Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Новгородский государственный университет имени Ярослава Мудрого»

ИНСТИТУТ ПОЛИТЕХНИЧЕСКИЙ

ПОЛИТЕХНИЧЕСКИЙ КОЛЛЕДЖ

УТВЕРЖДАЮ

Зам. директора по УМ и ВР

Л.Н. Иванова

« » 2025 г.

РАЗРАБОТКА ВЕБ-СЕРВИСА ДЛЯ ТРЕНИРОВКИ СЛЕПОЙ ПЕЧАТИ В ИГРОВОЙ ФОРМЕ

Пояснительная записка к дипломному проекту по специальности

09.02.07 Информационные системы и программирование

ПТК. ДП 2994 58. 000ПЗ

|  |  |
| --- | --- |
| Согласовано: |  |
| Консультант по спец. части  Л.Н. Цымбалюк  « » 2025 года  Консультант по экон. части  Г.В. Лебедева  « » 2025 года  Нормоконтроль  А.М. Чернега  « » 2025 года  Заместитель директора по УПР  А.М. Чернега  « » 2025 года | Руководитель  В.В. Бурбах  « » 2025 года  Выполнил:  обучающийся группы 2994  В.В. Иванов  « » 2025 года |

Содержание

[Введение 3](#_Toc194876158)

[1 Общая часть 5](#_Toc194876159)

[1.1 Постановка задачи 5](#_Toc194876160)

[1.2 Обоснование проектных решений 9](#_Toc194876161)

[1.3 Обзор и анализ существующих программных систем 14](#_Toc194876162)

[Список литературы 16](#_Toc194876163)

# Введение

Освоение новых навыков – то, что помогает расти как в личном, так и в профессиональном плане. Если раньше работа с компьютером требовала специальных знаний и усилий, то сегодня использование цифровых технологий стало повседневной рутиной. Современные технологии прочно вошли в нашу жизнь, и умение эффективно ими пользоваться превратилось в обязательный навык. Одним из таких навыков является слепая печать — способность быстро набирать текст, не глядя на клавиатуру. Этот навык экономит время и улучшает сосредоточенность на содержании работы, что важно для тех, кто ежедневно работает с текстами.

Для людей, которые проводят за компьютером большую часть дня, владение слепой печатью становится практически необходимым. Быстрый набор текста без необходимости постоянно смотреть на клавиатуру помогает сосредоточиться на задачах, а не на технических деталях. Однако процесс освоения этого навыка часто оказывается скучным и однообразным. Традиционные методы, такие как повторение упражнений или использование специализированных тренажеров, не всегда способны удержать интерес пользователя. В результате большинство пользователей бросают тренировки, так и не достигнув желаемого уровня.

Главная трудность в освоении слепой печати – это монотонность, которая может быстро вызывать у ученика уныние и снижать интерес к процессу. Решением этой проблемы может статьгеймификация – использование игровых приёмов в любом неигровом контексте. Геймификация способна превратить рутинные упражнения в захватывающий опыт, где каждый шаг приносит не только новые навыки, но и удовольствие.

В данной работе предлагается подход к обучению слепой печати через использование веб-сервиса, который объединяет образовательный процесс с игровыми элементами. Пользователь управляет персонажем, вводя символы на клавиатуре, и получает очки за правильные действия. Такой формат делает тренировки вовлекающими и даёт увидеть прогресс непосредственно в процессе.

Проект рассчитан на аудиторию школьников, студентов, офисных работников и всех, кто заинтересован в повышении эффективности при работе с текстами. Для школьников и студентов тренажер может стать полезным инструментом для развития навыков, которые пригодятся в учебе. Для офисных работников освоение слепой печати позволит ускорить выполнение повседневных задач.

Цель проекта – создание интерактивного веб-сервиса для освоения слепой печати, использующего элементы геймификации для повышения мотивации пользователей.

Объект исследования – веб-сервис для тренировки слепой печати с игровыми элементами «Typing Quest».

Предмет исследования – процесс разработки эффективной системы тренировки слепой печати через игровую механику, с повышением вовлеченности пользователей в процесс освоения навыка.

Задачами данной работы, являются:

* изучить существующие методы тренировки слепой печати, их преимущества и недостатки;
* разработать концепцию веб-сервиса, сочетающего обучение и игровые элементы;
* реализовать функционал тренажера с внедрением игровых элементов.

1. Общая часть
2. Постановка задачи
3. Обоснование необходимости разработки

В соответствии с заданием на выпускную квалификационную работу требуется разработать веб-сервис для обучения слепой печати с элементами игровой механики. Актуальность разработки обусловлена возрастающей потребностью в эффективном освоении навыка слепой печати, который позволяет:

* значительно повысить продуктивность работы с текстами;
* снизить нагрузку на зрение за счет отсутствия необходимости переключать внимание между клавиатурой и экраном;
* минимизировать количество опечаток при наборе.

Традиционные методы обучения слепой печати часто обладают недостаточной вовлекающей составляющей, что приводит к снижению мотивации пользователей. Предлагаемое решение интегрирует обучающий процесс в игровую среду, что способствует:

* повышению вовлеченности за счет визуализации прогресса;
* формированию устойчивой мотивации через систему постепенного усложнения заданий;
* созданию положительного эмоционального фона обучения.

Проект ориентирован на широкий круг пользователей, которым необходимо освоить или улучшить навык слепой печати:

* офисные сотрудники и IT-специалисты, работающие с большими объёмами текста;
* студенты и школьники, желающие повысить эффективность работы за компьютером;
* все, кто заинтересован в развитии полезного навыка для профессионального или личного использования.

Сервис рассчитан на пользователей любого возраста с базовыми навыками работы на компьютере. Игровая составляющая реализована как дополнительный мотивационный инструмент, а не как основная функция, что отличает проект от чисто игровых продуктов.

1. Технико-математическое описание задачи

Разрабатываемый веб-сервис представляет собой комплексное решение, сочетающее современные веб-технологии и математические методы для эффективного обучения слепой печати.

Технологии разработки:

* протокол HTTP для клиент-серверного взаимодействия;
* язык программирования TypeScript для строгой типизации кода;
* реляционная система управления базой данных PostgreSQL для хранения данных;
* REST API архитектура серверной части.

Математические модели:

* расчет скорости печати в словах за минуту;
* вычисление процентного соотношения правильно набранных символов;
* анализ равномерности временных интервалов между нажатиями клавиш.

Игровые механики:

* зависимость скорости перемещения персонажа от скорости набора текста;
* создание препятствий, соответствующих допущенным ошибкам;
* система бонусов за достижение целевых показателей;
* система отображения рейтинга в таблице лидеров.

Методы обработки данных:

* статистический анализ результатов тренировок;
* формирование и обновление таблицы лидеров;
* визуализация прогресса обучения.

1. Характеристика бизнес-процессов

Группы пользователей и их функционал:

* неавторизованные пользователи:

1. просмотр информационной страницы о проекте;
2. регистрация в системе;
3. просмотр таблицы лидеров.

* авторизованные пользователи:

1. полноценное прохождение уровней;
2. просмотр личной статистики;
3. участие в рейтинговой системе;
4. настройка профиля;
5. получение достижений.

* администратор:

1. управление контентом;
2. анализ пользовательской активности;

Ключевым процессом является обучающий цикл, который включает последовательное прохождение уровней с постепенно возрастающей сложностью. Каждый уровень разбит на несколько этапов, что позволяет плавно наращивать навыки печати. Система предоставляет мгновенную обратную связь – визуализацию правильности набора, скорость печати и количество допущенных ошибок. По завершении упражнения формируется детальный отчет с анализом результатов.

Важным элементом является мотивационный процесс, включающий систему достижений за выполнение определенных нормативов, обновляемую таблицу лидеров, где пользователи могут видеть свои позиции относительно других участников, а также возможность сравнивать свои текущие результаты с предыдущими попытками.

1. Требования к программе

Для клиентской части веб-интерфейса необходимо наличие современного веб-браузера последних версий, включая Chrome, Firefox, Edge или Safari. Минимальное разрешение экрана должно составлять 1280 на 720 пикселей. Для стабильной работы требуется интернет-соединение со скоростью не менее 5 Мбит/с.

Серверная часть требует процессора с двумя или более ядрами и тактовой частотой от 2.4 ГГц. Минимальный объем оперативной памяти составляет 4 ГБ, а дискового пространства – 20 ГБ, при этом рекомендуется использовать SSD-накопители. Сетевой интерфейс должен поддерживать скорость передачи данных от 100 Мбит/с.

Клиентская часть требует поддержки JavaScript стандарта ES6 и выше, а также разрешения на использование cookies в браузере.

Для серверной части необходима операционная система Linux Ubuntu. Система использует систему управления базой данных PostgreSQL, среду выполнения Node.js и веб-сервер Nginx.

Фронтенд разработан на фреймворке Next.js с использованием игрового веб-движка Phaser. Бэкенд реализован на NestJS, для работы с базой данных применяется TypeORM.

Приложение представляет собой веб-решение, не требующее установки на клиентские устройства. Доступ осуществляется через стандартные веб-браузеры.

Обязательно наличие SSL-сертификата для обеспечения безопасного HTTPS-соединения. Рекомендуется настроить ежедневное резервное копирование данных и систему мониторинга нагрузки, включая контроль использования процессора, оперативной памяти и сетевого трафика.

1. Обоснование проектных решений
2. Обоснование выбора языков программирования

Основным языком разработки проекта выбран TypeScript, который сочетает преимущества JavaScript с дополнительной надежностью за счет строгой типизации. Этот выбор особенно важен для тренажера слепой печати, где необходимо точно обрабатывать пользовательский ввод и взаимодействовать с различными компонентами системы. Статическая типизация позволяет на этапе компиляции обнаруживать потенциальные ошибки взаимодействия с DOM-элементами и обработки пользовательского ввода, что критически важно для системы обучения слепой печати.

HTML выбран в качестве основы для построения пользовательского интерфейса благодаря своей универсальности и широкой поддержке всеми современными веб-браузерами. Этот язык разметки обеспечивает четкую структуру и логическую организацию контента на страницах тренажера, включая интерфейс статистики, профиля пользователя и информационные разделы. HTML позволяет корректно отображать текстовое содержимое, формы ввода данных и другие элементы интерфейса.

Для стилизации интерфейса используется CSS через фреймворк TailwindCSS, который был выбран за свою практичность и эффективность. TailwindCSS предоставляет готовые утилитарные классы, позволяя быстро создавать адаптивные интерфейсы без необходимости написания собственных стилей. Это значительно ускоряет процесс разработки и обеспечивает единообразие визуального оформления всех компонентов системы. Особенно важно, что TailwindCSS хорошо интегрируется с компонентным подходом, используемым в Next.js, сохраняя при этом высокую производительность.

Такой подход к выбору языков программирования и разметки позволяет создать устойчивую основу для дальнейшего развития тренажера, сохраняя баланс между производительностью и удобством разработки.

1. Инструментальные средства

Для сборки проекта и управления зависимостями применяется npm –Node Package Manager. Этот инструмент позволяет эффективно работать с более чем 40 зависимостями проекта, включая основные фреймворки и вспомогательные библиотеки. npm обеспечивает выполнение скриптов сборки, тестирования и развертывания. В качестве сборщика фронтенда используется Vite, который значительно ускоряет процесс разработки благодаря мгновенному обновлению модулей и оптимизированной сборке.

Next.js выступает основным фреймворком для фронтенд-разработки, предоставляя серверный рендеринг, статическую генерацию страниц и встроенную маршрутизацию. Для серверной части выбран NestJS, который предлагает модульную архитектуру, систему внедрения зависимостей и готовые решения для построения REST API. Игровая механика реализована с помощью Phaser – специализированного движка для создания браузерных игр. Стилизация интерфейсов выполнена с использованием Tailwind CSS, что позволяет быстро создавать адаптивные компоненты с минимальным объемом кастомного CSS.

Тестирование API осуществляется с помощью Postman – мощного инструмента для работы с HTTP-запросами. Postman позволяет не только отправлять тестовые запросы и проверять ответы сервера, но и сохранять коллекции запросов, автоматизировать рутинные проверки и документировать endpoints API. Это значительно упрощает процесс отладки и валидации серверной части приложения.

Проект предусматривает возможность развертывания с использованием Docker и Nginx. Docker обеспечивает контейнеризацию приложения, создавая изолированную среду со всеми необходимыми зависимостями. Nginx может быть использован в качестве обратного прокси и веб-сервера, оптимизируя обработку запросов и обслуживание статических ресурсов.

Выбранный набор инструментов обеспечивает комплексный подход к разработке, охватывая все этапы – от проектирования до развертывания. Каждый инструмент был отобран с учетом его соответствия требованиям проекта, производительности и удобства использования, что в совокупности создает комфортные условия для разработки веб-сервиса.

1. Обоснование выбора среды программирования

В качестве основной среды разработки для проекта был выбран Visual Studio Code — современный редактор кода с широкими возможностями настройки и богатой экосистемой расширений. Этот выбор обусловлен рядом ключевых преимуществ, которые особенно важны для разработки веб-приложений.

VS Code обеспечивает превосходную поддержку TypeScript – основного языка разработки в проекте. Редактор предоставляет автодополнение кода, навигацию по типам, мгновенную проверку ошибок и рефакторинг, что значительно ускоряет процесс написания кода. Для Next.js и NestJS доступны специализированные расширения, упрощающие работу с этими фреймворками.

Редактор имеет встроенную интеграцию с Git, что позволяет выполнять основные операции с репозиторием прямо из интерфейса. Для работы с Docker доступны удобные инструменты управления контейнерами. Поддержка ESLint и Prettier обеспечивает соблюдение стиля кодирования и выявление потенциальных проблем.

VS Code отличается высокой производительностью даже при работе с крупными проектами. Редактор потребляет умеренное количество системных ресурсов, что важно для комфортной разработки. Возможность тонкой настройки интерфейса и горячих клавиш позволяет адаптировать среду под индивидуальные предпочтения.

VS Code доступен для всех основных операционных систем, что обеспечивает согласованность среды разработки независимо от используемой платформы.

Встроенный отладчик поддерживает как клиентский, так и серверный код. Для тестирования API через Postman доступно специальное расширение, позволяющее отправлять запросы и просматривать ответы прямо из редактора.

Богатая библиотека расширений позволяет добавлять в редактор только необходимые функции, избегая избыточности.

1. Информационное обеспечение

В проекте активно используется система Git в сочетании с платформой GitHub. Git обеспечивает надежный контроль версий, позволяя фиксировать изменения на разных этапах разработки. Это дает возможность отслеживать историю модификаций, возвращаться к предыдущим состояниям проекта и создавать изолированные ветки для реализации новых функций. GitHub служит облачным хранилищем для репозитория, предоставляя удобный интерфейс для управления кодом и дополнительными возможностями.

В качестве основного хранилища данных применяется реляционная система управления базой данных PostgreSQL. Эта система обеспечивает надежное хранение всей информации проекта, включая пользовательские данные, статистику тренировок и системные настройки. Для администрирования базы данных и выполнения запросов используется графический интерфейс pgAdmin, который предоставляет удобные инструменты для работы со структурой базы, выполнения SQL-запросов и анализа производительности.

Для работы с документацией и сопроводительными материалами использовались облачные офисные решения Google Workspace. Google Docs применялся для создания и совместного редактирования технической документации, обеспечивая удобный доступ к актуальным версиям документов с любого устройства. Для подготовки презентационных материалов и демонстрации возможностей системы применялся Google Slides, позволяющий создавать интерактивные презентации с возможностью онлайн-демонстрации.

В процессе разработки игровых элементов проекта использовались специализированные инструменты для работы с графикой. TexturePacker применялся для создания оптимальных текстурных атласов из отдельных графических ресурсов. Этот инструмент позволил значительно улучшить производительность рендеринга игровых сцен за счет:

* объединения множества отдельных изображений в единые текстуры;
* автоматической оптимизации пространства текстур;
* генерации соответствующих JSON-файлов с координатами спрайтов.

Tiled Map Editor использовался для создания и редактирования игровых карт и уровней. Этот редактор предоставил удобный визуальный интерфейс для:

* проектирования структуры игровых уровней;
* размещения объектов и декоративных элементов;
* настройки слоев и системы коллизий;
* экспорта карт в формате JSON для последующего использования в Phaser.

1. Обзор и анализ существующих программных систем

Stamina:

* достоинства:

1. сервис представляет собой классический тренажер слепой печати, доступный как в виде отдельной программы, так и в онлайн-версии.;
2. главным преимуществом является гибкость настроек – пользователь может самостоятельно регулировать сложность упражнений, выбирать тексты для тренировки и настраивать интерфейс под свои предпочтения;
3. программа отличается оригинальным юмористическим подходом, что делает процесс обучения менее напряженным;
4. наличие оффлайн-версии позволяет заниматься без подключения к интернету;

* недостатки:

1. интерфейс Stamina выглядит устаревшим и не соответствует современным стандартам UX/UI;
2. функционал ограничен базовыми упражнениями без продвинутой аналитики результатов.

Typing Study:

* достоинства:

1. сервис предлагает четко структурированную программу обучения;
2. поддерживается множество языковых раскладок, включая специализированные варианты;
3. модуль для тренировки работы с цифровой клавиатурой;

* недостатки:

1. функционал анализа результатов ограничен базовой статистикой по скорости и точности без персонализированных рекомендаций по улучшению навыков;
2. обучение построено на традиционной схеме выполнения упражнений без дополнительных мотивационных элементов.

Клавогонки:

* достоинства:

1. Клавогонки реализуют уникальный игровой подход к обучению через систему многопользовательских соревнований в реальном времени;
2. разнообразные режимы позволяют отрабатывать разные аспекты печати;
3. рейтинговая система и достижения создают сильную мотивационную составляющую;

* недостатки:

1. сервис требует стабильного интернет-соединения для полноценной работы;
2. обучающий контент представлен в минимальном объеме, основной акцент сделан на соревновательную составляющую;
3. отсутствует структурированная программа для начинающих.

# Список литературы