

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий

(наименование института полностью)

Кафедра «Прикладная математика и информатика»

(наименование)

09.03.03 Прикладная информатика

(код и наименование направления подготовки/специальности)

Разработка социальных и экономических информационных систем

(направленность (профиль)/специализация)

## ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему Интерактивный тренажёр «Читай, Считай, Пиши

Обучающийся

Е. А. Евдокимов

(Инициалы Фамилия)

(личная подпись)

Руководитель

д.с.н., доцент Е.В. Желнина

(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Тольятти 2023

## Аннотация

Данное приложение создано для дошкольников и младших школьников. Обучение детей всегда является актуальной темой и полезным направлением для разработки программного обеспечения, поэтому было решено выбрать именно эту тему. Целью проекта было упростить получение навыков письма и чтения с помощью приложения с простой и понятной визуализацией, удобным интерфейсом при низких системных требованиях.

Основной задачей создания и реализации тренажера является обучение детей в интерактивном режиме. Развитие образования в современных условиях – это организация учебной деятельности путём создания ИКТ образовательной среды информационно коммуникационные технологии – в сервисе обучение и ИС с веб-приложениями.

В работе рассматривается разработка web-приложения, которое позволяет: оптимизировать учебную деятельность, обеспечить эмоциональность и занимательность подачи учебного материала, контролировать выполнение заданий. Все это создает благоприятный фон для достижения успеха в образовательном процессе. Пользователи данного приложения - родители, учителя, учащиеся. Приложение разработано с использованием html, css, javascript, php, php framework laravel.

Структура работы включает в себя следующие элементы: титульный лист; аннотация; содержание; введение; основная часть; заключение; список используемой литературы.

Объём выполненной работы составляет 40 страниц.

## Оглавление

Введение .....	4
Глава 1 Теоретическая часть .....	7
1.1 Теоретические аспекты создание веб-приложения .....	7
1.2 Предпроектное исследование .....	9
1.3 Используемые программные продукты.....	12
Глава 2 Создание веб-приложения .....	15
2.1 Составление технической документации.....	15
2.2 Разработка веб-приложения.....	16
2.3 Тестирование веб-приложения.....	29
2.4 Публикация веб приложения .....	30
Глава 3 Расчет затрат .....	32
3.1 Экономическое и организационное обоснование проекта .....	32
3.2 Расчет себестоимости .....	32
Глава 4 Безопасность жизнедеятельности и охрана труда.....	35
Заключение.....	37
Список используемой литературы .....	39

## Введение

Данный проект предназначен для сотрудников детских садов и школ для обучения детей в интерактивном режиме.

Основная задача – реализовать тренажёр для удаленного обучения детей в интерактивном режиме. Была прекращена поддержка старой программы для обучения детей письму, поэтому появилась потребность создать интерактивный тренажёр, который помогает учителям заинтересовывать детей обучаться, так как материал представлен в игровой форме.

Один из главных векторов развития образования – это организация учебной деятельности путем создания ИКТ образовательной среды. К информационным коммуникациям относятся сервисы обучения и ИС с веб-приложениями. Использование веб-приложений в образовательном процессе упрощает обучение для учителей и учеников. Обеспечивает обратный отклик, с помощью веб-приложений можно честно оценить и проверять успеваемость учащихся. В время распространения коронавирусной инфекции школы, колледжи, ВУЗы, по рекомендации Министерства науки и высшего образования РФ «Временно приостановить посещение обучающимися указанных организаций... до даты окончания действия ограничительных мер...» и «Обеспечить реализацию образовательных программ, в том числе прохождение обучающимися промежуточной аттестации и государственной итоговой аттестации по соответствующим образовательным программам с применением электронного обучения и дистанционно образовательных технологий» [1] приняли решение о переходе на дистанционное обучение. И распоряжение Министерства просвещения РФ «Об утверждении методических рекомендаций для внедрения в основные общеобразовательные программы современных цифровых технологий» [2]

Цифровая образовательная среда – это комплекс средств для реализации образовательных программ с применением электронного обучения, дистанционных образовательных технологий, обеспечивающих освоение

обучающимися образовательных программ в полном объёме независимо от места нахождения обучающихся.

На данный момент тренажеров для детей не так много, а если и есть – стоят весьма дорого. Мой продукт будет иметь приемлемую цену на рынке и предоставлять последние технологии для удаленной работы с детьми.

Цель: разработка интерактивного тренажера «Читай, Считай, Пиши».

Решаемые задачи:

- разработка технического задания;
- изучение аналогов разработанного тренажера;
- дизайн модуля в программе Figma;
- разработка клиентской части JS;
- разработка socket.io сервера на Node.JS;
- тестирование программного продукта;
- выпуск программного продукта.

Объектом исследования является автоматизация обучения детей писать, читать, считать в интерактивном режиме.

Предметом исследования является интерактивный тренажёр «Пиши, Читай, Считай»

В данной работе использовались следующие методы исследования:

- анализ интерактивных тренажеров;
- тестирование готового приложения;
- расчет доходности проекта.

Анализ помог выявить принцип работы интерактивных приложений, дал ориентир куда нужно двигаться, от чего отталкиваться. Без его помощи не было бы готового приложения. А приложение, в свою очередь, для дальнейшего использования необходимо протестировать, дабы сократить ошибки и оптимизировать.

Также основным методом исследования является – расчёт доходности проекта, позволяющий понять есть ли у проекта коммерческая выгода.

Сейчас идёт апробация данного программного продукта у заказчика. По этому тренажёру было проведено несколько занятий с учениками. Детям было интересно осваивать материал, так как он был в игровой форме.

Выпускная квалификационная работа включает в себя введение, четыре главы основной части, заключение. Введение объясняет актуальность работы, обозначает цель, задачи работы. Аннотация содержит в себе общие сведения работе, смысл выполненной работы и главные результаты. В первой главе представлена теоретическая часть, описаны используемые для разработки приложения программные продукты. Во второй главе подробно показан процесс создания веб-приложения: оставление технической документации, разработка и тестирование веб-приложения. Третья глава посвящена расчету затрат, необходимых для разработки, публикации и последующей работы веб-приложения. Четвертая глава содержит рекомендации по обеспечению безопасности жизнедеятельности и охране труда при использовании предлагаемого веб-приложения. В списке используемой литературы находится 21 источник. Объем ВКР 40 страниц.

## Глава 1 Теоретическая часть

### 1.1 Теоретические аспекты создание веб-приложения

Фреймворк – программное обеспечение, которое помогает облегчить разработку и интегрирует все компоненты программного проекта.

Различия фреймворка от библиотеки

Библиотеку можно использовать в программном продукте как несколько подпрограмм схожего функционала, не оказывающего влияния на архитектуру программного продукта и не накладывающего на неё каких-либо запретов. Фреймворк наоборот задаёт принципы организации архитектуры приложения, введя на ранней стадии создания понятие – «каркас», который необходимо увеличивать и модернизировать в соответствии с заданными потребностями. Хорошим образцом программного фреймворка служит – С.М.Ф. (Content Management Framework), а библиотеки — модуль электронной почты.

Фреймворк может включать в себя немало разнообразных по теме библиотек, а библиотека содержит в себе набор близкой функциональности

Основное различие «фреймворка» от библиотеки – изменение управления и переход управления от пользователя к программному коду. Пользовательский код запрашивает возможности библиотеки и получает возможность управления после вызова. Во фреймворке пользовательский код может реализовывать конкретное поведение, встраиваемое в более общий – «абстрактный» код фреймворка. При этом «фреймворк» вызывает функции (классы) пользовательского кода.

Веб-фреймворк (WF) или платформа веб-приложений (WAF) – это программная среда, рассчитанная на помощь создания веб-приложений, веб-служб, веб-ресурсов и веб-API. Веб-фреймворки обеспечивают шаблонный метод разработки и развертки веб-приложений в глобальной паутине. Веб-фреймворки ориентированы на автоматизацию времени, связанного со

стандартными действиями, производимыми в веб-разработке. Примером служат различные веб-фреймворки, предоставляющие библиотеки для обеспечения доступа к базам данных, управление сессиями, и они во многом служат неоднократному воспроизведению кода.

Они используются для создания как динамических веб-сайтов, так и для статических.

Laravel – свободный веб-фреймворк с доступным кодом, ориентированный на разработку с возможностью использования архитектурной модели MVC (англ. Model View Controller – модель–представление–контроллер). Laravel вышел под лицензией MIT.

Исходный код проекта размещается на GitHub. В результате опроса на сайте [sitepoint.com](http://sitepoint.com) в декабре 2013 года о самых востребованных PHP-фреймворках Laravel оказался самым перспективным проектом на 2014 год.

В 2015 году в результате опроса того же [sitepoint.com](http://sitepoint.com) по применению PHP-фреймворков среди разработчиков занял призовое место в номинациях:

- фреймворк корпоративного уровня;
- фреймворк для личных проектов;
- кладезь антипаттернов.

Веб-приложение — это программное обеспечение, которое запускается в веб-браузере. Компании должны обмениваться информацией и оказывать услуги удаленно. Они используют веб-приложения для удобной и безопасной связи с клиентами. Наиболее распространенные функции веб-сайта, такие как корзины, поиск и фильтрация товаров, обмен мгновенными сообщениями и ленты новостей социальных сетей, по своей структуре являются веб-приложениями.



## 1.2 Предпроектное исследование

Уже сделано много детских тренажёров например: Логопедический тренажер «Научи Смешариков различать звуки с-ш» представлена на рисунке 1.



Рисунок 1 – Логопедический тренажер  
«Научи Смешариков различать звуки с-ш»

Достоинства:

- легкость в использовании;
- множество различных эффектов.

Недостатки:

- тренажер работает только с двумя звуками.

Интерактивный тренажер по иностранному языку для учащихся 5 класса представлен на рисунке 2.



Рисунок 2 – Интерактивный тренажер по английскому языку для учащихся 5 класса

Достоинства:

- легкость в использовании;
- большая база английских слов.

Недостатки:

- тренажер обучает только английскому языку.

Интерактивный тренажер по математике «Зодиак. Координатная плоскость» представлен на рисунке 3.



Рисунок 3 – Интерактивный тренажер по математике «Зодиак. Координатная плоскость»; 6 класс

Достоинства:

- легкость в использовании;
- хорошо раскрывается тема.

Недостатки:

- только одна тема.

Презентация по психологии «Радуга эмоций» для учеников представлен на рисунке 4.

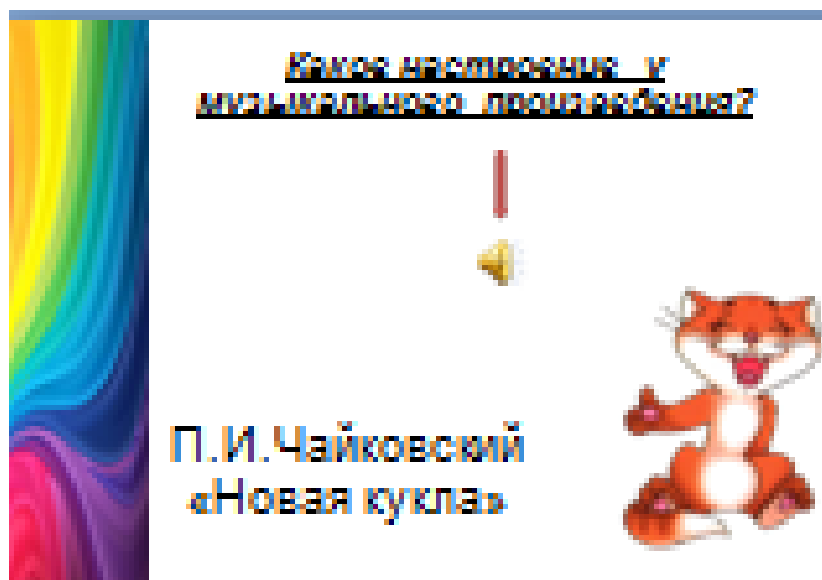


Рисунок 4 – Презентация по психологии «Радуга эмоций» для учеников

Достоинства:

- легкость в использовании;
- мало весит.

Недостатки:

- мало информации.

### 1.3 Используемые программные продукты

В работе были использованы следующие программные продукты:

Figma – Онлайн-сервис для разработки интерфейсов и прототипирования, предоставляющий организацию общей работы в режиме настоящего времени. Сервис доступен по подписке, предусмотрен бесплатный тариф для одного пользователя.

Docker - программное обеспечение для автоматизации развёртывания и управления приложениями в средах с поддержкой контейнеризации. Контейнеризатор приложений даёт возможность собрать приложение со всем его окружением и зависимостями в контейнер, который может быть развёрнут на любой Linux-системе с поддержкой контрольных групп в ядре, а также

обеспечивает набором команд для управления этими контейнерами. Первоначально использовал потенциал LXC, а с 2015 года начал использовать собственную библиотеку, абстрагирующую виртуальные возможности ядра Linux – libcontainer. Open Container Initiative способствовал переходу от монолитной к модульной архитектуре.

VS Code - текстовый редактор, разработанный Microsoft для Windows, Linux и macOS. Представляется как «лёгкий» редактор кода для кроссплатформенной разработки облачных и веб-приложений. Содержит в себе отладчик, инструменты для работы с Git, подсветку синтаксиса, IntelliSense и средства для рефакторинга. Располагает обширными возможностями для кастомизации: пользовательские темы, сочетания клавиш и файлы конфигурации. Предоставляется бесплатно, разрабатывается как программное обеспечение с открытым исходным кодом.

Google Chrome - браузер, созданный компанией Google на основе браузера Chromium, представленного в широком доступе, и движка Blink.

Chromea предоставляет персональная подборка новостей, быстрый переход на любимые сайты, скачивание контента, а также встроенные Google Поиск и Google Переводчик.

Особый режим в Chrome способствует меньшему расходу времени и трафика. Браузер уменьшает размер текста, изображений, видео и сайтов не в ущерб качеству, что экономит до 60% трафика.

Текст переводится на экране благодаря встроенному Google Переводчику.

Chrome фиксирует и принимает во внимание круг интересов пользователя. Открывая новую вкладку в Chrome, можно заметить статьи, основанные на персональной истории просмотров. Для того чтобы сделать поиск в Google ещё быстрее, Chrome высвечивает варианты запросов и веб-адресов – они появляются при вводе текста. Есть возможность сохранить конфиденциальность. Режим инкогнито, не сохраняет открываемые веб-страницы в истории просмотров.

Синхронизация данных. При входе в Chrome закладки, пароли и настройки автоматически синхронизируются и становятся доступны на всех устройствах, где установлен Chrome.

## Выводы по Главе 1

Таким образом, изученный материал по созданию веб-приложений обладает определёнными достоинствами: легкость в использовании, и недостатками: один из которых - несоответствие выбранному функционалу.

Figma – Онлайн-сервис для разработки интерфейсов и прототипирования.

Docker – программное обеспечение для автоматизации развёртывания и управления приложениями в средах с поддержкой контейнеризации.

VS Code – текстовый редактор, разработанный Microsoft для Windows, Linux и macOS.

Google Chrome – браузер, разрабатываемый компанией Google на основе свободного браузера Chromium и движка Blink.

Данный теоретический материал позволил создать продукт: интерактивный тренажер «Читай, считай, пиши».

## Глава 2 Создание веб-приложения

### 2.1 Составление технической документации

При разработке данного приложения использовались следующие продукты:

- php framework laravel (фреймворк);
- php (язык программирования);
- JQuery (javascript библиотека);
- Javascript (язык программирования);
- HTML (язык гипертекстовой разметки);
- CSS (язык таблиц стилей).

Авторизация и регистрация происходят с помощью библиотек фреймворка.

При отправке сообщение в техническую поддержку используется PHPMailer

Перенос и установка букв происходит посредством javascript:

- при зажатии левой кнопкой мыши на элементе происходит захват буквы и передача позиции курсора букве. После отпущения левой кнопки мыши элемент находит рядом стоящее поле и происходит передача позиции поля букве, если поле не находится, то буква возвращается на место;

- когда буква установлена, то её можно удалить двойным нажатием левой кнопкой мышки по букве;

- смена регистра происходит с помощью css стиля lowercase

При нажатии правой кнопкой мыши на букву, если стиля lowercase нет, то он добавляется, и буква становится меньше если есть, то убирается, и буква становится больше.

Работа с цифрами происходит обычно, стиль lowercase на цифры не действует.



## 2.2 Разработка веб-приложения

Сначала был создан мокап приложение который представлен на рисунке 5

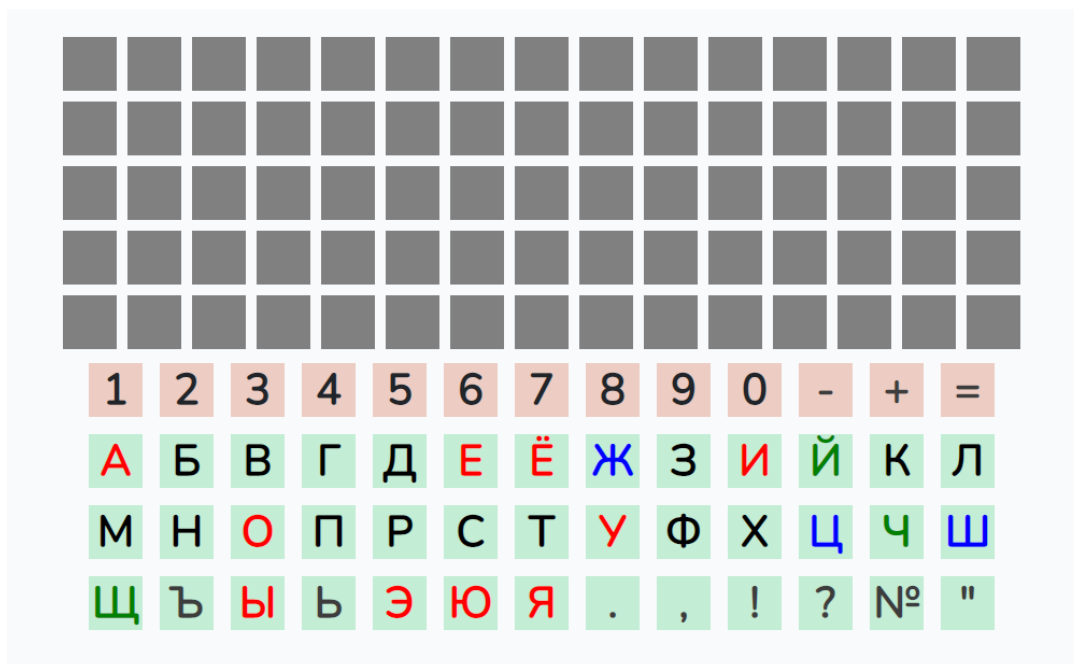


Рисунок 5 – Мокап веб-приложения

Case-модель базы данных веб-приложение [10], [13], [20] представлена на рисунке 6

Таблица users хранит в себе данные пользователей.

Атрибуты:

- id – идентификатор пользователя;
- name – имя пользователя;
- email – электронная почта пользователя;
- email\_verified\_at – дата подтверждение электронной почты пользователя;
- password – пароль пользователя;
- remember\_token – токен сессии;



- created\_at – дата регистрации пользователя;
- updated\_at – дата обновление данных;
- premium – дата окончание подписки на использование контента.

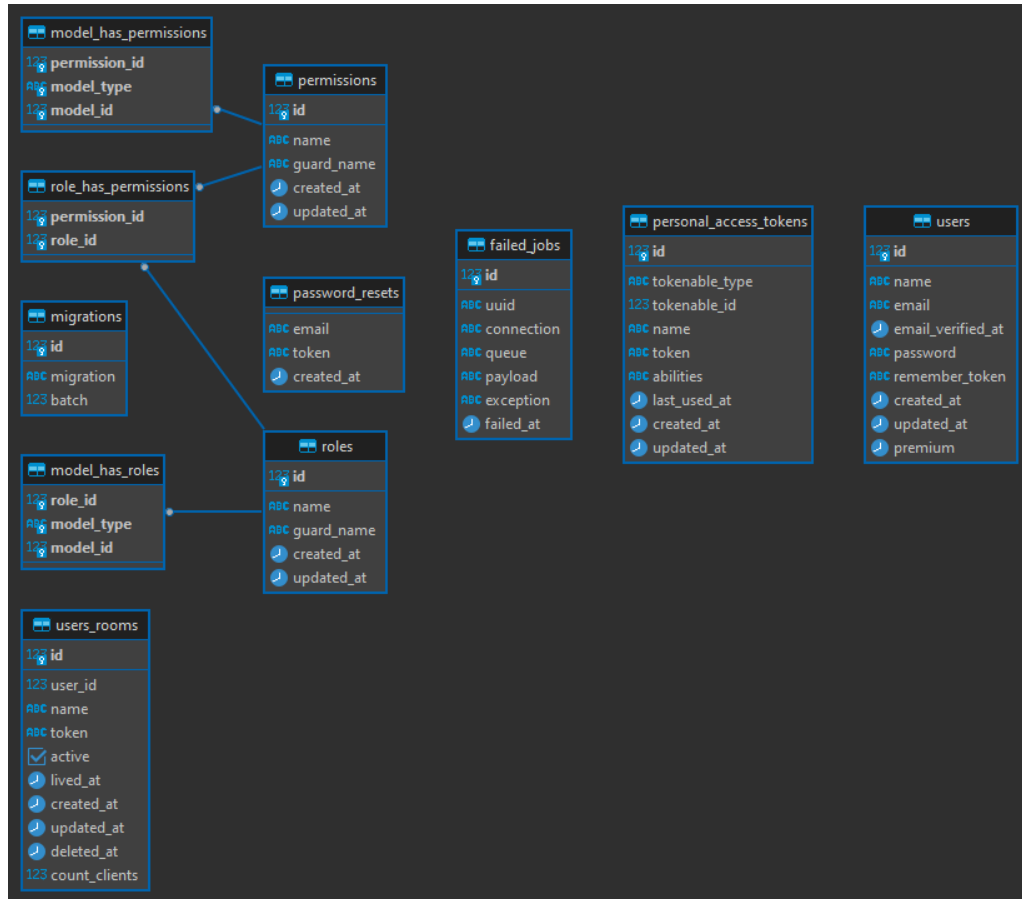


Рисунок 6 – Case-модель базы данных

Таблица users\_rooms хранит в себе комнаты пользователей.

Атрибуты:

- id – идентификатор комнаты;
- user\_id – идентификатор пользователя, которому принадлежит комната;
- name – название комнаты;
- token – токен комнаты для доступа;
- active – активность комнаты;
- lived\_at – дата активации комнаты;

- created\_at – дата создания комнаты;
- updated\_at – дата обновление комнаты;
- deleted\_at – дата удаление комнаты;
- count\_clients – максимальное количество клиентов в комнате.

Таблица roles хранит роли пользователей.

Атрибуты:

- id – идентификатор роли;
- name – название роли;
- guard\_name – название защитника laravel;
- created\_at – дата создание роли;
- updated\_at – дата обновление роли.

Таблица model\_has\_roles хранит связь роли с пользователем.

Атрибуты:

- role\_id – идентификатор роли;
- model\_type – тип модели;
- model\_id – идентификатор пользователя.

Таблица permissions хранит разрешения роли.

Атрибуты:

- id – идентификатор разрешение;
- name – название разрешение;
- guard\_name – название защитника laravel;
- created\_at – дата создание разрешение;
- updated\_at – дата обновление разрешение.

Таблица model\_has\_permissions хранит связь разрешение с пользователем.

Атрибуты:

- permission\_id – идентификатор разрешение;
- model\_type – тип модели;
- model\_id – идентификатор пользователя.

Таблица role\_has\_permissions хранит связь разрешение с ролью.

Атрибуты:

- permission\_id – идентификатор разрешение;
- role\_id – идентификатор роли.

Таблица password\_resets хранит токены сброса для пароля.

Атрибуты:

- email – электронная почта пользователя;
- token – токен для сброса пароля;
- created\_at – дата запроса сброса.

После создание мокапа была начата верстка веб-приложения показан на рисунке 7.

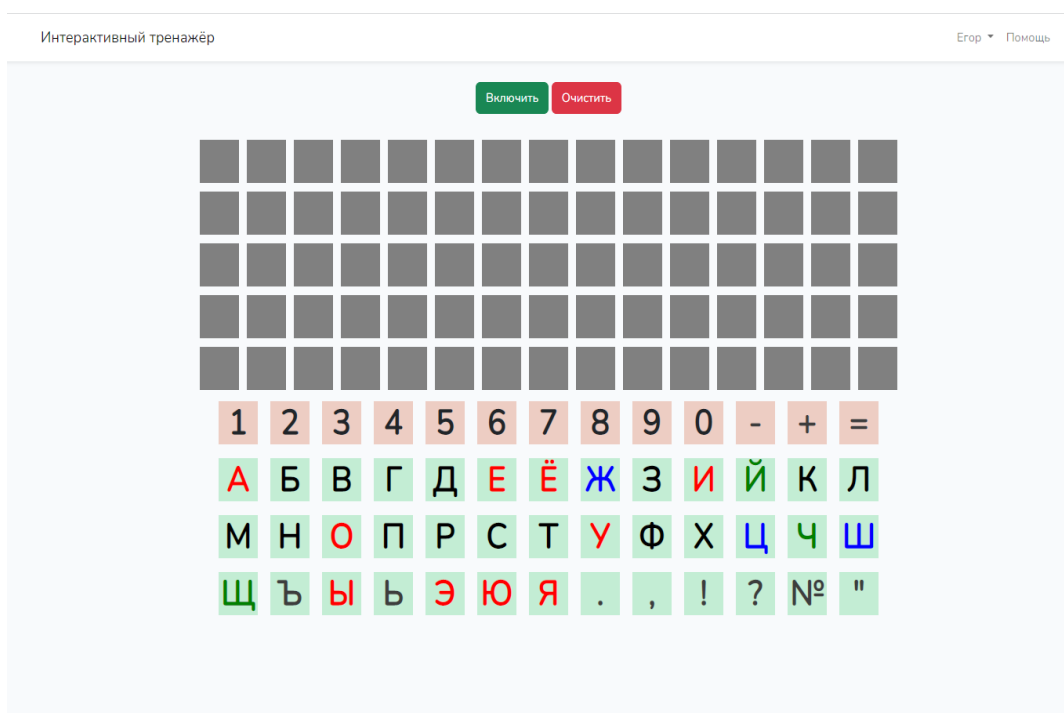


Рисунок 7 – Интерактивный тренажёр

Окно для обратной связи показано на рисунке 8.

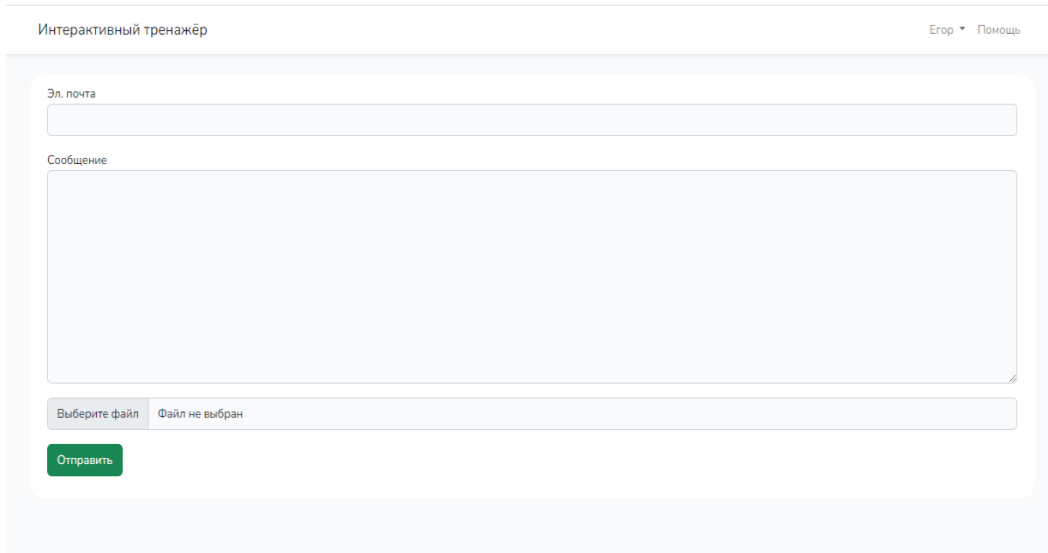


Рисунок 8 – Обратная связь

Код вёрстки интерактивного тренажёра [18] показан на рисунке 9.

```

1  @section('content')
2  @author($room->user_id)
3  @active($room->active)
4  <div id="left_menu.button" class="open"></div>
5  <div id="left_menu" class="close">
6  @yield('left_menu')
7  </div>
8  @endactive
9  @endauthor
10 <div class="container px-4 text-center element-center">
11 @author($room->user_id)
12 @active($room->active)
13 <input type="hidden" id="author_room" value="true">
14 <input type="hidden" id="author_name" value="{{Auth::user()->name}}">
15 @endactive
16 <div class="modal fade" id="ClearTrainerModal" tabindex="-1" aria-labelledby="ClearTrainerModal" aria-hidden="true">
17 <div class="modal-dialog">
18 <div class="modal-content">
19 <div class="modal-header">
20 <h5 class="modal-title" id="ClearTrainerModalLabel">Вы точно хотите очистить ?</h5>
21 <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
22 </div>
23 <div class="modal-footer">
24 <button type="button" class="btn btn-danger" data-bs-dismiss="modal">Нет</button>
25 <button type="button" class="btn btn-primary" data-bs-dismiss="modal" onclick="Clear()">Да</button>
26 </div>
27 </div>
28 </div>
29 </div>
30 <div class="buttons">
31 @can('public room')
32 <a href="{{route('active_user_room',$room->token)}}" class="btn btn-success">
33 @active($room->active)
34 @lang('room.inactive')
35 @else
36 @lang('room.active')
37 @endactive
38 </a>
39 @endcan
40 @can('management room')
41 <button class="btn btn-danger" data-bs-toggle="modal" data-bs-target="#ClearTrainerModal" @lang('room.clear')></button>
42 @endcan
43 </div>
44 @endauthor
45 @active($room->active)
46 <input type="hidden" id="token_room" value="{{ $room->token }}">
47 <input type="hidden" id="count_clients_room" value="{{ $room->count_clients }}">
48 <input type="hidden" id="active_room" value="{{ $room->active }}">
49 @endactive
50 <div id="trainer" oncontextmenu="return false;">
51 <div class="field">
52 @for ($i = 0; $i < 75; $i++)
53 <div class="pole" id="pole_{{ $i }}"></div>
54 @endfor
55 </div>
56 <div class="keyboard">
57 @foreach ($keyboard as $key)
58 <div class="noselect key_selected key_{{ $key['type'] }}">{{ $key['char'] }}</div>
59 @endforeach
60 </div>
61 </div>
62 </div>
63 <script src="{{ asset('js/room.js') }}"></script>
64 <script>
65 function Clear(){
66 $(".key_moved").remove();
67 $(".pole.busy").removeClass("busy");
68 if($("#active_room").val()){
69 socket.emit('clear_table', $("#token_room").val());
70 }
71 function ClickId{
72 console.log(id);
73 if($("#active_room").val()){
74 socket.emit('disconnect_client', id);
75 }
76 </script>
77 @endsection

```


Рисунок. 9 – Вёрстка тренажёра

Код для работы с socket.io [12], [15], [16], [17] показан на рисунке 10.

```
1 isAuthor = $('#author_room').val();
2 room = $('#token_room').val();
3 let name = null;
4 if(isAuthor){
5     name = $('#author_name').val();
6 }else{
7     name = prompt("Как вас зовут ?", "Ученик");
8     if(!name){
9         name = "Ученик"
10    }
11 }
12 window.socket = io(window.location.hostname + ':' + config.PORT);
13 socket.emit('create_room', room, isAuthor, $('#count_clients_room').val(), name);
14 socket.on('created_key', (message) => {
15     $('#<div class="" + message.class + ' key_remote' style="" + message.style + '>' + message.text + '</div>').appendTo('#trainer');
16 });
17
18 socket.on('busy_pole', (message) => {
19     $('#'+message.id).addClass('busy');
20 });
21
22 socket.on('unbusy_pole', (message) => {
23     $('#'+message.id).removeClass('busy');
24 });
25
26 socket.on('moved_key', (message) => {
27     let el = message.class;
28     while (el.indexOf(' ') != -1) {
29         el = el.replace(' ', '.');
30     }
31     el = el + ".key_remote"
32     $('#("." + el)[0]).attr('style', message.style);
33 });
34
35 socket.on('delete_key', (message) => {
36     let el = message.class;
37     while (el.indexOf(' ') != -1) {
38         el = el.replace(' ', '.');
39     }
40     el = el + ".key_remote"
41     $('#("." + el)[0]).remove();
42 });
43
44 socket.on('registr_key', (message) => {
45     let el = message.old_class;
46     while (el.indexOf(' ') != -1) {
47         el = el.replace(' ', '.');
48     }
49     el = el + ".key_remote"
50     $('#("." + el)[0]).attr('class', message.new_class + " key_remote");
51 });
52
53 socket.on('new_connect', (message) => {
54     alert("К комнате присоединился " +message.name);
55     $('#clients_list').append('<div class="client" id="'+message.id+'">' + message.name + '<button class="btn-cick" onclick="Click("'+message.id+'")\`
56 >></button></div>');
57 });
58
59 socket.on('new_disconnect', (message) => {
60     alert("От комнаты отсоединился " +message.name);
61     $('#'+message.id).remove();
62 });
63
64 socket.on('disconnect', (message) => {
65     alert("Вы отключены от комнаты");
66     location.reload();
67 });
68
69 socket.on('clear_table', () => {
70     $('#.key_moved').remove();
71     $('#.pole.busy').removeClass('busy');
72 });
```

Рисунок 10 – Функции сервера для синхронизации клиентов

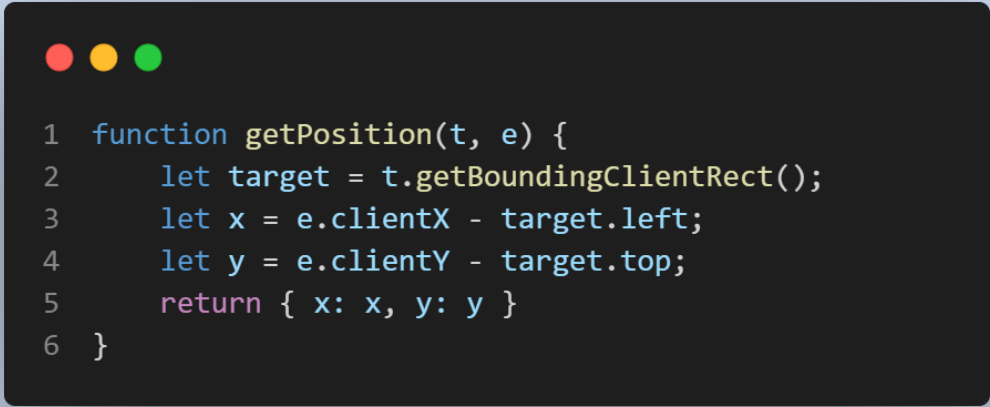
Функция отправки данных клиентам [15] показана на рисунке 11.



```
1 function SendMessage(event,message = null){
2     if(isPublic)
3         if(message)
4             socket.emit(event, room, message);
5     else
6         socket.emit(event, room);
7 }
```

Рисунок 11 – Функция отправки данных

Функция получение позиции буквы [12], [17] показана на рисунке 12.



```
1 function getPosition(t, e) {
2     let target = t.getBoundingClientRect();
3     let x = e.clientX - target.left;
4     let y = e.clientY - target.top;
5     return { x: x, y: y }
6 }
```

Рисунок 12 – Функция получение буквы

Функция получение поля для установки буквы [12] показана на рисунке

13.



```
1 function getPole(e, key = e.target) {
2   let temp = null;
3   $('.pole').each(function () {
4     let pole = $(this)[0];
5     if (
6       !$(pole).hasClass('busy') &&
7       (pole.offsetTop - pole.offsetHeight) < key.offsetTop &&
8       (pole.offsetTop + pole.offsetHeight) > key.offsetTop &&
9       (pole.offsetLeft - pole.offsetWidth) < key.offsetLeft &&
10      (pole.offsetLeft + pole.offsetWidth) > key.offsetLeft
11     )
12     {
13       temp = pole;
14     }
15   })
16   return temp;
17 }
```

Рисунок 13 – Получение поля для установки буквы

Получение стиля для отправки и подстановки в букву [17] показана на рисунке 14.



```
1 function getStyle(e, key = e.target, pole = null){
2   let style = null;
3   if(pole){
4     style = {
5       'z-index': '',
6       'left': (pole.offsetLeft - 8) + 'px',
7       'top': (pole.offsetTop - 8) + 'px'
8     }
9   }else{
10    let position = getPosition(document.getElementById('trainer'), e)
11    style = {
12      'position': 'absolute',
13      'z-index': 10,
14      'left': (position.x - (key.offsetWidth / 2)) + 'px',
15      'top': (position.y - (key.offsetHeight / 2)) + 'px'
16    }
17  }
18  return style
19 }
```

Рисунок 14 – Получение стиля буквы

Обработка нажатия на букву [12], [17] показана на рисунке 15.

A screenshot of a code editor with a dark background and light-colored text. The code is written in JavaScript and is numbered from 1 to 25. It shows a jQuery selector for elements with the class 'key\_selected'. A mousedown event listener is attached to these elements. Inside the event listener, there are several conditional checks and actions: removing the 'delete' class from a target element, checking if the target has the 'key\_selected' class, and if the key pressed is the spacebar (e.which == 1), cloning the element, adding a 'key\_moved' class with an incremented ID, and prepending it to the '#trainer' element. A second mousedown event listener is attached to the cloned element, which removes the 'delete' class when the delete key is pressed (e.which == 3). Finally, a message object is created with the class, style, and text attributes of the cloned element, and a 'created\_key' message is sent with this object. The cloned element is then stored in a variable 'key\_move' at index 0.

```
1 $(".key_selected")
2   .mousedown(function (e) {
3     $('.key.delete').removeClass('delete');
4     if ($(e.target).hasClass('key_selected')) {
5       if (e.which == 1) {
6         _move = true;
7         key_move = $(this).clone()
8           .addClass('key_moved ' + (key_ids = key_ids + 1))
9           .css(getStyle(e))
10          .prependTo("#trainer")
11          .mousedown(function (e) {
12            if(e.which == 3){
13              $('.key.delete').removeClass('delete');
14            }
15          })
16          let message = {
17            'class': $(key_move).attr('class'),
18            'style': $(key_move).attr('style'),
19            'text': $(key_move).text()
20          }
21          SendMessage('created_key',message)
22          key_move = $(key_move)[0]
23        }
24      }
25    })
```

Рисунок 15 – Обработка нажатие на букву

Обработка нажатий в тренере [12], [17] показана на рисунке 16.



```

1 $('#trainer')
2   .mousedown(function (e) {
3     if ($(e.target).hasClass('key') && !$(e.target).hasClass('key_remote')){
4       if (e.which == 1) {
5         _move = true;
6         if ($(e.target).hasClass('key_moved')){
7           key_move = e.target;
8           $('#'+ $(key_move).data('id')).removeClass('busy');
9           let message = {
10            'id': $(key_move).data('id'),
11          }
12          SendMessage('unbusy_pole', message)
13          if ($(key_move).hasClass('delete')) {
14            $(key_move).removeClass('delete');
15            $(key_move).remove();
16            let message = {
17              'class': $(key_move).attr('class'),
18            }
19            SendMessage('delete_key', message)
20          }
21          $('key.delete').removeClass('delete');
22          $(key_move).addClass('delete')
23        }
24      } else if (e.which == 3) {
25        let message = {
26          'old_class': $(e.target).attr('class'),
27          'new_class': null
28        }
29        if ($(e.target).hasClass('lowercase'))
30          $(e.target).removeClass('lowercase');
31        else
32          $(e.target).addClass('lowercase');
33        message.new_class = $(e.target).attr('class');
34        if ($(e.target).hasClass('key_moved'))
35          SendMessage('registr_key', message)
36      }
37    }
38  })
39  .mousemove(function (e) {
40    if (_move && key_move && e.which == 1) {
41      $(key_move).css(getStyle(e, key_move))
42      let message = {
43        'class': $(key_move).attr('class'),
44        'style': $(key_move).attr('style'),
45      }
46      SendMessage('moved_key', message)
47      if ($(key_move).hasClass('delete'))
48        $(key_move).removeClass('delete')
49    }
50  })
51  .mouseup(function (e) {
52    if ($(e.target).hasClass('key_moved') && key_move)
53      if (e.which == 1) {
54        _move = false;
55        let pole = getPole(e, key_move);
56        if(pole){
57          $(key_move).css(getStyle(e, key_move, pole)).data("id",$(pole).attr('id'));
58          let message = {
59            'class': $(key_move).attr('class'),
60            'style': $(key_move).attr('style'),
61          }
62          SendMessage('moved_key', message)
63
64          message = {
65            'id': $(pole).attr('id'),
66          }
67          $(pole).addClass('busy')
68          SendMessage('busy_pole', message)
69        }else{
70          $(key_move).remove();
71          let message = {
72            'class': $(key_move).attr('class'),
73          }
74          SendMessage('delete_key', message)
75        }
76        key_move = null;
77      }
78    }
79  });

```

Рисунок 16 – Обработка нажатий в тренере

Стили тренера [19] показаны на рисунке 17.



```
1 #trainer {
2     position: relative;
3     display: flex;
4     flex-direction: column;
5     align-items: center;
6     margin: 25px;
7     width: var(--trainer-width);
8 }
9
10 #trainer .field {
11     display: flex;
12     flex-wrap: wrap;
13     width: var(--field-width);
14 }
15
16 #trainer .field .pole {
17     margin: var(--pole-margin);
18     width: var(--pole-size);
19     height: var(--pole-size);
20     background-color: var(--pole-background);
21 }
22
23 #trainer .keyboard {
24     display: flex;
25     flex-wrap: wrap;
26     justify-content: center;
27     width: var(--keyboard-width);
28 }
29
30 #trainer .key {
31     margin: var(--key-margin);
32     width: var(--key-size);
33     height: var(--key-size);
34     background-color: var(--key-background);
35     display: flex;
36     justify-content: center;
37     align-items: center;
38     font-size: var(--key-font-size);
39     font-weight: bold;
40     cursor: pointer;
41 }
```

Рисунок 17 – Стили для тренера

Код верстки обратной связи [18] показан на рисунке 18.

```
1 @section('content')
2   <div class=
3     "container px-4 text-center help-box">
4     @if(session('message'))
5       <div class="alert alert-success">
6         <p>@lang(session('message'))</p>
7       </div>
8     @endif
9     <x-form-help />
10  </div>
11 @endsection
```

Рисунок 18 – Обратная связь

Код формы помощи [18] показан на рисунке 19.

```
1 <form action="{{route('help-form')}}" method="post" enctype="multipart/form-data">
2   @csrf
3   <div class='mb-3 text-left'>
4     <label for="email">@lang('help.email')</label>
5     <input class='form-control @error("email") form-error @enderror' id="email" name="email" type=
6       "email" value="{{old('email') ? old('email') : (Auth::user() ? Auth::user()->email : null)}}"/>
7     @error('email') {{ __($message) }} @enderror
8   </div>
9   <div class='mb-3 text-left'>
10    <label for="message">@lang('help.message')</label>
11    <textarea class="form-control @error('message') form-error @enderror" id="message" name="message"
12      cols="30" rows="10">{{old('message') ?? null}}</textarea>
13    @error('message') {{ __($message) }} @enderror
14  </div>
15  <div class='mb-3 text-left'>
16    <input class="form-control" type="file" name="attachment" />
17  </div>
18  <div class='mb-3 text-left'>
19    <button class="btn btn-success" type="submit">Отправить</button>
20  </div>
21 </form>
```

Рисунок 19 – Код формы обратной связи

Функция отправки сообщения [5], [6], [7], [9], [14], [20], [21] показан на рисунке 20.

```
1 public function send(HelperRequest $request){
2     $validate = $request->safe()->all();
3     Mail::to(config('mail.from.address'))
4         ->send(new HelpSend(
5         $validate['email'],
6         $validate['message'],
7         $validate['attachment'] ?? null
8     )
9 );
10     return back()->with('message', 'help.wait_response');
11 }
```

Рисунок 20 – Отправка сообщения

Функция активации комнаты [3], [4], [20], [21] показана на рисунке 21.

```
1 public function active_room(string $token)
2     {
3         $user = Auth::user();
4         $room = Room::where('token', $token)->first();
5         if(!$user && !$room) return abort('404', __('room.error'));
6         if($user && $user->id != $room->user_id) return abort('403', __('room.not_permission'));
7         $room->toggle(!$room->active);
8         return redirect()->route('user_room', $room->token);
9     }
```

Рисунок 21 – Активация комнаты

Функция отображение комнаты [4], [8], [10], [11], [14], [20], [21] показана на рисунке 22.

```

1 public function room(string $token){
2     $keyboard = $this::keyboard;
3     $user = Auth::user();
4     $room = Room::where('token',$token)->first();
5     if(!$room) return abort(404,__('room.not_found'));
6     if(!$room->active && (!$user || $user && $room->user_id != $user->id)) return abort(403, __('room.not_permission'));
7     return view('room.index',compact('room','keyboard'));
8 }

```

Рисунок 22 – Отображение комнаты

### 2.3 Тестирование веб-приложения

Тестирование приложения показано на таблице 1.

Таблица 1 – Тестирование приложения

Входные данные	Выходные данные		Вывод
Перемещение букв	Эталон	Факт	
Позитивное тестирование			
Переместить букву в ячейку	Оказался в ячейке	Оказался в ячейке	пройдено
Негативное тестирование			
Переместить букву за пределы поля	буква исчезла	буква исчезла	пройдено
Тестирование отправки сообщения помощи			
Позитивное тестирование			
Отправка сообщение с введенными данными	сообщение отправлено	сообщение отправлено	пройдено
Негативное тестирование			
Отправка пустого сообщение	сообщение о заполнении полей	сообщение о заполнении полей	пройдено
Отключение клиента от комнаты			
Отключение клиента	клиент отключен	клиент переподключился	Не пройдено

## 2.4 Публикация веб приложения

Создание поддомена <http://education.dogit-studios.ru>, и загрузка проекта на сервер с использованием FTP протокола. Созданный поддомен и загруженный проект продемонстрирован на рисунке 23.

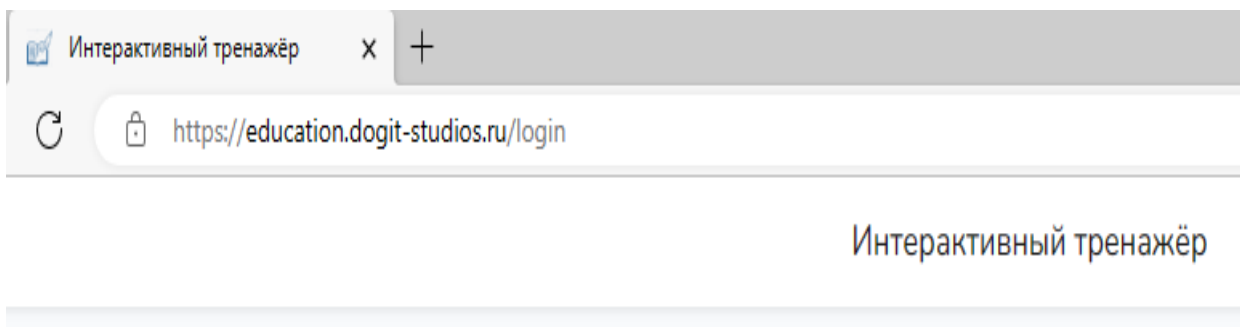


Рисунок 23 – Проект загружен

Для использования тренажёра необходимо зарегистрироваться показано на рисунке 24.

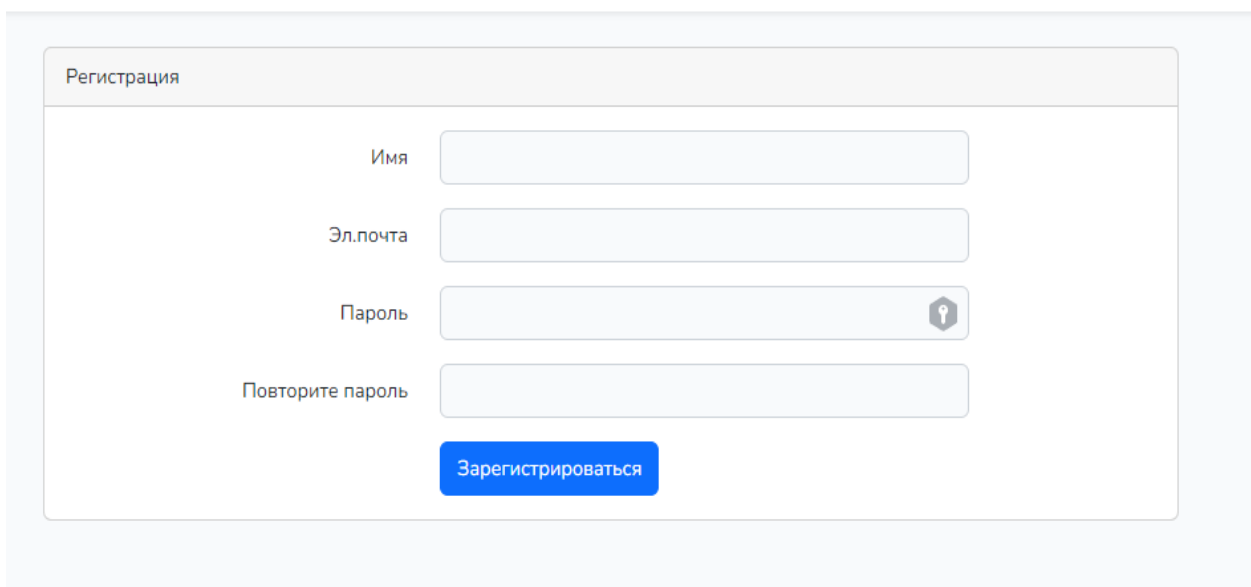
A screenshot of a registration form titled "Регистрация". The form is contained within a light gray border. It features four input fields stacked vertically, each with a label to its left: "Имя", "Эл.почта", "Пароль", and "Повторите пароль". The "Пароль" field includes a small eye icon on the right side. Below the input fields is a blue button with the text "Зарегистрироваться" in white.

Рисунок 24 – Страница регистрации

## Выводы по Главе 2

Таким образом, был разработан структура веб-приложения, case-модель базы данных, Моцир дизайн веб-приложения, верстка веб-приложения по созданному шаблону с прорисовкой деталей и эффектов, создан файл стилей для веб-приложения, подключена база данных для сохранения информации веб-приложения, произведена публикация веб-приложения в интернете.

В результате проделанной работы был создан интерактивный тренажёр для обучения учеников дошкольного и школьного возраста.

Работа имеет продолжение: можно добавить конструктор тренажёров и реализовать разнovidное управление тренажёрами.

## **Глава 3 Расчет затрат**

### **3.1 Экономическое и организационное обоснование проекта**

Одна из основных проблем обучения в школе была и остается мотивация. Использование компьютерных программ для обучения – одна из возможностей повысить интереса к получению знаний. Внедрение интерактивных тренажеров делает процесс обучения интереснее, легче. Повышается мотивация ученика к обучению. При использовании компьютерных обучающих приложений пользователи приобретают новые навыки и компетенции.

Данный проект обеспечивает получение постоянной обратной связи от учителя к ученику, что способствует вносить поправки в действия ученика. Возможные покупатели – учебные заведения, родители, педагоги.

Рынок компьютерных обучающих приложений предлагает продукты для разных ЦА (целевых аудиторий).

Целевая аудитория: студенты (высшее, среднее образование); репетиторы и дети; родители и дети; педагоги и ученики; языковая аудитория (изучающие языки, носители языка); работники разных отраслей.

Для начального этапа разработки был проанализирован рынок. Определена целевая аудитория. Изучены запросы и пожелания заказывающей стороны были выбраны языки программирования и произведен расчет себестоимости продукта.

### **3.2 Расчет себестоимости**

#### **3.2.1 Определение трудоемкости**

Определение сколько дней заняла разработка проекта показана на таблице 2.



Таблица 2 – Трудоемкость

Виды работ	Трудоемкость, дн.
Получение ТЗ	1
Сбор информации и ознакомление с предметной областью	2
Выбор объектного построения программы	2
Разработка общей методики создания продукта	2
Разработка основного алгоритма	3
Создание интерфейса	2
Отладка	2
Подготовка технической документации	1
Сдача продукта	1
Итого	16

### 3.2.2 Расчет затрат на материалы

Калькуляция стоимости основных материалов показана в таблице 3.

Таблица 3 – Основные материалы

Наименование	Кол-во, шт.	Цена ед., руб.	Сумма, руб.
laravel	1	0	0
хостинг	1	0	0
поддомен	1	0	0
Итого			0

### 3.2.3 Расчёт затрат на электроэнергию и амортизации оборудования

Амортизации оборудование:

- Ноутбук: 500 рублей;
- Raspberry PI 4B: 100 рублей.

Затраты: электричество: 256 рублей.

### 3.2.4 Расчет расходов на заработанную плату

Стоимость работы 5.000 рублей.

### 3.2.5 Расчет начислений на заработанную плату

Начисления на заработную плату показаны на таблице 4.

Таблица 4 – Начисление на заработанную плату

Начисления на заработную плату	Процент, %	Сумма, руб.
Пенсионный фонд (ПФ):		
– страховая часть	16	800
– накопительная часть	6	300
Фонд социального страхования (ФСС)	2,9	145
Федеральный фонд обязательного медицинского страхования (ФФОМС)	5,1	255
Итого	30	1500

### 3.2.6 Расчет себестоимости и цены разработки

Смета всех затрат показана на таблице 5.

Таблица 5 – Смета всех затрат

Наименование статей затрат	Сумма, руб.
Амортизация оборудование	600
электричество	256
заработная плата	1500
Итого	2356

### 3.2.7 Расчет прибыли

Предполагаемая прибыль:  $5000 - 2356 = 2644$ .

### 3.2.8 Расчёт показателей эффективности

На основе прибыли рассчитываются показатели:

рентабельность общая рассчитывается как отношение прибыли к среднегодовой стоимости затрат:

Рентабельность:  $2644 / 2356 \cdot 100 \% = 11,2\%$ .

## Глава 4 Безопасность жизнедеятельности и охрана труда

При использовании любой бытовой техники необходимо соблюдать правила безопасности, так как работа всех приборов связана с потреблением электроэнергии.

При работе с ПК так же нужно соблюдать ряд мер. В первую очередь идет огромная нагрузка на зрение. Не правильная посадка за компьютером ведет к нарушению осанки, вызывает заболевание шейного отдела позвоночника. Очень долгое сидение за компьютером приводит к сильному эмоциональному напряжению, что вызывает расстройство психики. В последствии это очень тяжело поддается лечению. Для того чтобы избежать негативного влияния компьютера необходимо соблюдать требования при работе за ПК

Техника безопасности при работе с компьютером едина для всех организаций. Эти правила изложены в СанПиНе. На рисунке 25 представлены свод основных правил при посадке за компьютером.

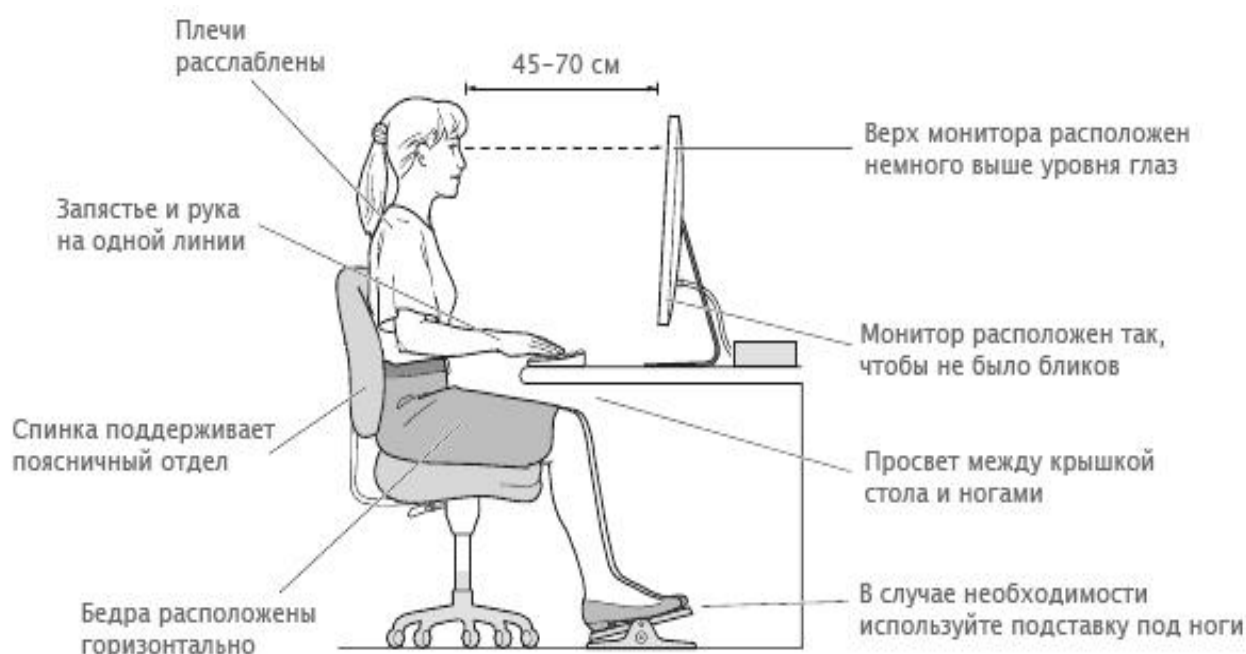


Рисунок 25 – Правильная посадка за компьютером

Вышеуказанные нормы установлены в СанПиНе.

Далее, для правильной работы рук используются эргономичные клавиатуры. Для того чтобы повысить безопасность при работе за компьютером для глаз надо:

- современный монитор с хорошим разрешением;
- правильная осанка;
- необходимость выполнения ежечасной гимнастики для глаз.

Кроме этого, необходимо производить влажную уборку в помещении, проветривания и оборудование помещения вентиляцией и системами кондиционирования воздуха.

СанПиН регламентирует следующие требования к атмосферным показателям в помещениях, оборудованных рабочими местами с ПК:

- температура воздуха – от 19 до 21 градусов;
- влажность – от 55 до 62 %;
- скорость движения воздуха – не больше 0,1 м/с.

Режим работы и перерывы.

Рекомендации:

- не сидеть долго за компьютером;
- через 40-55 минут перерыв 10-15 минут;
- выполнять комплекс упражнений для глаз;
- выполнять физические упражнения.

## Заключение

Подводя итоги работы, важно отметить, что все поставленные задачи были решены:

- разработано техническое задание на проектирование обучающего тренажера;

- изучены аналогичны решения и аналоги разработанного обучающего интерактивный тренажера, проведен сравнительный анализ по выявленным критериям;

- в программе Figma осуществлен дизайн модуля приложения обучающего интерактивного тренажера;

- разработана клиентская часть JS;

- разработан socket.io сервера на Node.JS;

- произведено тестирование программного продукта приложения обучающего интерактивного тренажера, показавшее его работоспособность;

- осуществлен выпуск программного продукта – приложения обучающего интерактивного тренажера.

Таким образом, в результате была достигнута основная цель, поставленная в выпускной квалификационной работе: разработан интерактивный тренажер «Читай, Считай, Пиши»

Изученный в процессе написания ВКР материал по созданию web приложений обладает определёнными достоинствами: легкость в использовании, и недостатками: несоответствие выбранному функционалу.

При использовании ряда сервисов (Figma, Docker, VS Code, Google Chrome) была разработана структура веб-приложения, case-модель базы данных, Мосир дизайн веб-приложения, верстка веб-приложения по созданному шаблону с прорисовкой деталей и эффектов, создан файл стилей для веб-приложения, подключена база данных для сохранения информации веб-приложения, произведена публикация веб-приложения в интернете.

В результате проведённой работы по разработке веб-приложение для школы №1, были выполнены следующие задачи:

- изучение аналогов;
- создания дизайна веб-приложения в программе Figma;
- верстка дизайна с применением языков разметки HTML и каскадных таблиц стилей CSS;
- разработка клиентской части JS;
- разработка серверной части PHP;
- разработка сервера для обмена информацией между пользователями Node.JS;
- тестирование программного продукта.

Данное веб-приложение создавалось для обучение начальных классов письму и чтению в дистанционном режиме.

Работа имеет продолжение: можно добавить конструктор тренажёров и реализовать разнovidное управление тренажёрами.

## Список используемой литературы

1. Приказ «О деятельности подведомственных Министерству науки и высшего образования Российской Федерации организаций в условиях предупреждения распространения новой коронавирусной инфекции (COVID-19) на территории Российской Федерации» от 28.05.20 г.
2. Распоряжение «Об утверждении методических рекомендаций для внедрения в основные общеобразовательные программы современных цифровых технологий» от 18.05.20 г. N P-44
3. Роберт Мартин. Идеальный программист. Как стать профессионалом разработки ПО / 2022 – 224 с.
4. Роберт Мартин. Чистая архитектура. Искусство разработки программного обеспечения / 2022. – 352 с.
5. Adel F. Architecture of complex web applications With examples in Laravel(PHP) / 2019. – 64 pages.
6. Alan Forbes. The Joy of PHP / 2020. – 128 pages.
7. Joseph Edmonds. The Art of Modern PHP 8 / 2021. – 421 pages.
8. Kevin Tatroe & Peter MacIntyre Foreword by Michael Stowe. Programming PHP Creating Dynamic Web Pages / 2020. – 542 pages.
9. Kristopher Wilson. The Clean Architecture in PHP / 2016. – 370 pages.
10. Sanjib Sinha. Beginning Laravel. 2nd Ed / 2019. – 429 pages.
11. Matt Stauffer. Laravel: Up & Running / 2019 – 544 pages.
12. Документация по jQuery [Электронный ресурс] URL: <https://api.jquery.com/> (дата обращения: 05.05.2023).
13. Документация laravel [Электронный ресурс] URL: <https://laravel.su/docs/8.x/installation> (дата обращения: 05.05.2023).
14. Документация по php [Электронный ресурс] URL: <https://php.ru/manual/> (дата обращения: 05.05.2023).
15. Документация по работе с библиотекой socket.io [Электронный ресурс] URL: <https://socket.io/docs/v4/> (дата обращения: 05.05.2023).

16. Документация по node.js [Электронный ресурс] URL: <https://nodejs.org/ru/docs/> (дата обращения: 05.05.2023).
17. Документация по javascript [Электронный ресурс] URL: <https://javascript.ru/manual/> (дата обращения: 05.05.2023).
18. Документация по html [Электронный ресурс] URL: <http://htmlbook.ru/html/> (дата обращения: 05.05.2023).
19. Документация по css [Электронный ресурс] URL: <http://htmlbook.ru/css/> (дата обращения: 05.05.2023).
20. Официальная документация laravel [Электронный ресурс] URL: <https://laravel.com/docs/9.x> (дата обращения: 05.05.2023).
21. Официальная документация php [Электронный ресурс] URL: <https://www.php.net/> (дата обращения: 05.05.2023).