**Министерство общего и профессионального образования Ростовской области**

**ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**

**РОСТОВСКОЙ ОБЛАСТИ**

**«Ростовский-на-Дону колледж связи и информатики»**

**КУРСОВОЙ ПРОЕКТ**

по дисциплине

**ОП.10 «ПРОГРАММИРОВАНИЕ НА JAVASCRIPT»**

Тема «Создание тренажера скоропечатания с помощью библиотеки jQuery и localStorage»

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

# РКСИ.К18.09.02.03.ОП.10.18/2284.00ПЗ

(шифр)

Специальность 09.02.03 «Программирование в компьютерных системах»

Выполнил студент \_\_\_\_\_\_\_\_\_\_\_\_ Аброськин Д.А. Группа ПОКС-31

подпись Ф.И.О.

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_ Колесников П.С.

подпись Ф.И.О.

Проект защищен с оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ "\_\_\_"\_\_\_\_\_\_\_\_2018г.

##### 2018 г.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Содержание  Введение…………………………………………...….………………………... 3  1 Проектирование………………………………………..………….…..….….. 4  1.1 Use case диаграмма………………………………………...…………… 4  1.2 Прототип интерфейса (Mockup)………………………………………. 5  1.3 Разработка архитектуры ПП…………………………………………… 7  2 Разработка программного продукта……………………………...…............ 9  3 Документация…………………………………………………………………10  Заключение……………………………………………………………..…….....12 | | | | | | | | | | |
|  |  |  |  |  | РКСИ.К18.09.02.03.18/2284 | | | | | |
|  |  |  |  |  |
| *изм* | *лист* | *№ докум.* | *Подпись* | *Дата* |
| *Студент.* | | *Аброськин* |  |  | Создание тренажера скоропечатания с помощью библиотеки jQuery и localStorage  Отчет. | *Лит* | | | *Лист* | *Листов* |
| *Консульт.* | |  |  |  |  |  |  | *2* | *12* |
| *Руковод.* | | *Колесников* |  |  | ПОКС-31 | | | | |
| *Н. контр* | |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Введение**  В наш век цифровых технологий взаимодействовать с электронно-вычислительными машинами, в частности компьютерами приходиться очень часто. В связи с этим навыки, которые облегчают процесс этого взаимодействия и делают его более комфортным, имеют огромное значение. И тема моей курсовой очень сильно пересекается с развитием этих навыков.  Мне нужно было разработать тренажер скоропечатания, используя библиотеку jQuery и технологию localStorage.   Прежде всего я сделал лицо сайта, которое при каждом первом запуске отражало самое необходимое, а именно логотип сайта, который выделяет его среди остальных, поле для выбора режима «тренировки», что очень важно, ведь каждый человек имеет разный уровень владения данным навыком и интересы. Также на начальном экране находится кнопка «Рекорды», при нажатии на которую можно получить список лучшей скорости в каждом режиме, который храниться в локальном хранилище браузера, и изменяется либо при установлении нового рекорда в данном режиме, либо при нажатии на кнопку «Очистить», которая находится под списком рекордов и удаляющая все имеющиеся в локальном хранилище данные. И последним элементом начального экрана являются поля, в пределах которых и появится текст при выборе режима пользователем. Сделано это для того, чтобы пользователь знал, где ему ожидать появление текста и не был дезориентирован при его появлении.  После выбора режима уже появляется связанный с вводом интерфейс, а именно элемент состояния, помогающий пользователю понять что делать, который при первом появлении предлагает начать печатать, во время печати превращается в кнопку паузы а также указывает, какая клавиша также отвечает за эту функцию и опять же предлагает пользователю продолжить печать при установки на паузу.  Также имеется элемент, отображающий текущую скорость печати пользователя, который динамически изменяется при наборе. Во время печатания при неправильном вводе символа он подсвечивается красным и после нажатия Backspace пользователю дается возможность заново его ввести. При окончании набора данного фрагмента текста, появляется оставшаяся часть текста режима, если она есть. Если же пользователь закончил ввод всего текста, ему показывается результат, в котором отражается средняя скорость, время печатания текста и существующий рекорд в данном режиме, если он имеется, также обновляя при установлении рекорда список рекордов и указывая на получение рекорда характерным зеленым подсвечиванием и текстовым указанием. | | | | | | |
|  |  |  |  |  | РКСИ.К18.09.02.03.18/2284 | *лист 3* |
|  |  |  |  |  |
|  |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Проектирование**  Use Case диаграмма | | | | | | |
|  |  |  |  |  | РКСИ.К18.09.02.03.18/2284 | *лист 4* |
|  |  |  |  |  |
|  |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Mock-up прототип | | | | | | |
|  |  |  |  |  | РКСИ.К18.09.02.03.18/2284 | *лист 5* |
|  |  |  |  |  |
|  |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Mock-up прототип | | | | | | |
|  |  |  |  |  | РКСИ.К18.09.02.03.18/2284 | *лист 6* |
|  |  |  |  |  |
|  |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Разработка архитектуры ПП  Подойти к реализации данного ПП можно было с разных сторон, выбрав довольно разные по своей концепции архитектуры. Я выбрал ту, что считаю весьма удобной и практичной.  Стоит внести справку, что в моём случае архитектурой ПП является то, как я обрабатывают пользовательствкий ввод, как отвечаю на него и какие вообще есть возможности у пользователя.  **Получение текста**  В архитектуре моего ПП я решил, что, как только пользователь выбрал режим, появляется текст данного режима, притом клавиша, позволяющая обновлять текст данного режима, не даст попасться одному тексту 2 раза подряд.  **Обработка пользовательского ввода**  Полей для ввода как таковых у меня нет.  Я назначил 2 обработчика события на нажатия клавиш.   1. Считывает нажатие любой клавиши, но используем мы его только для Backspace и Esc, выполняющие роль стирания текущего символа и паузы соответственно. 2. Считывает нажатие лишь символьной клавиши, а после чего уже проверяется, та ли клавиша нажата, которая требуется   **Ответ на пользовательский ввод**  После того, как мы определили что нажал пользователь, следуют  соответствующие действия:   1) Esc – останавливаем счет времени и учет скорости, текст становится чуть бледнее. После нажатия символьной клавиши учет времени и скорости опять начинается, а текст становиться, каким был.  2) Backspace – если этому предшествовала ошибка, то просто меняет цвет текущего неправильно введенного символа на обычный, давая возможность ввести его еще раз, а если ошибки не было то просто сдвигает каретку на 1 символ назад, если он и так не первый | | | | | | |
|  |  |  |  |  | РКСИ.К18.09.02.03.18/2284 | *лист 7* |
|  |  |  |  |  |
|  |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 3) Правильно введённый символ – в этом случае каретка переходит на следующий, делая весь предшествующий текст зеленым. Если это был конец фрагмента кода, то он заменяет имеющийся на оставшийся, либо, если такого нет, прекращает возможность ввода и выводит результат.  4) Неправильно введенный символ – каретка на нем и сам символ становится красными. Чтобы избавиться от этого эффекта, нужно нажать Backspace, и уже потом попытаться ввести данный символ еще раз.  **Работа с текстом**  Получаем мы текст либо в виде сплошной строки, либо с имеющимися в нем переносами. Для начала я форматирую текст, устанавливая через определенное кол-во символов перенос строки, по которому и ориентируюсь. Это определенное кол-во символов должно быть на 2-3 меньше того кол-ва символов, которое может поместиться в 1 строку отображаемого блока.   С уже отформатированным текстом я и работаю, считаю кол-во переносов, помещая в блок отображения текста необходимое кол-во текста, а когда этот текст был набран, заменяю его на оставшийся фрагмент.  В процессе замены также проверяется, не конец ли это всего ввода.  **Получение скорости**  Для получения текущей скорости я каждые 2 секунды узнаю кол-во введённых за это время правильных символов, делю на 30 и получаю скорость. Для получения средней скорости в конце ввода я делю кол-во введенных сиволов на потраченное на ввод время в секундах, деленное на 60.  **Вывод и хранение рекордов**  Для рекорда в каждом режиме существует свое свойство в localStorage. При первом запуске, завершении ввода с новым рекордом, очистке рекордов происходит запись в таблицу рекордов. Перезапись конкретного рекорда осуществляется только в случае превышения текущего результата над существующим рекордом. | | | | | | |
|  |  |  |  |  | РКСИ.К18.09.02.03.18/2284 | *лист 8* |
|  |  |  |  |  |
|  |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Разработка программного продукта**  В ходе разработки своего программного продукта я использовал:  1) IDE Visual Studio Code, по причине быстрой работы, большого количества встроенных функций, очень удобный плагины и приятный глазу дизайн.  2) язык программирования JavaScript, так как курсовая у меня по теме, связанной непосредственно с изучением этого языка. А также он довольно удобный и с каждой версией становится все лучше и лучше, о чем говорит его мировая не безызвестность.  3) Библиотека jQuery, так как, во-первых, это было одним из условия моего задания, а во-вторых, она сильно сокращает код и упрощает реализацию многих вещей, для которых на чистом js потребовалось бы куда больше кода  4) Система контроля версий Git, так она позволяла мне осуществлять контроль версий, тем самым упрощая процесс разработки и уменьшая время написания кода.  5) Консоль разработчика Google Chrome, так она очень быстро работает, имеет широкий спектр функций, удобна и понятна в использовании, а также очень лаконично оформлена. | | | | | | |
|  |  |  |  |  | РКСИ.К18.09.02.03.18/2284 | *лист 9* |
|  |  |  |  |  |
|  |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Документация**  rusEz, rusNorm, rusHard, engEz, engNorm, engHard – классы, каждый из которых хранит массив из текстов, тематически совпадающих с названием этого класса.  textInputing(event) – функция, срабатывающая при нажатии символьной клавиши и совершающая много действий, связанных с проверками на ввод, запуском, остановкой учета и счета и т.д.  pauseOrBackTyping(event) – функция, которая Проверяет нажатие паузы кнопкой Esc или стирания напечатанного символа. Осуществляет переход на текущий(при исправлении) или на предыдущий символ(при нажатии без предшествующей ошибки) по нажатию клавиши Backspace  pauseDeals() – функция, которая отключает часть интерфейса, связанного с процессом ввода, останавливает счет времени и учёт скорости.  writeRecords() - Заполняет список рекордов в зависимости от хранящихся на localStarage данных о них  funcForRandom (array, nowText) - Подбирает текст случайным образом так, чтобы он не повторялся за использованным в прошлый раз  TimeGo() - функция, которая осуществляет инкремент счетчика времени  getSpeedTyping(text, time) - функция, которая вычисляет среднюю скорость на протяжении всего процесса ввода   getStringMinuteFromSec(time) - функция, которая преобразует секунды в минуты  speedShowNow() - функция, которая показывает текущую скорость  formatText(text) - функция, которая форматирует текст так, чтобы было легче обрабатывать его | | | | | | |
|  |  |  |  |  | РКСИ.К18.09.02.03.18/2284 | *лист 10* |
|  |  |  |  |  |
|  |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Документация**  getUseText(text) - функция, в которой мы получаем из отформатированного текста тот фрагмент, который можем отобразить  replaceUseText() - функция, которая переносит каретку текущего символа в начало отображаемого фрагмента; заменяет текущий фрагмент на следующий, получив его из оставшегося текста; изменяет оставшийся текст  searshAllEntry(str, target) - функция, возвращающая сумму всех вхождений подстроки в строку | | | | | | |
|  |  |  |  |  | РКСИ.К18.09.02.03.18/2284 | *лист 11* |
|  |  |  |  |  |
|  |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Заключение**  Так как данный программный продукт имеет довольно высокую практическую ценность, перспективы у него определенно есть. Но для этого нужно принять ряд мер, которые будут делать эти самые перспективы все ближе и ближе к реальности.  Сейчас я приведу примеры изменений, которые можно сделать в данном программном продукте и какие меры надо применять для увеличения его практической и общей ценности.  Практическая:  1) добавить больше режимов, относящихся к другим темам  2) добавить режимов, более легких для меньшего порога вхождения в данный программный продукт  3) создание курсов обучения или улучшения навыка скоропечатания с цельной программой, которую, возможно можно будет и магнетизировать  4) сделать более глубокий анализ пройденного текста, акцентирую внимание на слабых местах, указывая на них и предлагая подбор тех режимов и текстом, которые будут направлены на решение этих проблем (возможна монетизация)  Общая:   1. Сделать интерфейс более наполненным, добавить символические картинки к соответствующим смысловым блокам, чтобы пользователю было приятнее находится на сайте как можно дольше. 2. Добавить новые режимы, которые будут заинтересовывать пользователей дольше оставаться на сайте. К таким режимам можно отнести «соревновательный», «без ошибок», «5 минут». 3. Возможность добавлять свои тексты, вплоть до целой книги, где будет возможность сохранять степень прогресса в ее печатании, возвращаясь к ней снова и снова. И всё это на нашем сайте. 4. Добавление возможности создания аккаунтов, чтобы пользователи имели возможность сохранять свои достижения, делиться ими с другими пользователями и получать за это определенные косметические бонусы 5. Добавить рекламу в самые подходящие для пользователя время. Например, кода он поставил рекорд, не ставя его до этого 5 раз. В хорошем расположении духа шанс, что пользователь откликнется на рекламу, выше. | | | | | | |
|  |  |  |  |  | РКСИ.К18.09.02.03.18/2284 | *лист 12* |
|  |  |  |  |  |
|  |  |  |  |  |