

Министерство образования и науки Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ **«Национальный исследовательский ядерный университет МИФИ» (НИЯУ МИФИ)**

**Саровский физико-технический институт – филиал НИЯУ МИФИ**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

ФАКУЛЬТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И ЭЛЕКТРОНИКИ

КАФЕДРА ВЫЧИСЛИТЕЛЬНОЙ И ИНФОРМАЦИОННОЦЙ ТЕХНИКИ

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**К выпускной квалификационной работе**

**На тему:**

**Клавиатурный тренажер**

Студентка Балушкина Анастасия Вячеславовна

Руководитель работы ст. преподаватель Романова М.Д.

Зав. Кафедрой доцент, к.ф-м.н. Холушкин В.С.

г.Саров

2020 г.

Оглавление

[**Введение** 3](#_Toc42716126)

[**Глава 1. Теоретическая часть** 4](#_Toc42716127)

[1.1. Основные технологии создания веб-приложения 4](#_Toc42716128)

[1.2. Анализ клавиатурных тренажеров 4](#_Toc42716129)

[**Глава 2. Практическая часть** 7](#_Toc42716130)

[2.1. Выбор редактора кода 7](#_Toc42716131)

[2.2. Выбор программы проектирования интерфейса 8](#_Toc42716132)

[2.3. Структура и функционал приложения 9](#_Toc42716133)

[2.4. Разработка макета 10](#_Toc42716134)

[2.5. Процесс разработки 15](#_Toc42716135)

[2.6. Перспектива развития 20](#_Toc42716136)

[**Заключение** 22](#_Toc42716137)

[**Литература** 23](#_Toc42716138)

# **Введение**

Умение быстрого набора позволяет свободно владеть клавиатурой. Научиться быстрому набору текстов можно с помощью специальных программ, которые называют клавиатурными тренажерами.

Такие программы часто используют синтез игры и упражнений, базирующихся на серьезных научных методиках обучения быстрому набору.

Ежедневные занятия на клавиатурном тренажере позволяют не только выучить расположение клавиш, но также увеличить скорость набора, уменьшить количество опечаток и улучшить ритмичность набора.

Целью работы является создание клавиатурного тренажера, предназначенная для обучения слепой печати.

Для выполнения проекта были поставлены следующие задачи:

* Создать интерактивные веб-страницы;
* Проектировать интерфейс;
* Изучить инструменты дизайна и редактирования кода для современной веб-разработки;
* Реализовать действующее приложение.

# **Глава 1. Теоретическая часть**

* 1. Основные технологии создания веб-приложения

При создании любого сайта обязательно используются HTML, CSS и JavaScript. Если первые две технологии задают структуру и внешний вид, то при помощи последней можно обеспечить динамику интерфейса. Это, например, обработка полей ввода, проверка правильности данных,

изменений структуры документа.

**HTML/CSS/JS**

С помощью языка гипертекстовой разметки и каскадной таблицы стилей мы можем создавать красивые и удобные интерфейсы. Особенно после появления пятой версии HTML и третей версии CSS.

HTML – язык разметки гипертекста, который позволяет пользователю создавать и структурировать разделы, параграфы, заголовки, ссылки и блоки для веб-страниц и приложений.

CSS - язык описания внешнего вида HTML-документа. Это одна из базовых технологий в современном интернете. Практически ни один сайт не обходится без CSS, поэтому HTML и CSS действуют в единой связке.

JavaScript – это язык программирования, благодаря которому можно добавлять интерактивности веб-страницам. На данный момент язык активно развивается и поддерживает различные парадигмы программирования – наиболее популярны функциональный и объектно-ориентированный подход. Наиболее популярные варианты использования данного языка – проверка правильности заполнения формы, слайдеры, запросы на сервер, любое изменение структуры документа.

* 1. Анализ клавиатурных тренажеров

На данный момент существует немало сервисов и приложений, позволяющих научиться десятипальцевому методу печати. Наиболее популярны следующие решения:

* Stamina – полноценная десктопная программа, доступная для компьютеров с ОС Windows и Linux. Одна из особенностей приложения – забавные звуки при ошибках;



Рис. 1. Stamina.

* Klava.org – сайт, где можно не просто научиться быстро печатать на клавиатуре, но и выбрать определенный режим – программирование, тексты, слова. Один из наиболее функциональных онлайн-сервисов;

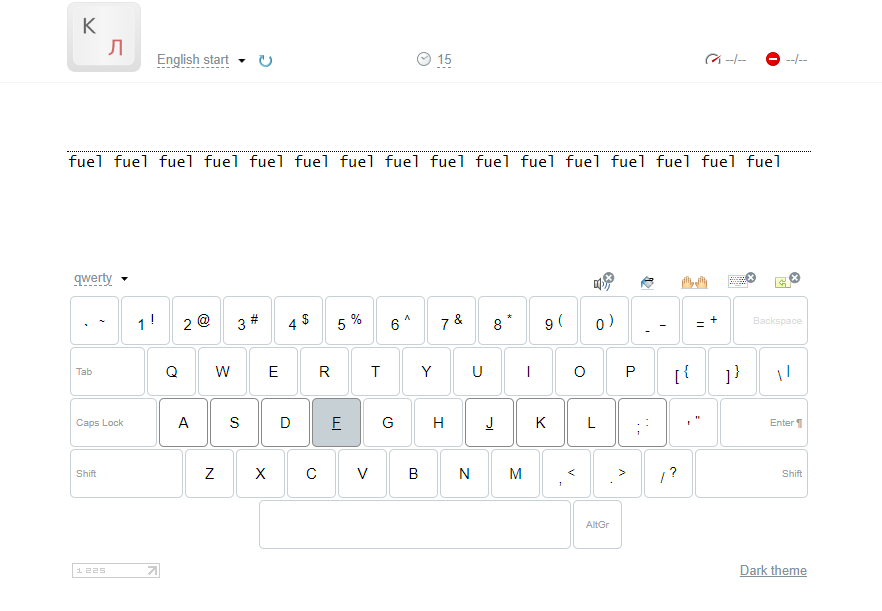


Рис. 2. Klava.org

* Nabiraem.ru – достаточно продвинутый сайт по обучению. Часть услуг доступна без оформления подписки, за более глубокое обучение нужно платить.

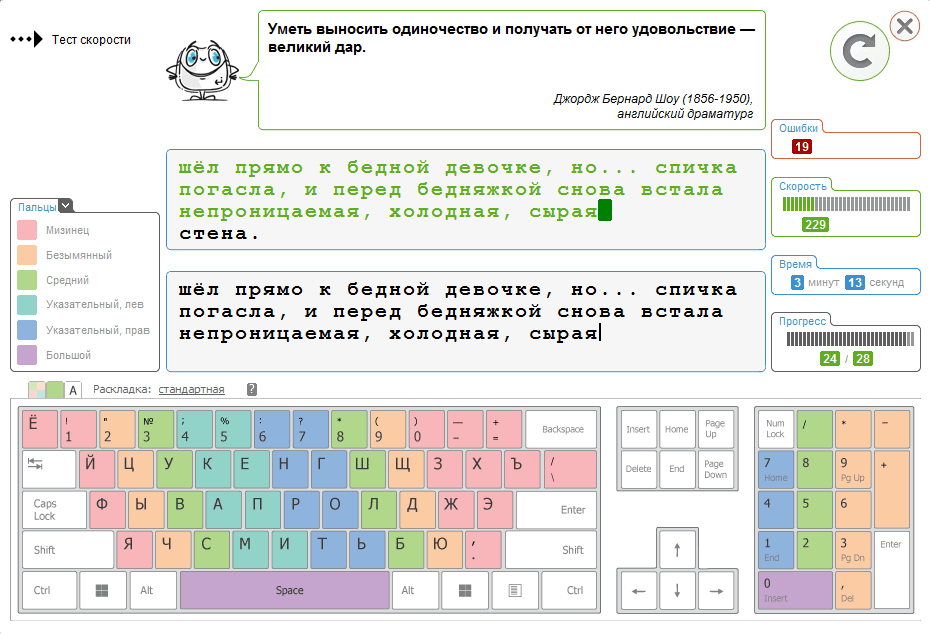


Рис. 3. Nabiraem.ru.

# **Глава 2. Практическая часть**

## 2.1. Выбор редактора кода

Для разработки данного веб-приложения выбран Visual Studio Code.

VS Code является бесплатным программным редактором. Причиной выбора стали, в том числе, такие параметры как:

* поддержка практически всех языков программирования,
* наличие встроенной библиотеки элементов кода,
* подсветка синтаксиса,
* одновременная работа с несколькими проектами,
* поддержка многооконного и двух панельного режима,
* интегрированная система подсказок,
* расширение функционала с помощью плагинов, так как VS Code имеет обширную библиотеку плагинов и подпрограмм среди других бесплатных редакторов.

VS Code поддерживает множество настроек, как для интерфейса, так и для всей программы.

Конечно, существует немало других редакторов кода и полноценных IDE для работы с JavaScript:

* WebStorm – полноценная среда разработки веб-приложений. Множество полезных функций, таких как работа с git через удобный интерфейс, умный анализатор кода и плагины идут «из коробки». Однако, все это не бесплатно. Пользователь должен оформить подписку, либо получить студенческую лицензию;
* Atom – приятный и простой в использовании редактор. Поддерживает все основные языки программирования. Главные недостатки – потребляет много ресурсов системы и не имеет большого количества плагинов, сложнее настроить под себя;
* Sublime Text 3 – некогда популярная программа, высокая производительность и огромное количество уроков по настройке под различные проекты. Однако сейчас его функционал не настолько широк, как у того же VS Code. Подойдет для создания простых сайтов.

## 2.2. Выбор программы проектирования интерфейса

Для проектирования интерфейса был выбран сервис Figma. Это кроссплатформенный онлайн-сервис, предназначенный для разработки дизайна интерфейсов.

Для сравнения возьмем несколько аналогичных приложений - Sketch и Adobe XD, проведем анализ, какое средство будет удобнее и легче в работе.

На сегодняшний день Sketch является самым востребованным приложением для создания интерфейса. Имеет простой и понятный функционал. Минус в том, что пользоваться бесплатно им можно только 30 дней. Далее требуется внести оплату. Еще одним недостатком этого сервиса является невозможность его установки на Windows. Он совместим только с MacOS. Это неудобство особенно сильно сказывается в процессе разработки программы – если программист используют компьютер на Windows, то ему придется использовать дополнительные средства для чтения файлов Скетча.

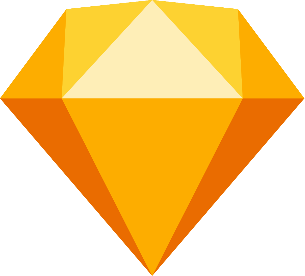


Рис. 4. Логотип Sketch.

Adobe XD был создан как аналог Photoshop для проектирования дизайна сайтов или мобильных приложений. Однако, ввиду политики компании и не такому широкому функционалу как у конкурентов, Adobe XD не получил большой популярности. Хотя основные потребности дизайнеров данная программа закрывает и ее вполне можно использовать на реальных проектах.



Рис. 5. Логотип AdobeXD.

Наиболее популярным сервисом для создания интерфейсов считается Figma. Разработчики проделали большую работу для того, чтобы создать действительно производительное ПО, способное стабильно и корректно работать на самых простых устройствах. Данный продукт можно полноценно использовать в браузере, но, если хочется иметь локальный доступ к макетам – можно скачать десктопное приложение. Кроме того, дизайнеры могут даже экспортировать CSS прямо внутри программы, что улучшает связь с командой разработчиков. Для индивидуального пользования абсолютно весь функционал бесплатен. Платной опцией является только возможность командного ведения проекта. Сервис доступен на таких операционных системах как Windows, MacOS, Linux.

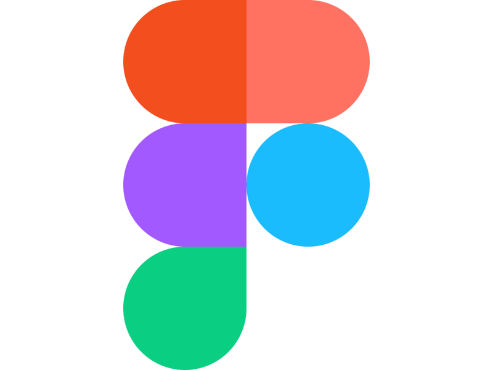


Рис. 6. Логотип Figma.

## 2.3. Структура и функционал приложения

После того, как была определена среда разработки, необходимо определить структуру и функционал приложения. Ранее упоминалось о том, что структура сайта должна быть простой и дружелюбной в использовании.

Поэтому структура должна содержать следующее:

* Страницы с клавиатурным тренажерным;
* Страницы с правилами.

Далее необходимо определить функционал сайта:

* Возможность просматривания правил;
* Переключение светлого и темного фона;
* Подсвечивание клавиш определёнными цветами, в соответствии со схемой, правильности расположения пальцев каждой руки.
* Отображение клавиш определённым цветом в зависимости от правильности нажатия (красный – неверно, зелёный - верно).

## 2.4. Разработка макета

При разработке современного сайта очень важно учитывать его внешний вид и функционал.

Интерфейс должен быть максимально понятным и дружелюбным для пользователей. Именно поэтому этап разработки макета особенно важен для любого проекта.

Прежде чем приступить к работе с клавиатурным тренажером, необходимо знать правила работы с ним. Для этого была создана вкладка с правилами, которая появляется при запуске программы (рис. 7). Вкладку также можно вызвать по нажатию на надпись «Правила».

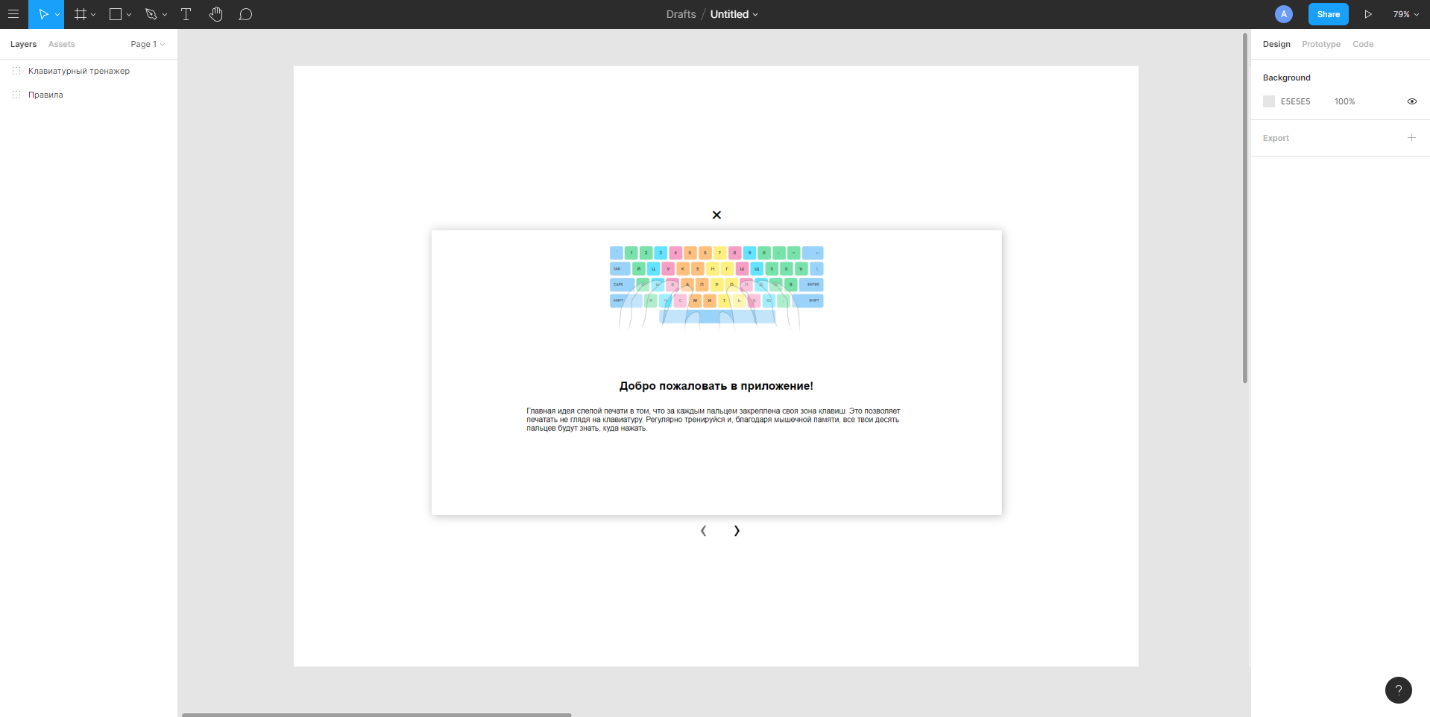


Рис. 7. Разработанный дизайн правил в онлайн-сервисе “Figma”.

После ознакомления с правилами, пользователь попадает на страницу самого тренажера. Он выглядит как клавиатура раскладки QWERTY и тренировочный текст над ней, который предлагается набрать пользователю (рис. 8).

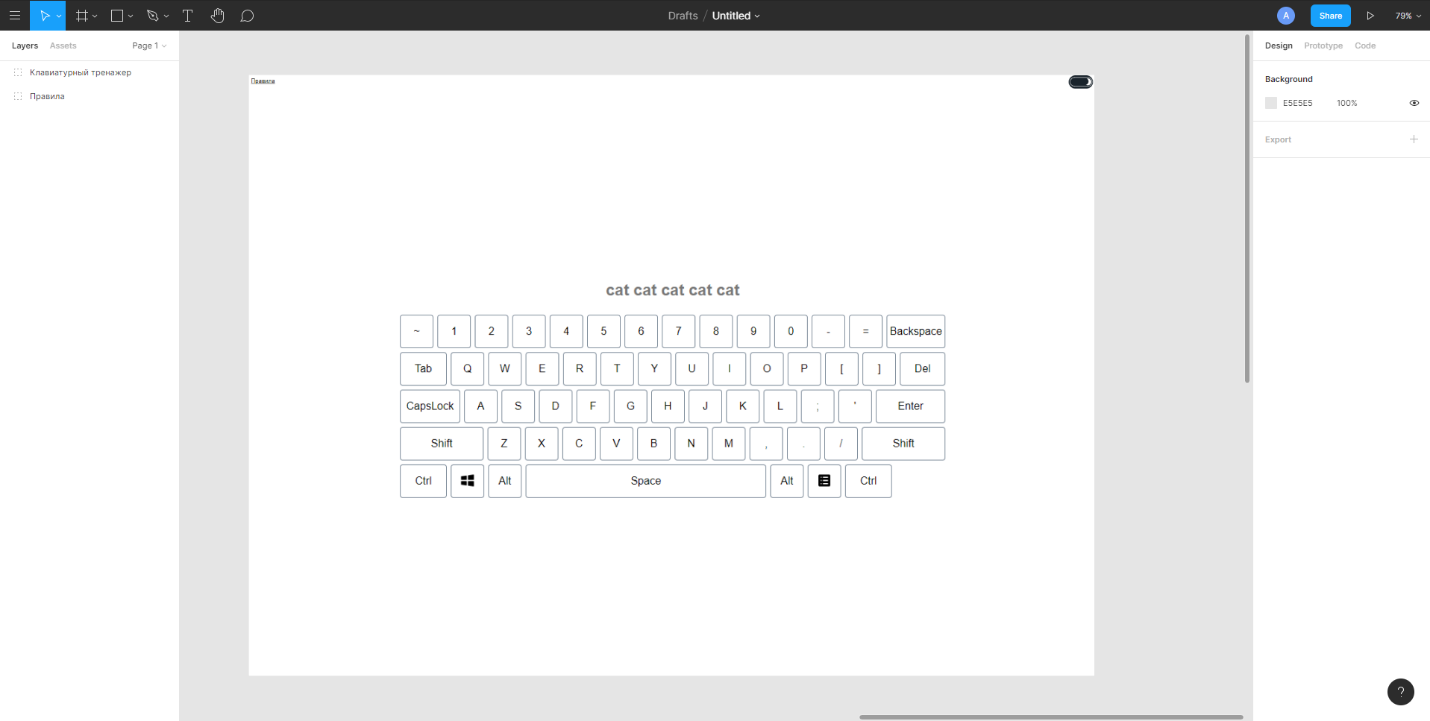


Рис. 8 Интерфейс клавиатурного тренажера.

Для комфортных занятий в вечернее время предусмотрен ночной режим. Он активируется при нажатии соответствующего переключателя (рис. 9).

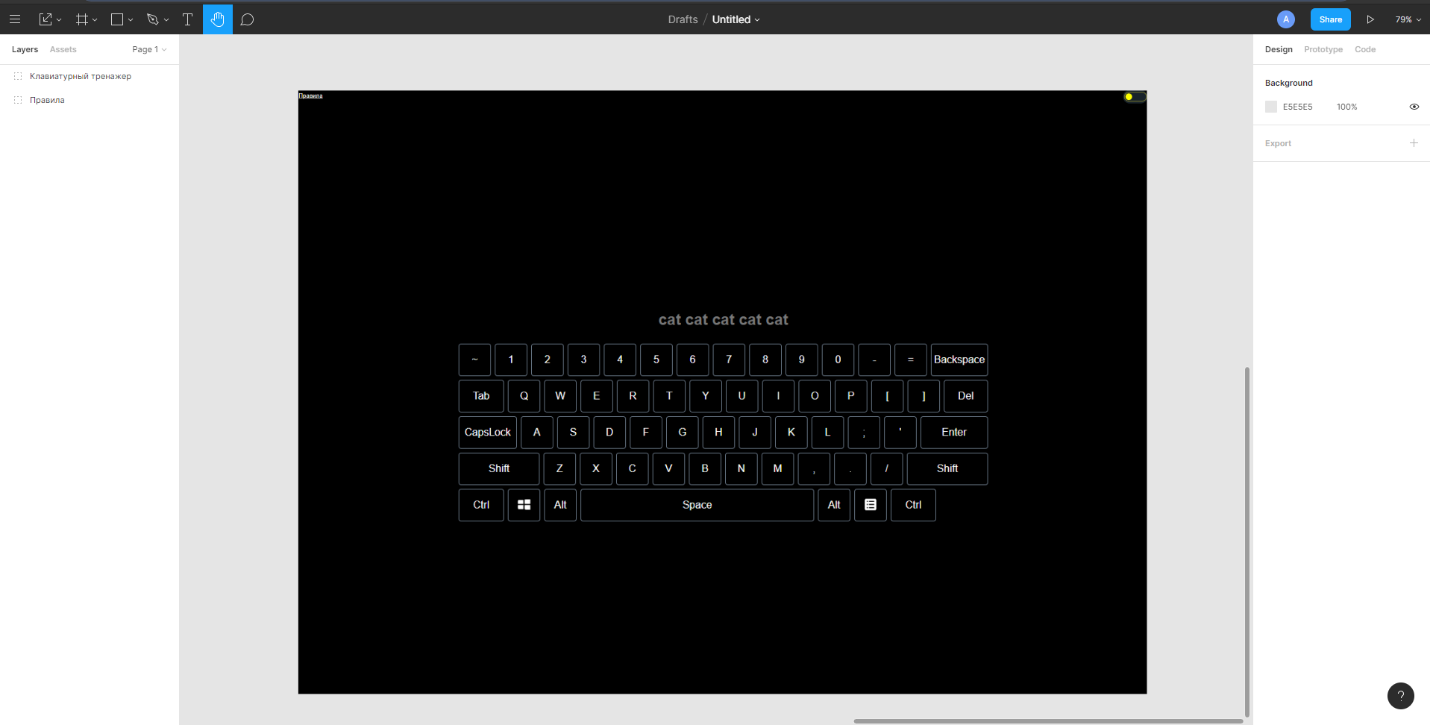


Рис. 9. Дизайн ночного режима в Figma.

Над клавиатурой расположены тренировочные слова. При правильном наборе, количество слов увеличивается. Это предусмотрено для повышения правильности печати (рис. 10-11).

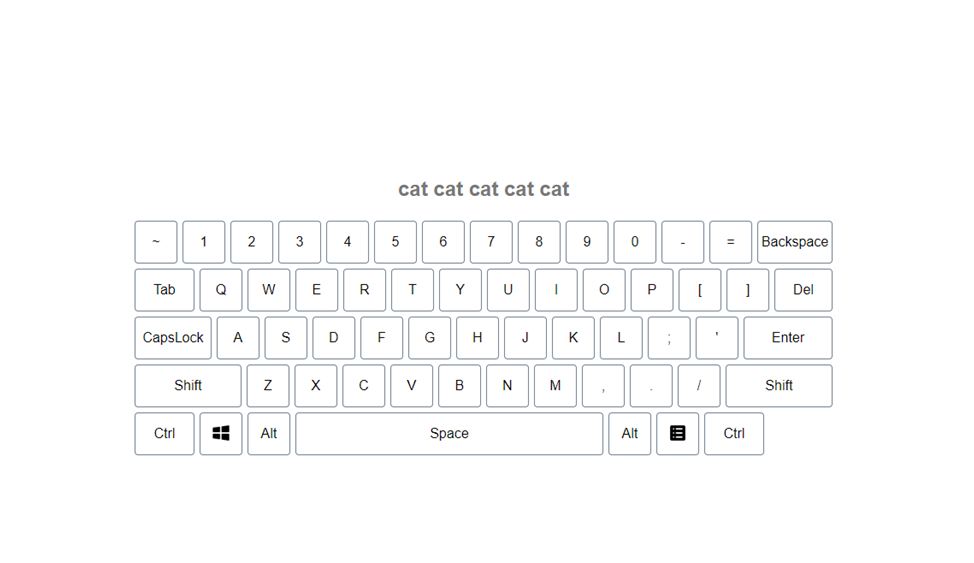


Рис. 10. Начальные тренировочные слова.

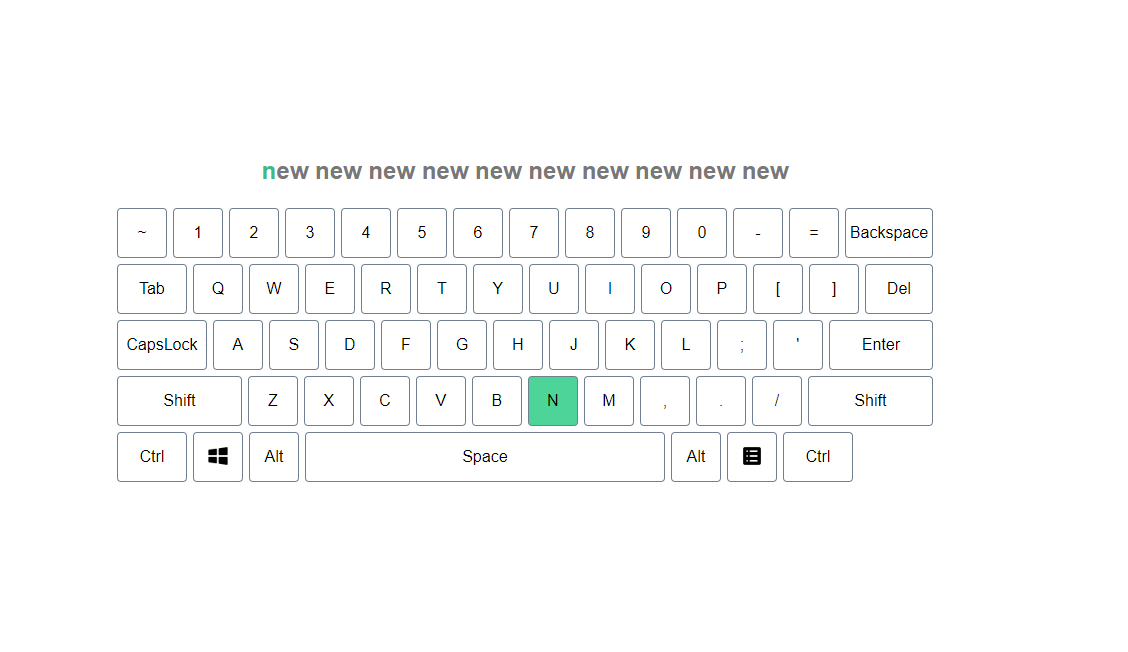


Рис. 11. Следующие тренировочные слова.

При нажатии соответствующей буквы, клавиша подсвечивается зеленым, в ошибочном случае красным. Правильно написанные буквы в слове подсвечиваются зеленым цветом (рис. 12-13).



Рис. 12. Выделение зеленым цветом надлежащей клавиши, и буквы в слове.

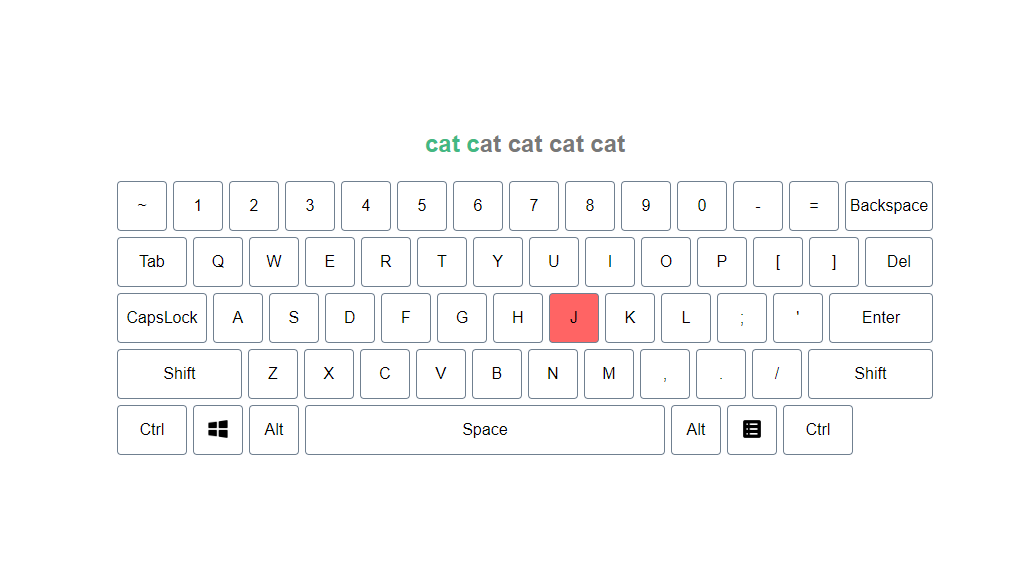


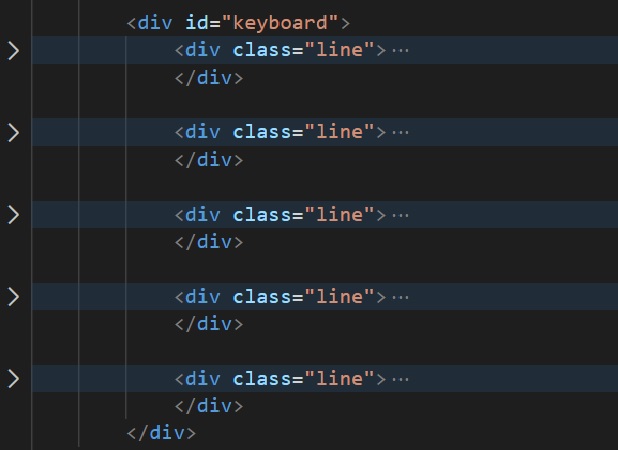
Рис. 13. Выделение красным цветом ошибочной кнопки клавиатуры.

Для пользователя введена подсказка, в виде акцентирования клавиши. (рис. 14). При наводке на кнопку, она отображает тот цвет, на котором должен стоять определённый палец, как указано в правилах (рис. 4).

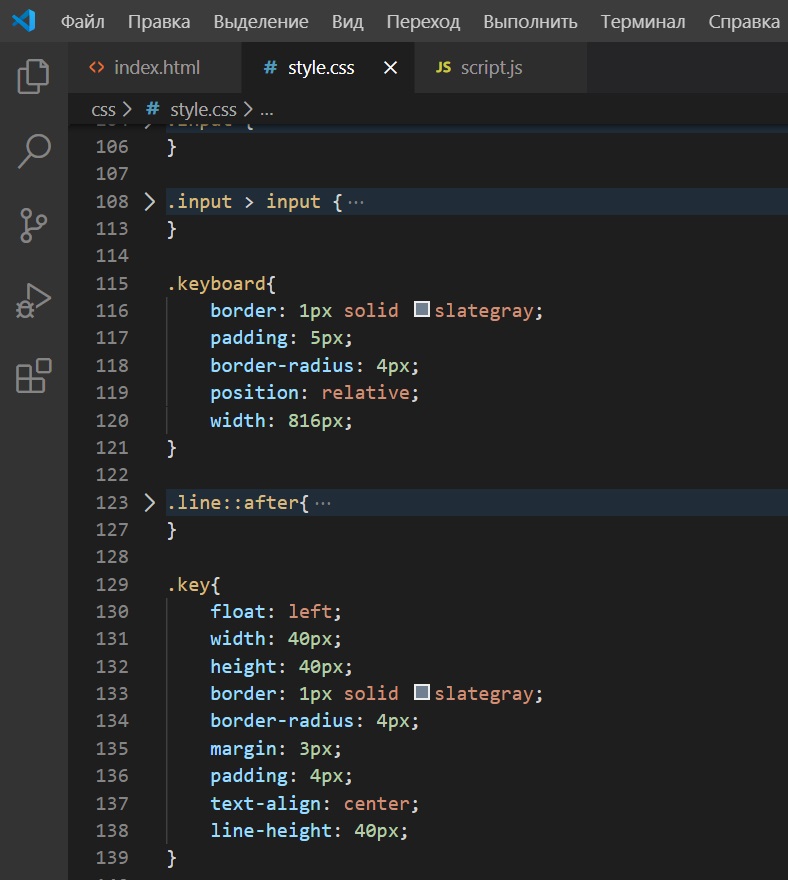


Рис. 14. Подсвечивающаяся клавиша, которая означает, что на клавишу “W” нужно нажать безымянным пальцем левой руки.

## 2.5. Процесс разработки

После проектирования интерфейса, следует процесс написания кода. Для начала необходимо создать HTML разметку. Поскольку основой приложения является виртуальная клавиатура, ее и следует сверстать.

Данная секция кода представляет собой пять блоков – это строки клавиатуры. Внутри каждого class=”line” находится обертка для отдельной клавиши. Чтобы вся эта верстка приобрела привычный для пользователя внешний вид, следует прописать немного css-стилей.



После того, как базовая разметка была спроектирована – самое время добавить виртуальной клавиатуре интерактивности при помощи JavaScript.

Для начала создадим одномерный массив, который будет выступать в роли словаря. Его легко дополнять новыми словами – достаточно просто несколько раз прописать одно и то же слово, и оно тут же покажется на странице.



Далее массив строк выводится на экран при помощи стандартного API браузера.

Наиболее интересным этапом была проверка правописания, подсвечивание вводимых букв.

JavaScript позволяет отслеживать события клавиатуры, поэтому при нажатии на клавишу мы будем постоянно узнавать ее код и сравнивать с текстом на экране. Поскольку с клавиатуры мы вводим символы по одному, слова на экране следует также разбить на отдельные буквы.

Делается это при помощи встроенных функций языка.

Если значение кода клавиши совпало с буквой на экране, буква записывается в новый тег span, которому задается класс «color» и прописывается зеленый цвет в CSS.

Кроме подсветки самой буквы из предлагаемого текста, подсвечивается и нажатая клавиша (также зеленым цветом). Принцип работы такой же.

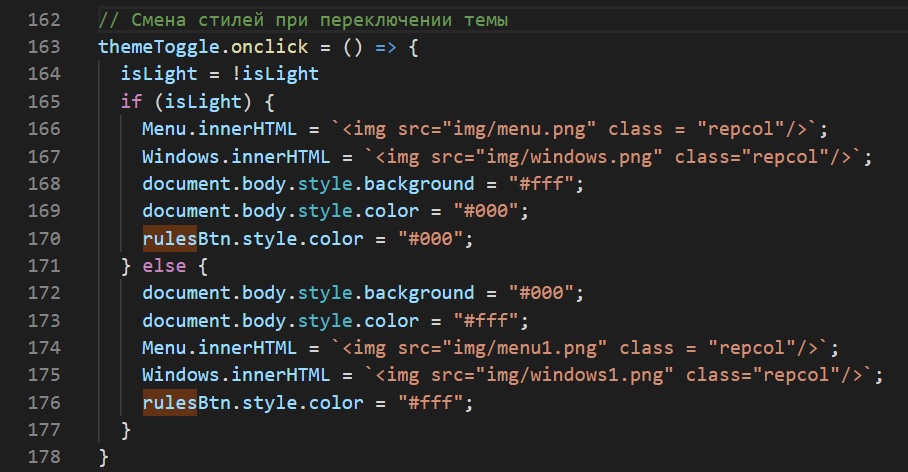
В случае, если пользователь ошибся и введенный им символ не соответствует букве на экране, нажатая клавиша подсвечивается красным цветом.

HTML классы, отвечающие за правильность/неправильность ввода, уничтожаются при каждом нажатии и заново добавляются при выполнении определенных условий.

Современные веб-приложения позволяют менять цветовую тему оформления со светлой на темную. Это делает использование программы комфортнее в темное время суток, поскольку нагрузка на глаза уменьшается.

В клавиатурном тренажере есть достаточно удобный анимированный переключатель, нажав на который все цвета будут заменены на противоположные.

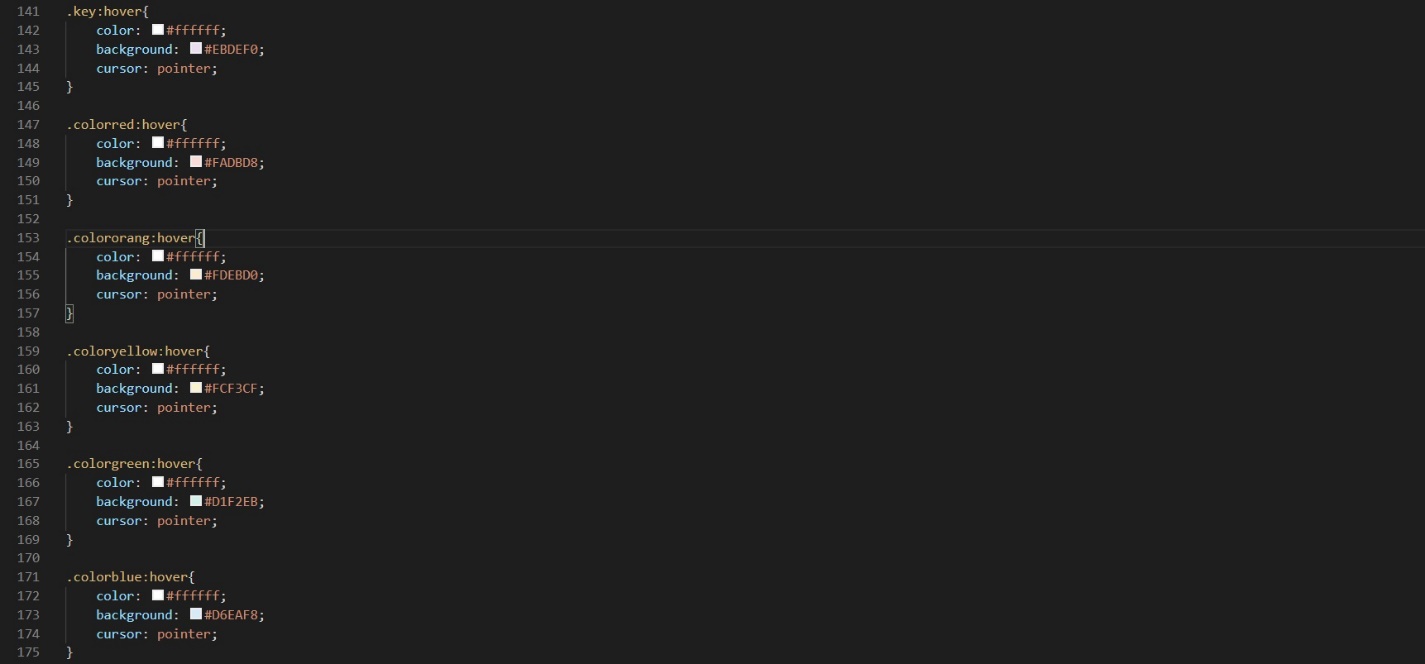
Такой прием реализуется очень просто – мы получаем элементы со страницы и стилизуем их через свойство style в JavaScript.



Однако, при работе с переключателем темы были свои трудности. Если кликнуть по переключателю и потом нажать на пробел – тема опять менялась и это довольно сильно портило общее впечатление от программы. Это происходит потому, что смена темы реализована при помощи input с типом checkbox. Его особенность в том, что этот тег реагирует именно на пробел.

Решить проблему удалось с помощью проверки – если нажат пробел и checkbox в фокусе, то нужно снять фокус. В результате удалось решить проблему и сделать использование приложения более предсказуемым.

Помимо интерактивности при работе с клавиатурой и переключении темы, необходимо было реализовать удобные подсказки, чтобы пользователь иметь представление о том, каким пальцем на какую клавишу следует нажимать. Такой функционал не потребовал работы с JavaScript, достаточно было задать определенные классы секциям клавиш и через CSS поработать с эффектами при наведении (hover-эффектами).



Итогом является стабильно функционирующее приложение, в котором есть необходимый минимум для обучения печати десятипальцевым методом.

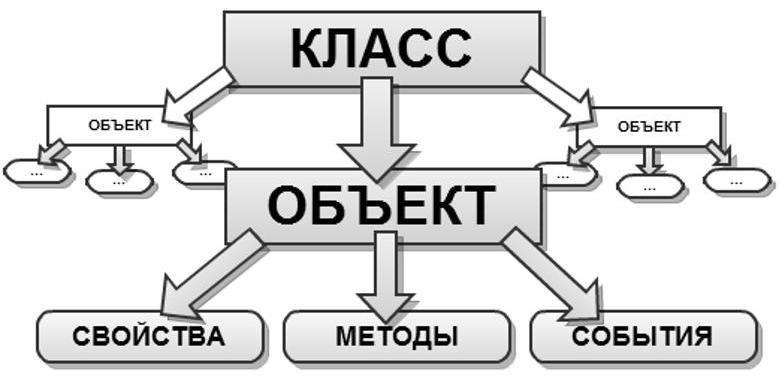
## 2.6. Перспектива развития

Как и любое программное обеспечение, клавиатурный тренажер можно развивать как в плане функционала, так и в плане оптимизации.

Из новых функций можно добавить возможность изменять словарь слов – так каждый пользователь будет использовать для тренировок тот текст, который ему по душе. Возможность засекать скорость печати позволила бы следить за прогрессом в обучении, равно, как и счетчик опечаток.

Из оптимизаций – написать менее монолитный код, чтобы новые функции было проще внедрять и использовать. А поскольку операции с DOM довольно затратны по производительности, можно более эффективно работать с элементами на странице.

Еще один способ улучшить приложение – переписать его с использованием современных библиотек и фреймворков. Но это было бы уместно в том случае, если функционал программы был действительно большим и неконтролируемым с точки зрения кода.



Идей по доработке и добавлению новых возможностей много. Однако, не менее важным является выпуск первой версии приложения, в которой все будет стабильно работать и отвечать минимальным запросам пользователей.

# **Заключение**

Итогом выполненной работы является действующее приложение, помогающее научиться печати десятипальцевым методом. В приложении предусмотрены:

* краткая инструкция,
* словарь,
* подсказки,
* возможность менять цветовую схему для удобной работы.

В процессе проектирования были проанализированы инструменты дизайна, такие как Figma, Sketch и Adobe XD. А также были изучены редакторы кода для современной веб-разработки – Visual Studio Code, WebStorm, Atom и Sublime Text 3.

С помощью кроссплатформенного онлайн-сервиса – Figma, был спроектирован интерфейс приложения, а сама разработка выполнена посредством HTML/CSS/JS.

Получившийся проект полностью работоспособен, прошел этап тестирования, стабилен и готов к работе. Если сравнить его с аналогами, то рассматриваемый клавиатурный тренажер имеет следующие особенности:

* Максимально простой функционал - с его помощью новички смогут максимально быстро приступить к обучению;
* Интерактивность и удобство подсказок - минимум теории и максимум практики;
* Клавиатурный тренажер способен работать на любом компьютере, где установлен любой браузер.

Таким образом, следует отметить:

* поставленные задачи выполнены,
* итоговая цель достигнута.

# **Литература**

1. Дэвид Сойер Макфарланд. Новая большая книга CSS — Санкт-Петербург: Питер, 2017. — 720 с.
2. Стивен Шафер. HTML, XHTML и CSS. Библия пользователя – Москва, 5-е издание, 2011. — 656 с.
3. Резиг Джон, Бибо Беэр, Марас Иосип. Секреты Javascript ниндзя – Москва, 2-е изд., 2017. — 544 с.
4. Флэнаган Д. JavaScript подробное руководство, 6-е изд. – СПБ.: Символ-Плюс, 2012. – 1080 с.
5. Флэнаган Д. JavaScript карманный справочник, 3-е изд. – М.: ООО «И.Д. Вильямс», 2013. – 320 с.
6. Фаулер, Мартин. Рефакторинг кода на JavaScript: улучшение проекта существующего кода, 2-е изд. — Москва, 2019. — 464 с.
7. Бен Фрэйн. HTML5 и CSS3. Разработка сайтов для любых браузеров и устройств – Санкт-Петербург: Питер, 2017. – 272 с
8. Фримен, Робсон. Изучаем программирование на JavaScript – O`reilly, 2015 – 640 с.
9. Дуглас Крокфорд. Как устроен JavaScript – Санкт-Петербург: Питер, 2019. – 304 с.
10. Хрусталев, Кириченко. Основы современного WEB-дизайна – Наука и Техника, 2018. – 352 с.
11. Web Dev с нуля, YouTube, <https://www.youtube.com/watch?v=3fy6jUKO_ro> , свободный доступ.
12. Web Dev с нуля, YouTube, <https://www.youtube.com/watch?v=EqhY1b4I1Zo&list=PLM7wFzahDYnFxysA3eKw3yj3exkw-qFMR>, свободный доступ.
13. Фрилансер по жизни, YouTube, <https://www.youtube.com/watch?v=z3GS5oYGq5U&list=PLM6XATa8CAG4F9nAIYNS5oAiPotxwLFIr>, свободный доступ.
14. Владилен Минин, YouTube, <https://www.youtube.com/watch?v=Bluxbh9CaQ0>, свободный доступ.
15. Glo Academy, YouTube,  
    <https://www.youtube.com/watch?v=xtksHZdcHss&list=PL3LQJkGQtzc6c2zdHBGFkrKZ848nZzl9v>, свободный доступ.