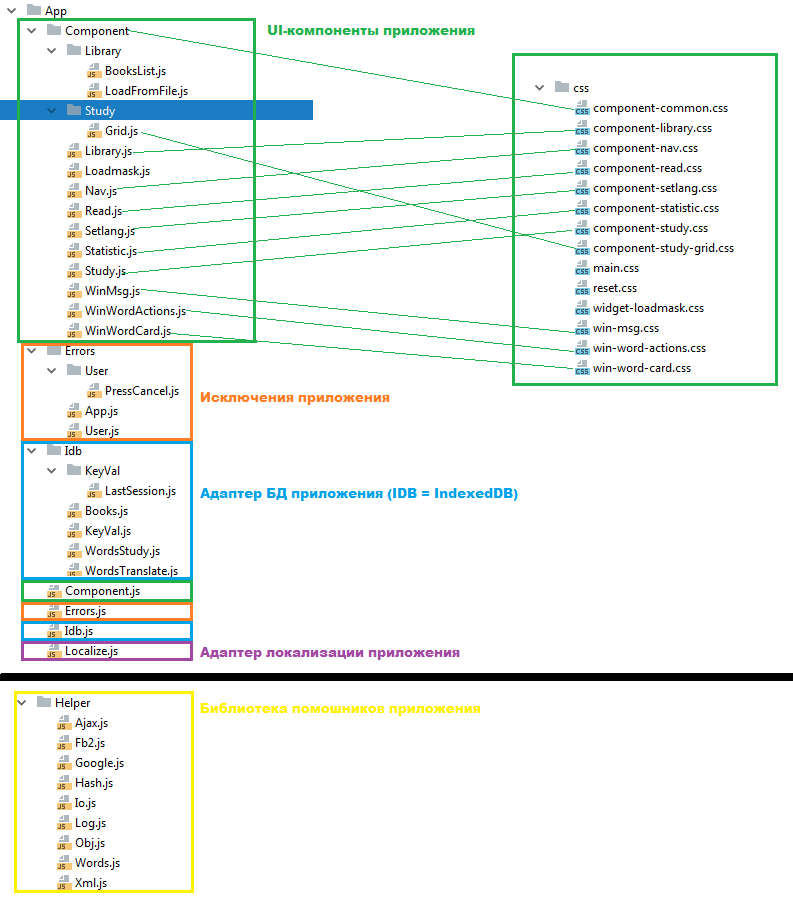
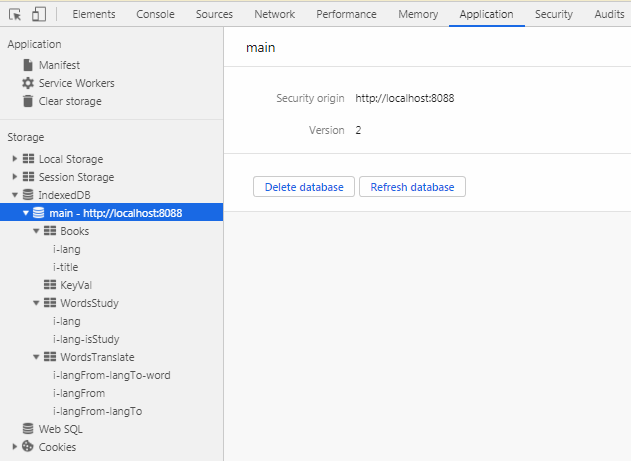


Рис.1 Структура програмного кода приложения.

Структура БД приложения.

* «main» – имя БД  
    
  Рис.2 БД IndexedDB в панели «Инструменты разработчика» Google Chrome
* “Books” – таблица для хранения книг, загруженных пользователем.  
  *// app/js/App/Idb/Books.js****App***.**Idb**.**Books** = {  
   */\*\*  
   \* Структура, описывающая хранимую книгу  
   \*/* **Struct**: {  
   */\*\*  
   \* Версия структуры на момент записи книги  
   \** ***@type*** *{Number}  
   \*/* **version**: 1,  
   */\*\*  
   \* Язык книги (Напр. "ENG" или "RUS")  
   \** ***@type*** *{String}  
   \*/* **lang**: **undefined**,  
   */\*\*  
   \* Заглавие книги на языке книги.  
   \*/* **title**: **undefined**,  
   */\*\*  
   \* BASE64-кодированное изображение титульной страницы.  
   \** ***@type*** *{String}  
   \*/* **image**: **undefined**,  
   */\*\*  
   \* MD5-хеш контента книги  
   \** ***@type*** *{String}  
   \*/* **hash**: **undefined**,  
   */\*\*  
   \* Контент FB2-книги  
   \** ***@type*** *{String}  
   \*/* **content**: **undefined** },

Первичный ключ: **keyPath**: **'hash'**

Индексы:  
«i-lang» - индекс по поля языку, на котором написана кгига. Позволяет выбирать только те книги, которые соответствуют изучаемому пользователем языку.

«i-title» - индекс по заглавию книги. Позволяет осуществлять поиск по заглавию.

* “KeyVal” – хеш-таблица общего назначения. Предназначена для хранения данных, которые по своей природе являются элементами «хеш-массива».  
  *// app/js/App/Idb/KeyVal/LastSession.js****App***.**Idb**.**KeyVal** = {  
   */\*\*  
   \* Ключи, которыми помечаются записи в таблице.  
   \*/* **KEYS** :{  
   */\*\*  
   \* Элемент хранит информацию о последней сессии пользователя  
   \*/* **LAST\_SESSION** : **'KEY\_LAST\_SESSION'** },

Первичным ключем является стандартный строковый ключ таблицы.

***App***.**Idb**.**KeyVal.KEYS** – перечисление допустимых ключей.

* + “LastSession” – единственная запись следующей структуры  
    *// app/js/App/Idb/KeyVal/LastSession.js****App***.**Idb**.**KeyVal**.**LastSession** = {  
     */\*\*  
     \* Структура, описывающая последнюю сессию пользователя.  
     \*/* **Struct**: {  
     */\*\*  
     \* Версия структуры на момент записи информации.  
     \** ***@type*** *{Number}  
     \*/* **version**: 1,  
     */\*\*  
     \* Язык пользовательского интерфейса (Напр. "ENG" или "RUS")  
     \** ***@type*** *{String}  
     \*/* **langGui**: **undefined**,  
     */\*\*  
     \* Изучаемый язык (Напр. "ENG" или "RUS")  
     \** ***@type*** *{String}  
     \*/* **langStudy**: **undefined**,  
     */\*\*  
     \* Идентификатор читаемой книги  
     \** ***@type*** *{String}  
     \*/* **bookHash**: **undefined**,  
     */\*\*  
     \* Позиция читаемой книги, на которой остановился пользователь  
     \** ***@type*** *{Number}  
     \*/* **bookPosition**: **undefined** },
* “WordsStudy” – таблица для хранения изучаемых и изученных слов.  
  *// app/js/App/Idb/WordsStudy.js  
  /\*\*  
   \*  
   \* Объект, предоставляющий интерфейс к таблице "Изучаемые слова".  
   \** ***@type*** *{Object}}  
   \*/****App***.**Idb**.**WordsStudy** = {  
   */\*\*  
   \* Структура, описывающая изучаемое слово.  
   \*/* **Struct**: {  
   */\*\*  
   \* Версия структуры на момент записи информации.  
   \** ***@type*** *{Number}  
   \*/* **version**: 1,  
   */\*\*  
   \* Язык слова (Напр. "ENG" или "RUS")  
   \** ***@type*** *{String}  
   \*/* **lang**: **undefined**,  
   */\*\*  
   \* Нормализованный текст слова (Напр. "dog", "яблоко")  
   \** ***@type*** *{String}  
   \*/* **word**: **undefined**,  
   */\*\*  
   \* Флаг: Слово для изучения? 0 - нет, 1 - Да.  
   \* 0 - слово уже изучено.  
   \* 1 - слово не изучено - прогнать через процесс изучения.  
   \** ***@type*** *{Number}  
   \*/* **isStudy**: **undefined**,  
   */\*\*  
   \* Флаг: Завершено ли изучение слова с родного языка на изучаемый?  
   \* 0 - Нет, 1 - Да  
   \** ***@type*** *{Number}  
   \*/* **isFinishedLang1ToLang2**: **undefined**,  
   */\*\*  
   \* Флаг: Завершено ли изучение слова с изучаемого языка на основной  
   \* 0 - Нет, 1 - Да  
   \** ***@type*** *{Number}  
   \*/* **isFinishedLang2ToLang1**: **undefined** }

Первичный ключ: композитный **keyPath**: [**'lang'**, **'word'**]

Индексы:

“i-lang” – позволяет искать слова по языку

“i-lang-isStudy” – позволяет искать изученные, либо неизученные слова для конкретного языка.

* “WordsTranslate” – таблица закешированных переводов слов.

*// app/js/App/Idb/WordsTranslate.js  
/\*\*  
 \* Объект, предоставляющий интерфейс к таблице "Перевод слов".  
 \** ***@type*** *{Object}}  
 \*/****App***.**Idb**.**WordsTranslate** = {  
 */\*\*  
 \* Структура, описывающая перевод слова.  
 \*/* **Struct**: {  
 */\*\*  
 \* Версия структуры на момент записи информации.  
 \** ***@type*** *{Number}  
 \*/* **version**: 1,  
 */\*\*  
 \* Для какого языка перевод слова (Напр. "ENG" или "RUS")  
 \** ***@type*** *{String}  
 \*/* **langFrom**: **undefined**,  
 */\*\*  
 \* На какой язык перевод слова (Напр. "ENG" или "RUS")  
 \** ***@type*** *{String}  
 \*/* **langTo**: **undefined**,  
 */\*\*  
 \* Нормализованный текст слова (Напр. "dog" или "яблоко")  
 \** ***@type*** *{String}  
 \*/* **word**: **undefined**,  
 */\*\*  
 \* Нормализованный перевод слова (Напр. "собака" или "apple")  
 \** ***@type*** *{String}  
 \*/* **translate**: **undefined**,  
 */\*\*  
 \* Частота употребления слова.  
 \* Дробное от 0 до 1. 1 = 100% означает, что слово имеет единствее значение.  
 \** ***@type*** *{Number}  
 \*/* **score** : **undefined**,  
 */\*\*  
 \* Время последней записи слова в таблицу (формат timestamp).  
 \** ***@type*** *{Number}  
 \*/* **insertAt**: 0  
 },

Первичный ключ: композитный **keyPath**: [**'langFrom'**, **'langTo'**, **'word'**]

Индексы:  
“i-langFrom” – Поиск всех слов по языку  
“i-langFrom-langTo” – Поиск всех переводов с основного языка на дополнительный

“i-langFrom-langTo-word” – Поиск перевода для конкретного слова