# ВВЕДЕНИЕ

Книгопечатная продукция утрачивает свои позиции на мировом рынке и к 2019 году число книг, проданных в электронном формате, сравняется с продажами бумажных экземпляров, а потом и неминуемо обгонит их. Это обусловлено широким распространением смартфонов, планшетов и стоимостью электронных книг, которые как правило дешевле печатных изданий, так как очевидно, не требуют материалов для производства каждого экземпляра.

В настоящий момент существует большое количество решений для чтения электронных книг, большая часть которых состоит в программном обеспечении, устанавливаемом на устройстве пользователя. Приложения на мобильных телефонах, планшетах и компьютерах имеют ряд преимуществ, таких как скорость работы, использовании библиотеки пользователя и широкие возможности для разработчиков.

Недостатком описанных выше приложений является привязка пользователя к одному устройству, на котором он обычно читает. При переходе, например, с мобильного телефона на компьютер или планшет, пользователь вынужден использовать другое ПО, загружать книгу и тратить время на поиск места, где чтение было прервано, что занимает не так мало времени.

Благодаря последним стандартам представления данных в сети интернет, таким как HTML5, CSS3, веб-приложения могут приблизиться по функциональности и скорости работы к нативным приложениям. Веб-приложение может решить проблему синхронизации при использовании разных устройств для чтения, предоставляя единый интерфейс для любых устройств, учитывая при этом особенности управления и разное разрешение экранов. Другим преимуществом веб-приложений можно считать возможность разработчика вносить изменение в приложение по функционалу, стабильности работы, внешнего вида и любые другие, без необходимости в обновлении ПО пользователем.

Целью дипломной работы является разработка веб-приложения для чтения электронных книг на любых современных устройствах имеющих доступ в интернет.

Для разработки программного обеспечения необходимо решить следующие задачи:

* провести обзор веб-приложений, которые предоставляют интерфейс для чтения книг;
* провести обзор популярных форматов электронных книг;
* провести обзор документации выбранных форматов;
* разработать структуру базы данных;
* разработать UI-компоненты веб-интерфейса;
* определиться с архитектурой клиентской части веб-приложения;
* написать бизнес логику клиентской части веб-приложения;
* определиться с архитектурой серверной части веб-приложения;
* написать бизнес логику серверной части веб-приложения.

# АНАЛИЗ И ВЫБОР СРЕДСТВ РЕАЛИЗАЦИЙ WEB-ПРИЛОЖЕНИЙ

# Анализ и описание предметной области

В связи с тем, что уже в текущее время люди имеют не одно, а несколько устройств, для чтения они могут выбирать устройство в зависимости от ситуации. Например, в общественном транспорте, где в час пик недостаточно пространства для планшета или электронной книги, самым распространённым устройством является смартфон. На работе или дома удобнее пользоваться компьютером или планшетом. При смене устройств возникает необходимость иметь копии книг на каждом устройстве. Данное приложение призвано решить проблему копирования книг с устройства на устройство, и поиска нужной страницы, что позволит избежать лишней траты времени.

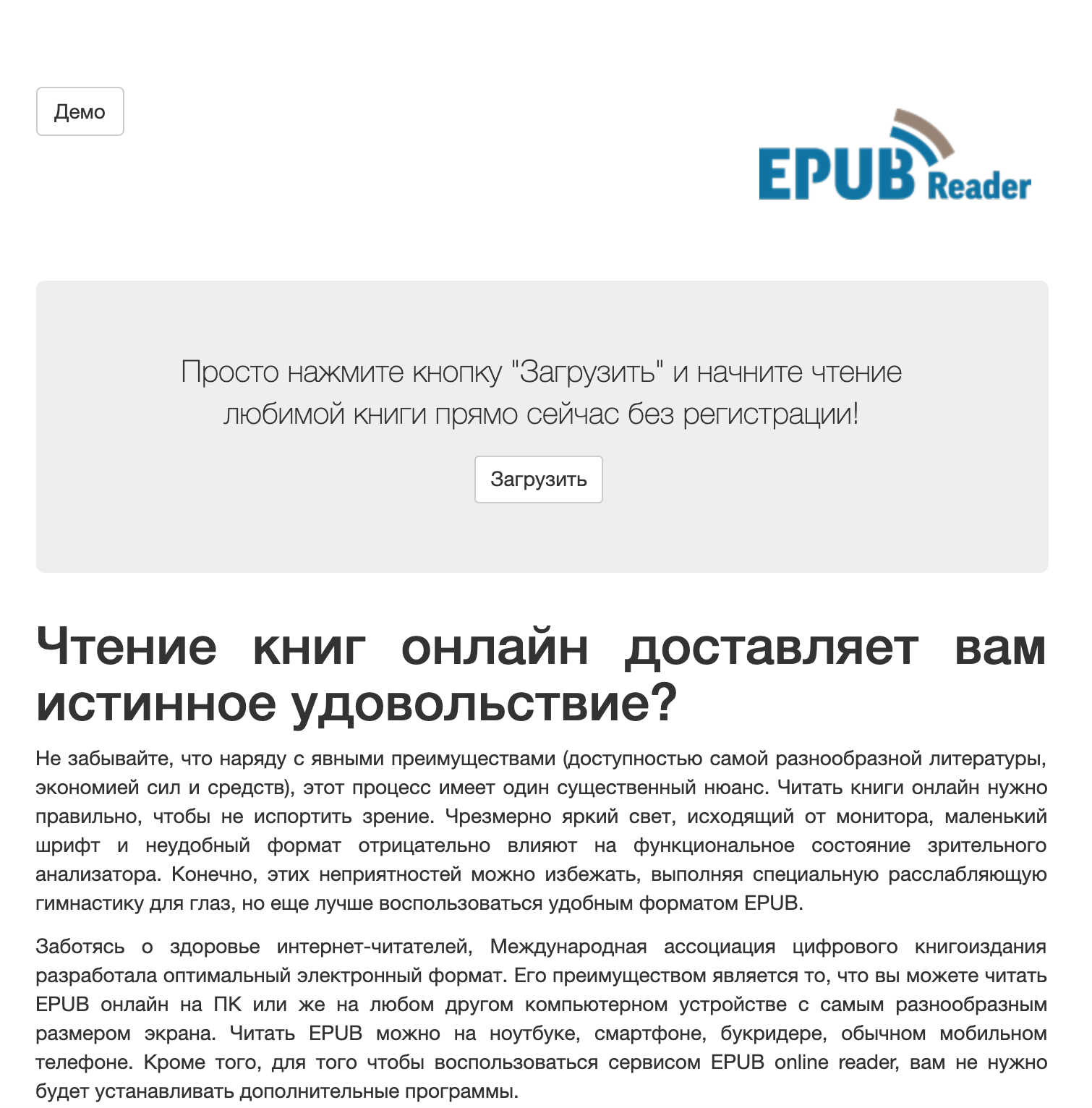
Предметной областью приложения, данной выпускной квалификационной работы является чтение книг на устройствах с выходом в интернет. Элементами предметной области являются книги которые обозначены следующими признаками – название, автор, описание, жанр.

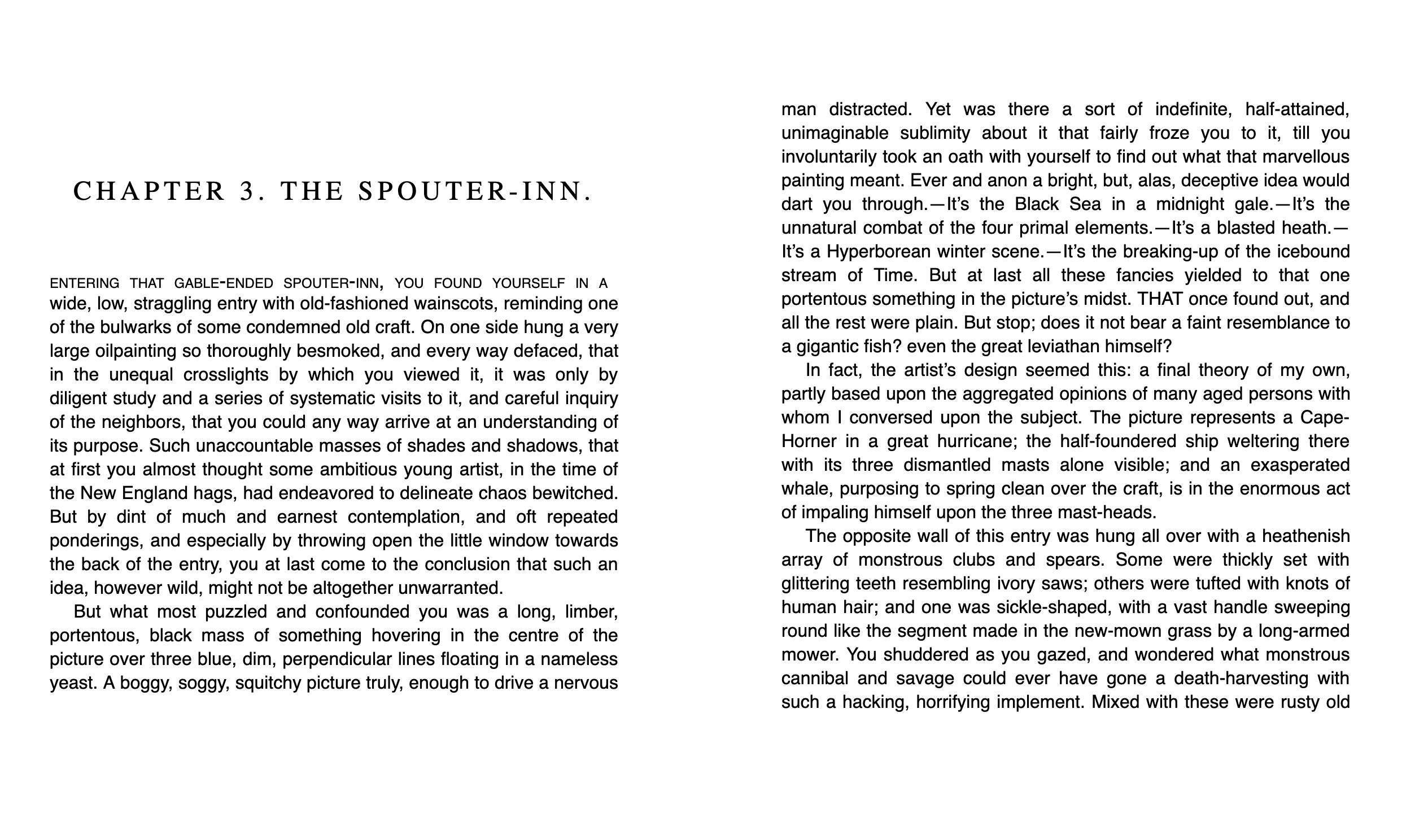
Процесс работы приложения можно представить следующим образом:

* в приложении изначально имеется каталог книг из которого пользователь может выбирать книги для чтения;
* часть книг в каталоге доступна бесплатно;
* часть книг в каталоге доступна только по подписке на приложение;
* пользователь может оформить подписку на месяц;
* пользователь может оформить подписку на 3 месяца;
* пользователь может оформить подписку на 6 месяцев;
* при выборе книги из каталога книг выбранная книга добавляется в библиотеку пользователя;
* пользователь имеет возможность сам загрузить книгу в формате epub в свою библиотеку со своего устройства;
* по мере чтения книги сохраняется количество прочитанных страниц, чтобы при повторном открытии книги пользователю не приходилось искать страницу на которой он закончил чтение в прошлый раз;
* к каждой книге можно оставить комментарий который могут увидеть другие пользователи;
* пользователь имеет возможность читать текст книги.

# Обзор аналогичных веб-приложений

EpubReader – приложение предоставляет возможность загрузить книгу с компьютера в формате epub и сразу начать ее читать. Данное приложение не сохраняет какую-либо информацию о книгах, прочитанном количестве страниц и пользователе, это означает, что после повторного открытия браузера вся информации будет удалена и придется заново загружать книгу и открывать нужную страницу, так что можно сказать, что это приложение «одноразовое».

Рисунок 1 – приложение EpubReader

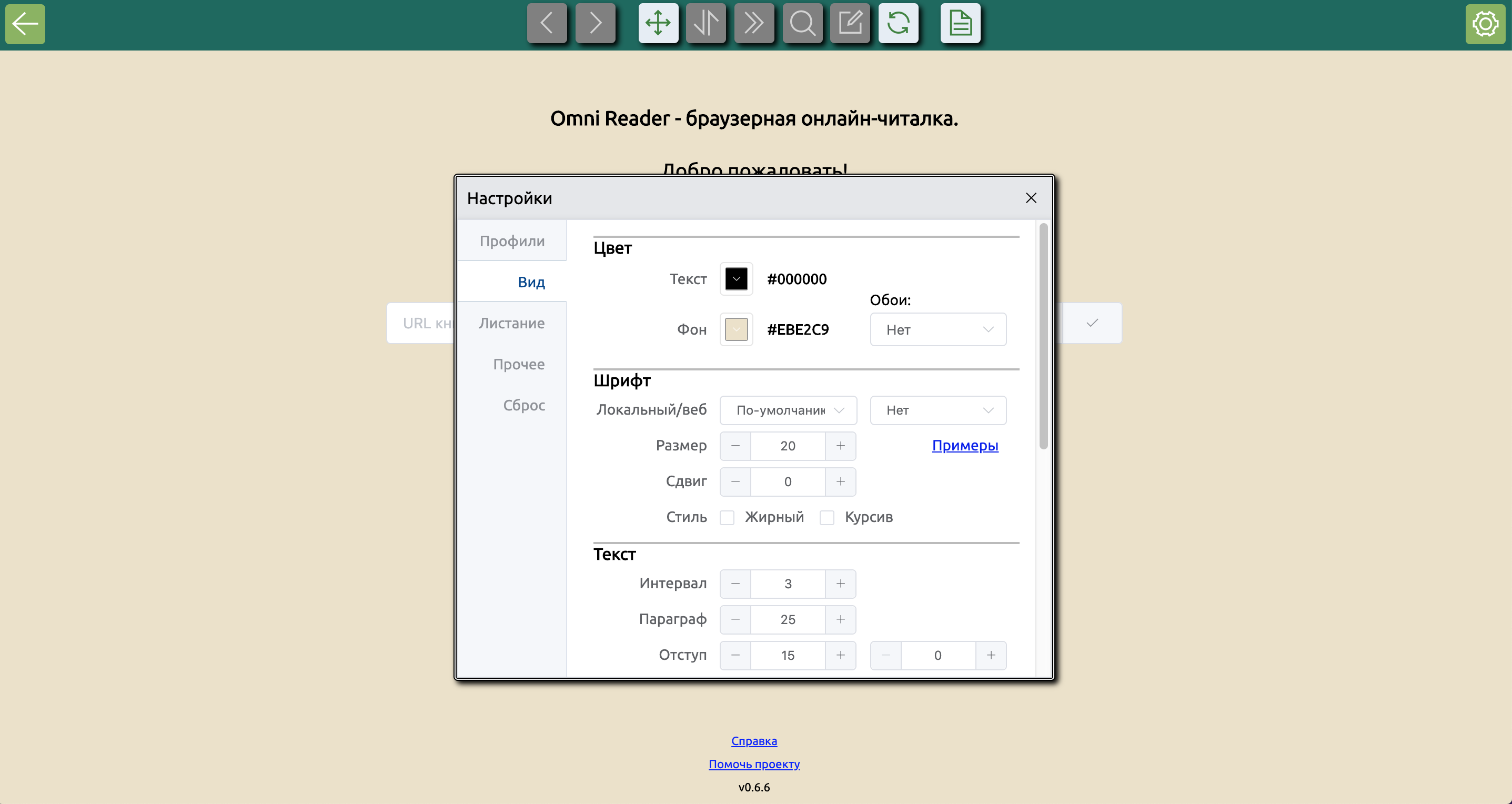
Рисунок 2 – читалка в приложении EpubReader

Omni Reader - браузерная онлайн-читалка. Приложение имеет 2 способа загрзкии книги:

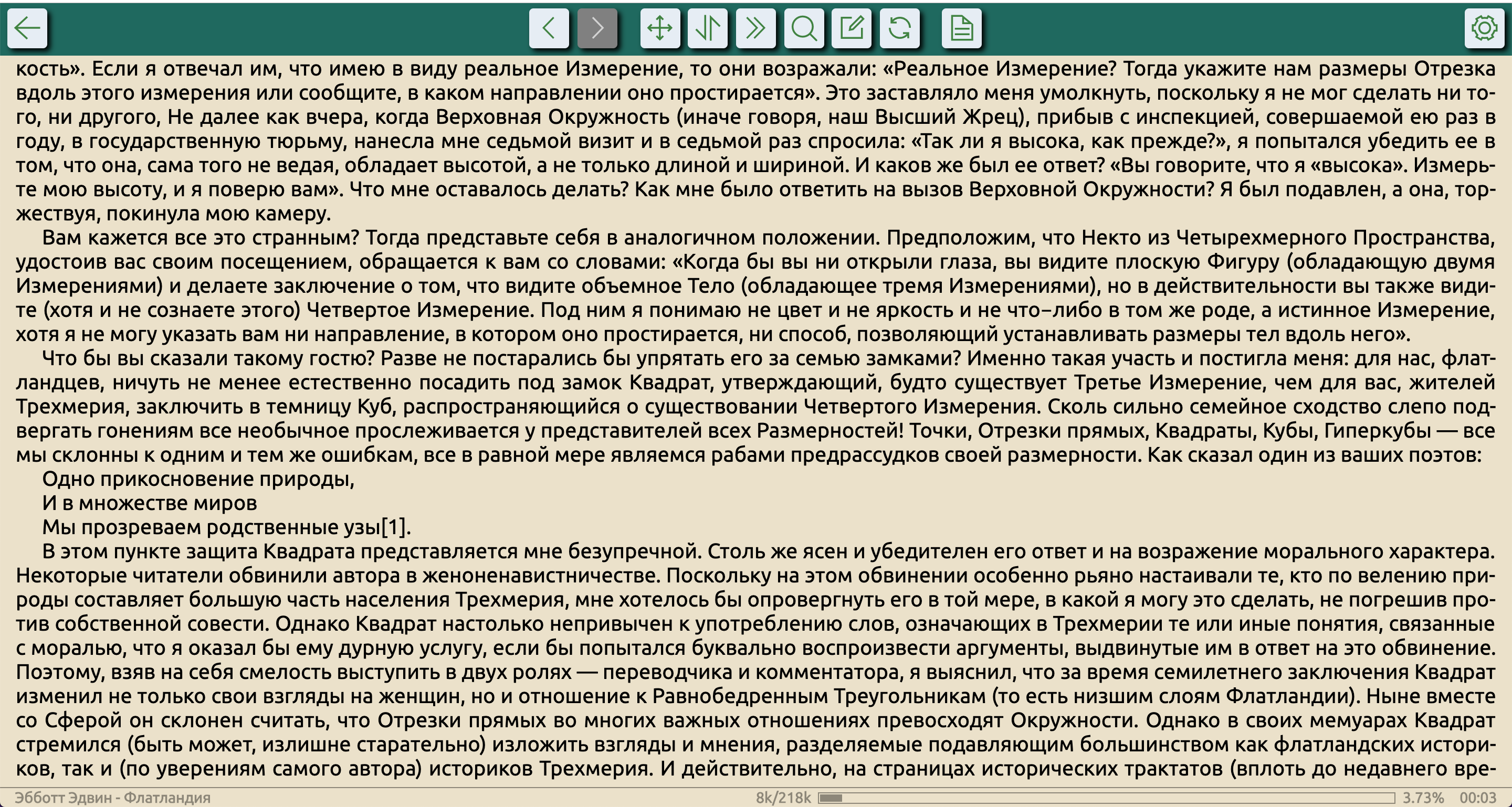
* вставить url адрес на книгу;
* загрузить книгу с жесткого диска;

**Рисунок 3 – приложение *Omni Reader*

Приложение имеет довольно большое количество настроеек, что очень хорошо, пользоваетель может настроить отображение текста так как ему будет удобнее его воспринимать.

**Рисунок 4 – настройки текста в приложениее *Omni Reader*

С помощью стрелок на клавиатуре компьютера либо на панели инструментов можно перемещаться между страницами книги, внизу можно отслеживать прогресс чтения загруженной книги. К сожалению на широком экране читать книги в этом приложение не достаточно удобно, а под мобильное разрешение экрана данное приложение не подстраивается, из-за этого чтение на мобильно устройстве будет некомфортным и затруднительным.

**Рисунок 5 – читалка в приложениее *Omni Reader*

В данном приложении тоже отсутствует библиотека пользователя, но сохраняется иформация о недавно открываемых книга. После отчистки кеша, эта информация удаляется.

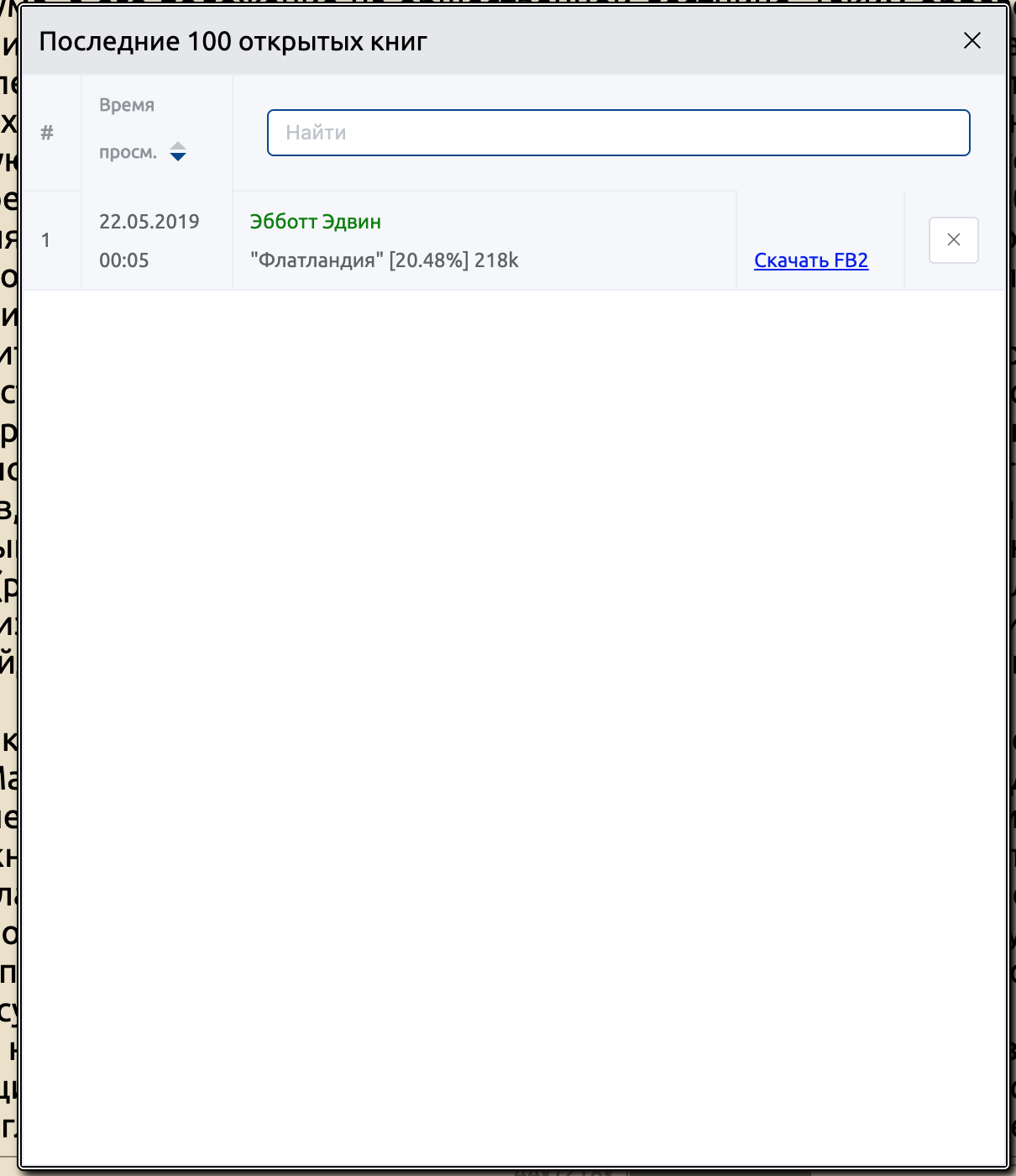


Рисунок 6 – последние 100 открытых книг в приложении *Omni Reader*

# Обзор технологий разработки веб-приложений

JavaScript — мультипарадигменный язык программирования. Поддерживает объектно-ориентированный, императивный и функциональный стили. Является реализацией языка ECMAScript (стандарт ECMA-262).

JavaScript обычно используется как встраиваемый язык для программного доступа к объектам приложений. Наиболее широкое применение находит в браузерах как язык сценариев для придания интерактивности веб-страницам.

Основные архитектурные черты: динамическая типизация, слабая типизация, автоматическое управление памятью, прототипное программирование, функции как объекты первого класса.

На JavaScript оказали влияние многие языки, при разработке была цель сделать язык похожим на Java, но при этом лёгким для использования непрограммистами. Языком JavaScript не владеет какая-либо компания или организация, что отличает его от ряда языков программирования, используемых в веб-разработке.

Название «JavaScript» является зарегистрированным товарным знаком компании Oracle Corporation в США.

PHP — скриптовый язык общего назначения, интенсивно применяемый для разработки веб-приложений. В настоящее время поддерживается подавляющим большинством хостинг-провайдеров и является одним из лидеров среди языков, применяющихся для создания динамических веб-сайтов.

PostgreSQL — свободная объектно-реляционная система управления базами данных (СУБД).

Существует в реализациях для множества UNIX-подобных платформ, включая AIX, различные BSD-системы, HP-UX, IRIX, Linux, macOS, Solaris/OpenSolaris, Tru64, QNX, а также для Microsoft Windows.

Express.js, или просто Express, фреймворк web-приложений для Node.js, реализованный как свободное и открытое программное обеспечение под лицензией MIT. Он спроектирован для создания веб-приложений и API. Де-факто является стандартным каркасом для Node.js. Автор фреймворка, TJ Holowaychuk, описывает его как созданный на основе написанного на языке Ruby каркаса Sinatra, подразумевая, что он минималистичен и включает большое число подключаемых плагинов. Express может являться backend'ом для программного стека MEAN, вместе с базой данных MongoDB и каркасом Vue.js, React или AngularJS для frontend'а.

AngularJS — JavaScript-фреймворк с открытым исходным кодом. Предназначен для разработки одностраничных приложений. Его цель — расширение браузерных приложений на основе MVC-шаблона, а также упрощение тестирования и разработки.

Фреймворк работает с HTML, содержащим дополнительные пользовательские атрибуты, которые описываются директивами, и связывает ввод или вывод области страницы с моделью, представляющей собой обычные переменные JavaScript. Значения этих переменных задаются вручную или извлекаются из статических или динамических JSON-данных.

React — [JavaScript](https://ru.wikipedia.org/wiki/JavaScript)-[библиотека](https://ru.wikipedia.org/wiki/%D0%91%D0%B8%D0%B1%D0%BB%D0%B8%D0%BE%D1%82%D0%B5%D0%BA%D0%B0_JavaScript) с [открытым исходным кодом](https://ru.wikipedia.org/wiki/%D0%9E%D1%82%D0%BA%D1%80%D1%8B%D1%82%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5) для разработки [пользовательских интерфейсов](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81_%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8F).

React разрабатывается и поддерживается [Facebook](https://ru.wikipedia.org/wiki/Facebook), [Instagram](https://ru.wikipedia.org/wiki/Instagram) и сообществом отдельных разработчиков и корпораций.

React может использоваться для разработки [одностраничных](https://ru.wikipedia.org/wiki/%D0%9E%D0%B4%D0%BD%D0%BE%D1%81%D1%82%D1%80%D0%B0%D0%BD%D0%B8%D1%87%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5) и мобильных приложений. Его цель — предоставить высокую скорость, простоту и масштабируемость. В качестве библиотеки для разработки пользовательских интерфейсов React часто используется с другими библиотеками, такими как [Redux](https://ru.wikipedia.org/wiki/Redux).

Vue.js — [JavaScript](https://ru.wikipedia.org/wiki/JavaScript)-[фреймворк](https://ru.wikipedia.org/wiki/%D0%A4%D1%80%D0%B5%D0%B9%D0%BC%D0%B2%D0%BE%D1%80%D0%BA) с [открытым исходным кодом](https://ru.wikipedia.org/wiki/%D0%9E%D1%82%D0%BA%D1%80%D1%8B%D1%82%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5) для создания [пользовательских интерфейсов](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81_%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8F). Легко интегрируется в проекты с использованием других [JavaScript-библиотек](https://ru.wikipedia.org/wiki/%D0%91%D0%B8%D0%B1%D0%BB%D0%B8%D0%BE%D1%82%D0%B5%D0%BA%D0%B0_JavaScript). Может функционировать как [веб-фреймворк](https://ru.wikipedia.org/wiki/%D0%9A%D0%B0%D1%80%D0%BA%D0%B0%D1%81_%D0%B2%D0%B5%D0%B1-%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B9) для разработки [одностраничных приложений](https://ru.wikipedia.org/wiki/%D0%9E%D0%B4%D0%BD%D0%BE%D1%81%D1%82%D1%80%D0%B0%D0%BD%D0%B8%D1%87%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5) в [реактивном](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%B0%D0%BA%D1%82%D0%B8%D0%B2%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) стиле.

[Redux](http://rackt.github.io/redux/) — это одно из наиболее интересных явлений мира JavaScript. Он выделяется из сотни библиотек и фреймворков тем, что грамотно решает множество разных вопросов путем введения простой и предсказуемой модели состояний, уклоне на функциональное программирование и неизменяемые данные, предоставления компактного API. Что ещё нужно для счастья? Redux — библиотека очень маленькая, и выучить её API не сложно.

NodeJS — программная платформа, основанная на движке V8, превращая JS из узкоспециализированного языка в язык общего назначения. NodeJS Добавляется возможность JS взаимодействовать с устройствами ввода-вывода через свой API (написанный на C++), подключать другие внешние библиотеки, написанные на разных языка, обеспечивая вызов к ним из JS-кода. NodeJS применяется преимущественно на сервере, выполняя роль веб-сервера , но есть возможность разрабатывать на NodeJS и десктопные оконные приложения и даже программировать микроконтроллеры. В основе NodeJS лежит событийно-ориентированное и асинхронное программирование с неблокирующим вводом/выводом.

# Подходы к написанию веб-приложений

На сегодняшний день существует два принципиальных подхода к созданию веб-приложений: традиционные веб-приложения, большая часть логики которых выполняется на сервере, а также одностраничные приложения, логика пользовательского интерфейса которых выполняется преимущественно в веб-браузере, а взаимодействие с веб-сервером осуществляется главным образом через веб-API. Также возможен гибридный подход, при котором в простейшем случае в рамках крупного традиционного веб-приложения размещаются одно или несколько полнофункциональных подчиненных приложений, построенных на основе одностраничной модели.

Традиционные веб-приложения обычно используются в следующих случаях:

* На стороне клиента к приложению применяются минимальные требования, например, используются только функции чтения.
* Приложение должно работать в браузерах без поддержки JavaScript.
* Ваша команда не знакома с принципами разработки на JavaScript или TypeScript.
* Одностраничные приложения рекомендуется применять в следующих случаях:
* В приложении требуется полнофункциональный пользовательский интерфейс.
* Ваша команда знакома с принципами разработки на JavaScript или TypeScript.
* В вашем приложении должен предоставляться API для других внутренних или общедоступных клиентов.

Кроме того, для работы с платформой одностраничных приложений требуется гораздо больший опыт в построении архитектуры и обеспечении безопасности. Они характеризуются большей частотой обновления и появления новых платформ по сравнению с традиционными веб-приложениями. Кроме того, по сравнению с традиционными веб-приложениями при работе с одностраничными приложениями усложняются настройка процессов автоматизированного построения и развертывания, а также использование таких вариантов развертывания, как контейнеры.

Эти моменты необходимо учитывать, оценивая улучшения модели взаимодействия с пользователем, которые могут дать одностраничные приложения.

# Анализ инструментов разработки веб-приложений

Webpack *–* это инструмент, позволяющий скомпилировать, например, JavaScript модули в единый JS-файл. Webpack также известен как сборщик модулей.

При большом количестве файлов он создает один объемный файл (или несколько файлов) для запуска вашего приложения.

Он также способен выполнять множество иных операций:

помогает собрать воедино ваши ресурсы

* следит за изменениями и повторно выполняет задачи;
* может выполнить транспиляцию JavaScript следующего поколения до более старого стандарта JavaScript (ES5) с помощью Babel, что позволит использовать новейшие функции JavaScript, не беспокоясь о том, поддерживает их браузер или нет;
* может выполнить транспиляцию CoffeeScript в JavaScript;
* может конвертировать встроенные изображения в data:URI;
* позволяет использовать require() для CSS файлов;
* может запустить webpack-dev-server (в нём встроен локальный сервер и livereload (“живая перезагрузка браузера”));
* может работать с Hot Module Replacement (замена горячего модуля);
* может разделить выходной файл (output file) на несколько файлов, чтобы избежать медленной загрузки страницы из-за большого размера JS-файла;
* может выполнить Tree Shaking;

Webpack не ограничивается одним лишь фронтендом, его также успешно применяют в бэкенд разработке на Node.js.

[Gulp](http://gulpjs.com/)*–* инструментарий, облегчающий жизнь веб-разработчика, с его помощью можно задавать различные задачи. Часто используется для таких front-end задач как: поднятие сервера, автоматическое обновление страницы браузера в любой момент сохранения файла, использование препроцессоров как Sass или LESS, оптимизация CSS, JavaScript и изображений.

Docker — программное обеспечение для автоматизации развёртывания и управления приложениями в средах с поддержкой [контейнеризации](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BD%D1%82%D0%B5%D0%B9%D0%BD%D0%B5%D1%80%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F). Позволяет «упаковать» приложение со всем его [окружением](https://ru.wikipedia.org/w/index.php?title=%D0%9E%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BE%D0%BA%D1%80%D1%83%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5&action=edit&redlink=1) и зависимостями в контейнер, который может быть перенесён на любую [Linux](https://ru.wikipedia.org/wiki/Linux)-систему с поддержкой [cgroups](https://ru.wikipedia.org/wiki/Cgroups) в [ядре](https://ru.wikipedia.org/wiki/%D0%AF%D0%B4%D1%80%D0%BE_Linux), а также предоставляет среду по управлению контейнерами. Изначально использовал возможности [LXC](https://ru.wikipedia.org/wiki/LXC), с 2015 года применял собственную библиотеку, абстрагирующую виртуализационные возможности ядра Linux — libcontainer. С появлением ​Open Container Initiative начался переход от монолитной к модульной архитектуре.

Разрабатывается и поддерживается одноимённой компанией-[стартапом](https://ru.wikipedia.org/wiki/%D0%A1%D1%82%D0%B0%D1%80%D1%82%D0%B0%D0%BF), распространяется в двух редакциях — общественной (Community Edition) по [лицензии Apache 2.0](https://ru.wikipedia.org/wiki/%D0%9B%D0%B8%D1%86%D0%B5%D0%BD%D0%B7%D0%B8%D1%8F_Apache) и для организаций (Enterprise Edition) по [проприетарной](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%BF%D1%80%D0%B8%D0%B5%D1%82%D0%B0%D1%80%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5)лицензии. Написан на языке [Go](https://ru.wikipedia.org/wiki/Go_(%D1%8F%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F)).

Visual Studio Code  —  [редактор исходного кода,](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%B4%D0%B0%D0%BA%D1%82%D0%BE%D1%80_%D0%B8%D1%81%D1%85%D0%BE%D0%B4%D0%BD%D0%BE%D0%B3%D0%BE_%D0%BA%D0%BE%D0%B4%D0%B0) разработанный [Microsoft](https://ru.wikipedia.org/wiki/Microsoft) для [Windows](https://ru.wikipedia.org/wiki/Windows), [Linux](https://ru.wikipedia.org/wiki/Linux) и [macOS](https://ru.wikipedia.org/wiki/MacOS). Позиционируется как «лёгкий» редактор кода для [кроссплатформенной](https://ru.wikipedia.org/wiki/%D0%9A%D1%80%D0%BE%D1%81%D1%81%D0%BF%D0%BB%D0%B0%D1%82%D1%84%D0%BE%D1%80%D0%BC%D0%B5%D0%BD%D0%BD%D0%BE%D1%81%D1%82%D1%8C) разработки [веб-](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5) и [облачных](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D0%BB%D0%B0%D1%87%D0%BD%D1%8B%D0%B5_%D0%B2%D1%8B%D1%87%D0%B8%D1%81%D0%BB%D0%B5%D0%BD%D0%B8%D1%8F) приложений. Включает в себя [отладчик](https://ru.wikipedia.org/wiki/%D0%9E%D1%82%D0%BB%D0%B0%D0%B4%D1%87%D0%B8%D0%BA) инструменты для работы с [Git](https://ru.wikipedia.org/wiki/Git), [подсветку синтаксиса](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%B4%D1%81%D0%B2%D0%B5%D1%82%D0%BA%D0%B0_%D1%81%D0%B8%D0%BD%D1%82%D0%B0%D0%BA%D1%81%D0%B8%D1%81%D0%B0),  [IntelliSense](https://ru.wikipedia.org/wiki/IntelliSense) и средства для [рефакторинга](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D1%84%D0%B0%D0%BA%D1%82%D0%BE%D1%80%D0%B8%D0%BD%D0%B3). Имеет широкие возможности для [кастомизации](https://ru.wikipedia.org/wiki/%D0%9A%D0%B0%D1%81%D1%82%D0%BE%D0%BC%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F): пользовательские темы, [сочетания клавиш](https://ru.wikipedia.org/wiki/%D0%A1%D0%BE%D1%87%D0%B5%D1%82%D0%B0%D0%BD%D0%B8%D0%B5_%D0%BA%D0%BB%D0%B0%D0%B2%D0%B8%D1%88) и [файлы конфигурации](https://ru.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB_%D0%BA%D0%BE%D0%BD%D1%84%D0%B8%D0%B3%D1%83%D1%80%D0%B0%D1%86%D0%B8%D0%B8). Распространяется бесплатно, разрабатывается как программное обеспечение с [открытым исходным кодом](https://ru.wikipedia.org/wiki/%D0%9E%D1%82%D0%BA%D1%80%D1%8B%D1%82%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5), но готовые сборки распространяются под [проприетарной лицензией](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%BF%D1%80%D0%B8%D0%B5%D1%82%D0%B0%D1%80%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5).

Visual Studio Code основан на [Electron](https://ru.wikipedia.org/wiki/Electron) — [фреймворк](https://ru.wikipedia.org/wiki/%D0%A4%D1%80%D0%B5%D0%B9%D0%BC%D0%B2%D0%BE%D1%80%D0%BA), позволяющий с использованием [Node.js](https://ru.wikipedia.org/wiki/Node.js) разрабатывать настольные приложения, которые работают на [движке Blink](https://ru.wikipedia.org/wiki/Blink_(%D0%B4%D0%B2%D0%B8%D0%B6%D0%BE%D0%BA)). Несмотря на то, что редактор основан на [Electron](https://ru.wikipedia.org/wiki/Electron), он не использует редактор [Atom](https://ru.wikipedia.org/wiki/Atom_(%D1%82%D0%B5%D0%BA%D1%81%D1%82%D0%BE%D0%B2%D1%8B%D0%B9_%D1%80%D0%B5%D0%B4%D0%B0%D0%BA%D1%82%D0%BE%D1%80)). Вместо него реализуется веб-редактор Monaco разработанный для Visual Studio Online.

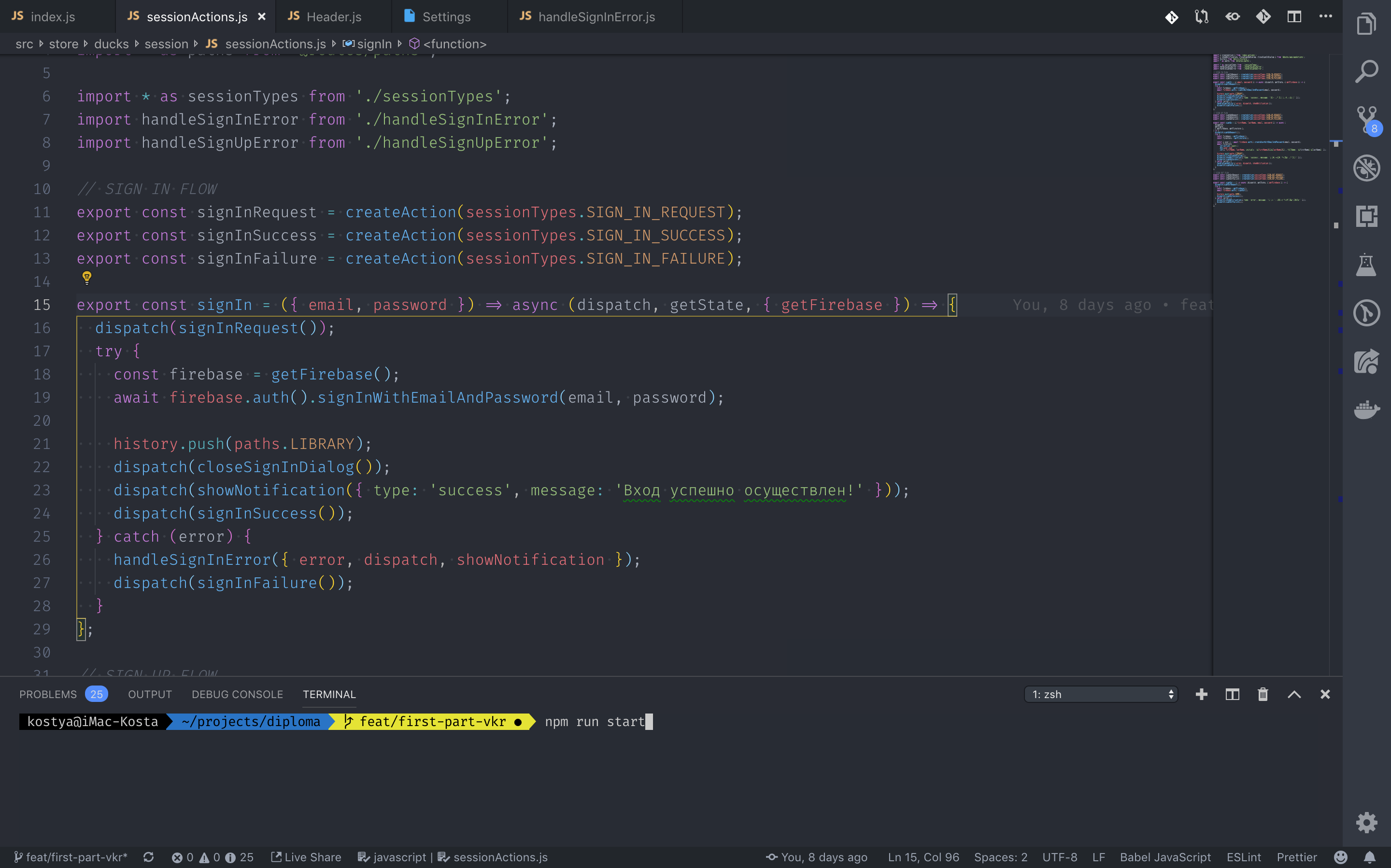


Рисунок 7 – редактор Visual Studio Code

ESLint — это инструмент, который позволяет проводить анализ качества вашего кода, написанного на любом выбранном стандарте JavaScript. Он приводит код к единому стилю, помогает избежать ошибок, умеет автоматически исправлять многие из найденных проблем и отлично интегрируется со многими инструментами разработки, такими как редактор Visual Studio Code и Web Storm IDE от JetBrains. ESLint не обязывает к одному какому-то конкретному стилю. Наоборот можно выбрать что-то из лучших практик и доработать по усмотрению!



Рисунок 8 – отображение стилистических ошибок с помощью ESLint

Emmet (ранее Zen Coding) — набор плагинов для текстовых редакторов, которые в некоторой степени ускоряют написание кода HTML, XML, XSL, а также кода на некоторых других языках. Проект был начат Вадимом Макеевым в 2008 году и активно разрабатывается Сергеем Чикуенком начиная с 2009 года, а также сообществом пользователей Zen Coding на основе идеи Zen Coding 2.0. C 2015 года за опыт пользователей в проекте отвечает Микаэл Гелецян.

Инструменты Zen Coding были вложены в популярные текстовые редакторы, а некоторые плагины, разработанные командой Zen Coding и другими, сделаны независимыми друг от друга. Zen Coding написан на JavaScript и поэтому хорошо переносим между платформами.

Возможность использования Zen Coding есть во многих интегрированных средах разработки. В некоторых из них, например, NetBeans, эта возможность реализована в виде плагина, разработанного независимым от Sun Microsystems и Oracle — производителя NetBeans — разработчиком, в некоторых — встроена в саму среду, например, в IntelliJ IDEA, PhpStorm и WebStorm от JetBrains.

Zen Coding является программным обеспечением с открытым исходным кодом и распространяется под лицензией MIT.

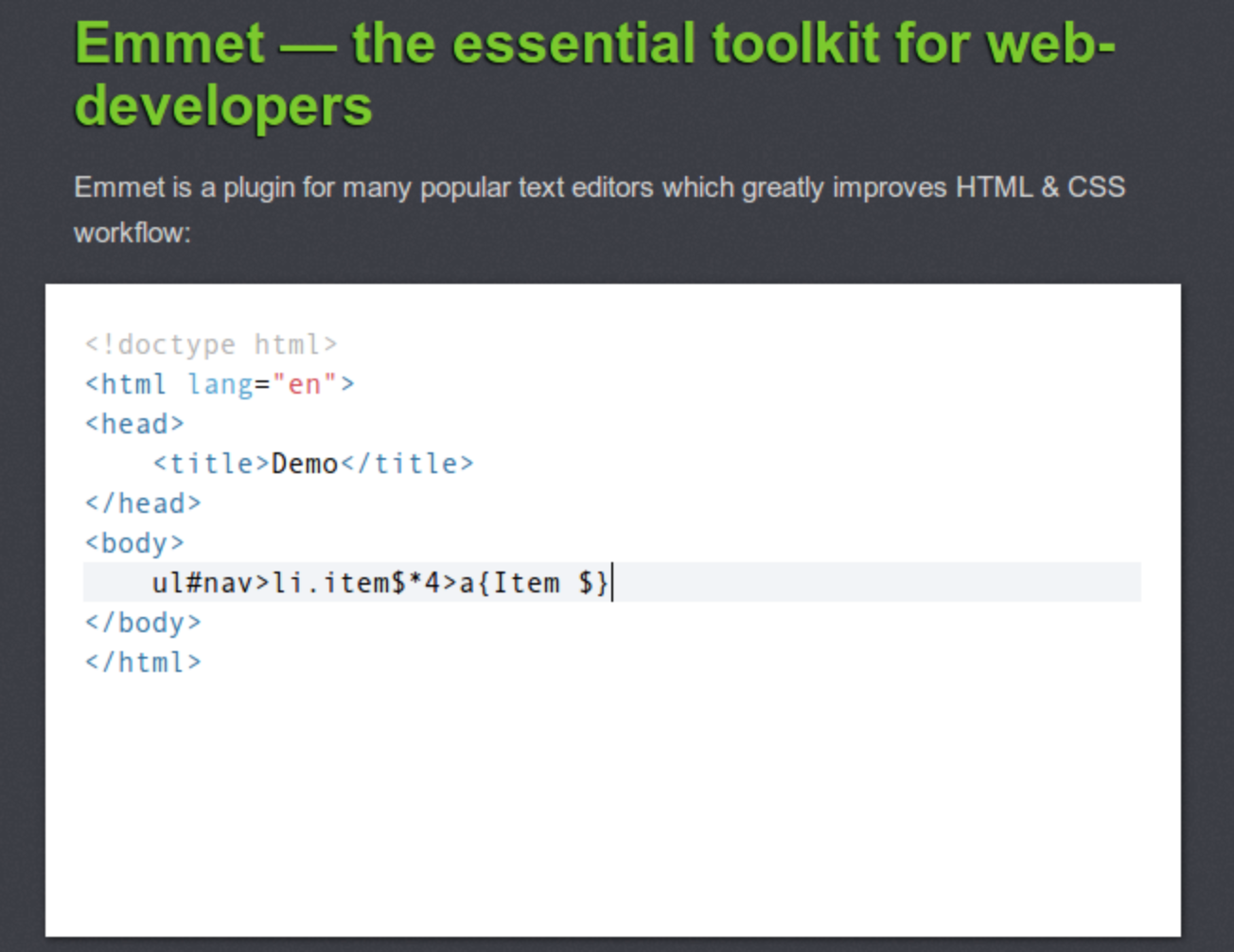


Рисунок 9 – плагин Emmet

Koala - кросс-платформенное приложение, которое автоматически компилирует файлы таких css препроцессоров как Less SASS и другие.

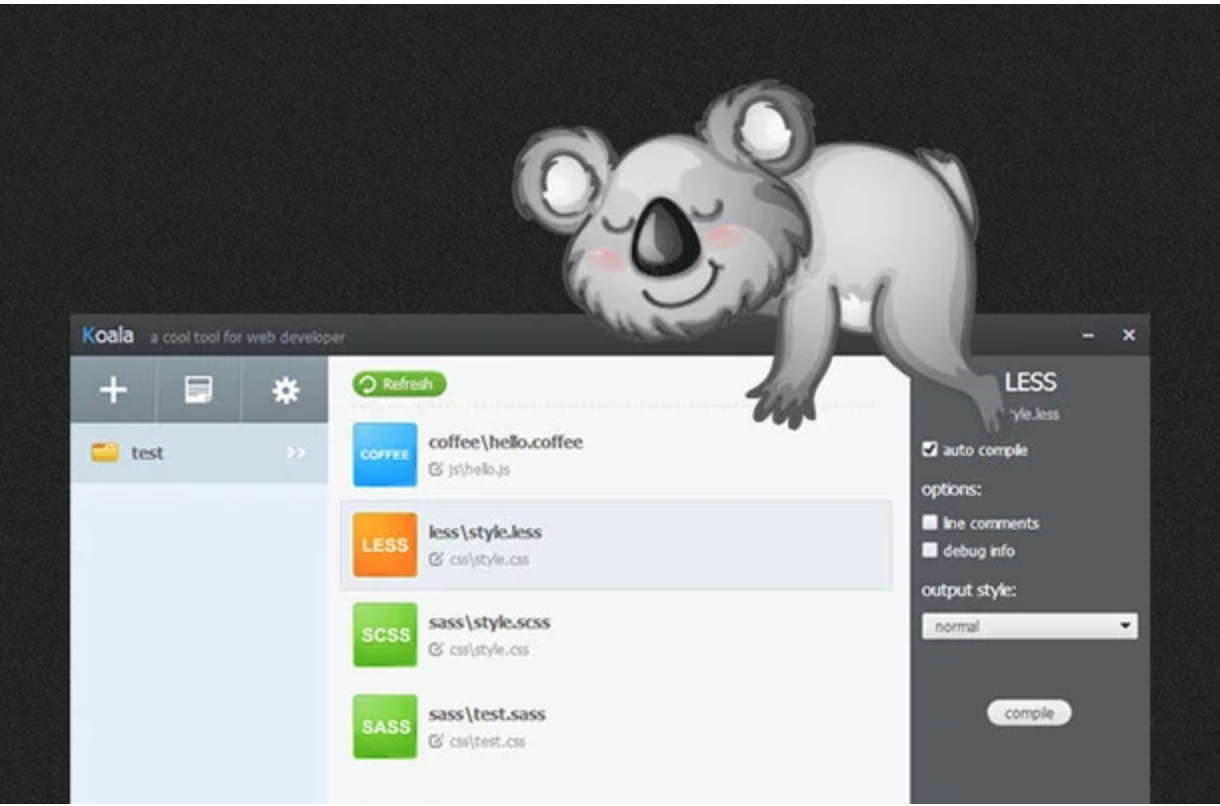


Рисунок 10 – Koala

CSS препроцессор - это программа, которая имеет свой собственный синтаксис (syntax), но может сгенерировать из него CSS код . Существует множество препроцессоров. Большинство из них расширяет функционал чистого CSS, добавляя такие опции как: примеси, вложенные правила, селекторы наследования и др. Эти особенности облегчают работу с CSS: упрощают чтение кода и его дальнейшую поддержку.

JetBrains WebStorm — интегрированная среда разработки на JavaScript, CSS & HTML от компании JetBrains, разработанная на основе платформы IntelliJ IDEA.

WebStorm обеспечивает автодополнение, анализ кода на лету, навигацию по коду, рефакторинг, отладку, и интеграцию с системами управления версиями. Важным преимуществом интегрированной среды разработки WebStorm является работа с проектами[1] (в том числе, рефакторинг кода JavaScript, находящегося в разных файлах и папках проекта, а также вложенного в HTML). Поддерживается множественная вложенность (когда в документ на HTML вложен скрипт на Javascript, в который вложен другой код HTML, внутри которого вложен Javascript) — то есть в таких конструкциях поддерживается корректный рефакторинг.

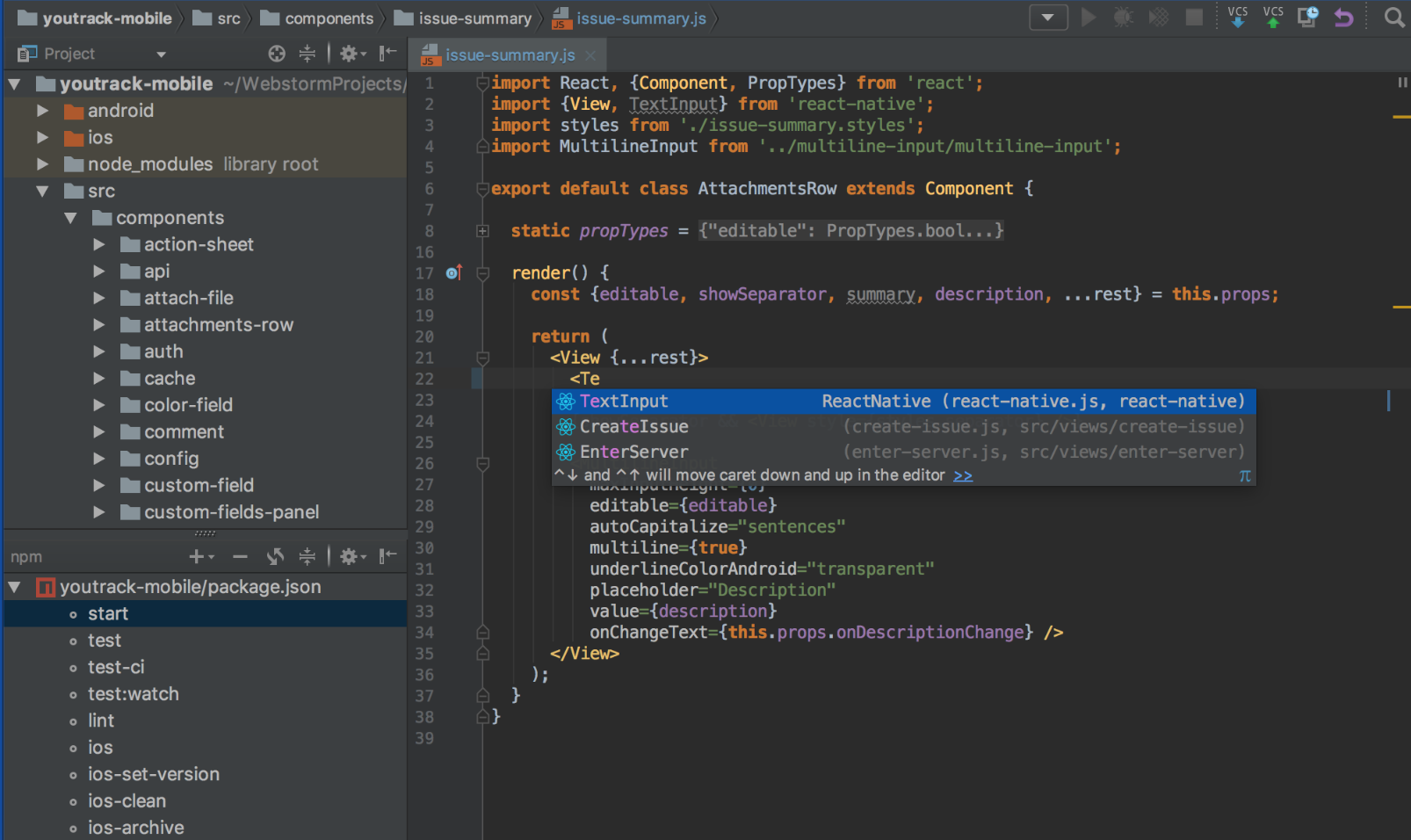


Рисунок 11 – JetBrains WebStorm

Atom — бесплатный текстовый редактор с открытым исходным кодом для Linux, macOS, Windows с поддержкой плагинов, написанных на Node.js, и встраиваемых под управлением Git. Большинство плагинов имеют статус свободного программного обеспечения, разрабатываются и поддерживаются сообществом.

Atom основан на Electron (ранее известный как Atom Shell) — фреймворке кросс-платформенной разработки с использованием Chromium и io.js. Редактор написан на CoffeeScript и LESS. Версия 1.0 была выпущена 25 июня 2015 г.

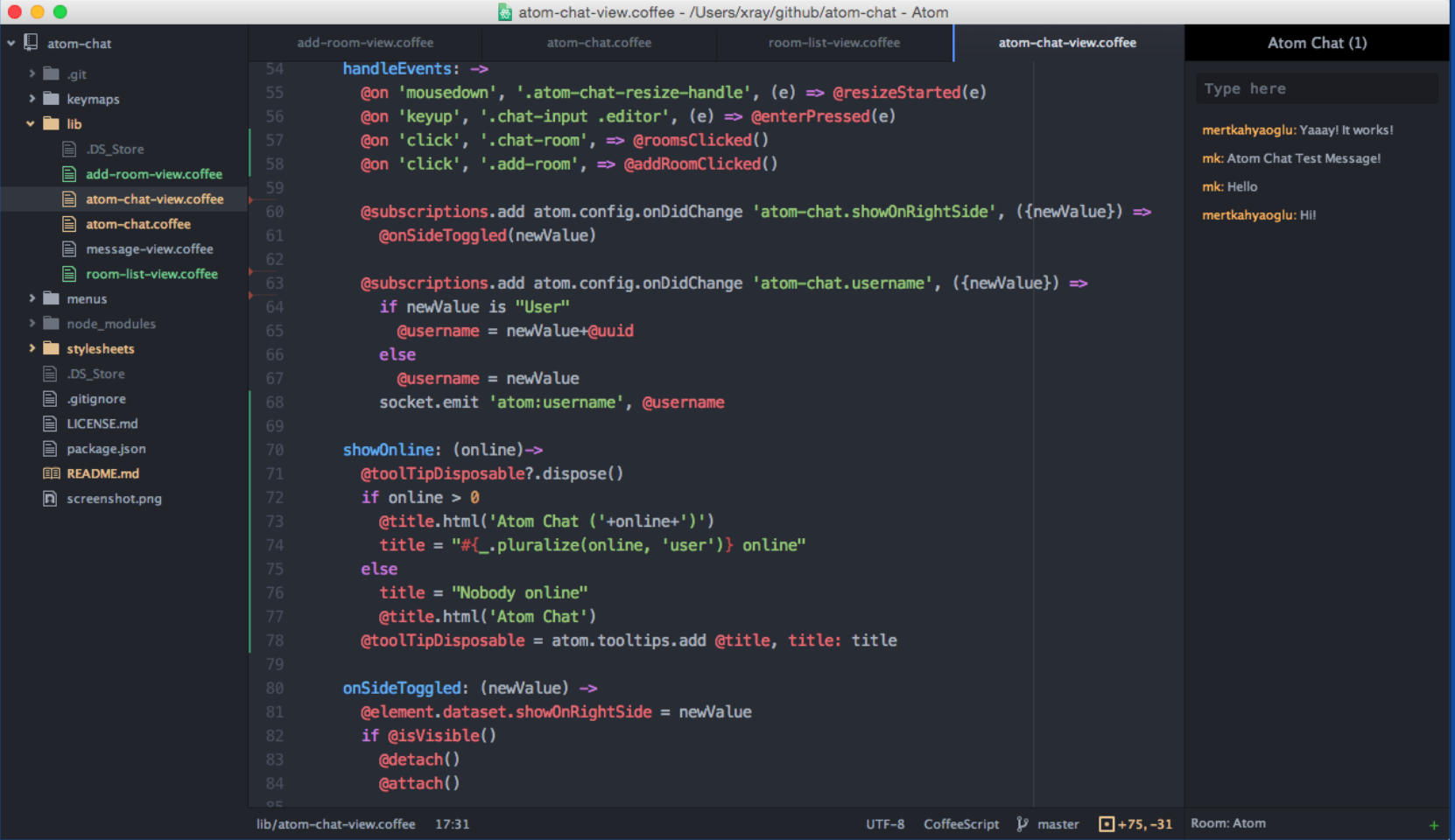


Рисунок 12 – текстовый редактор Atom