МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное бюджетное образовательное учреждение высшего образования

ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ

Факультет «Вычислительная техника» (наименование)

Кафедра «Системы автоматизированного проектирования» (наименование)

Направление подготовки 09.03.01 Информатика и вычислительная техника (код и наименование)

Профиль Системы автоматизированного проектирования (наименование)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА БАКАЛАВРА

на тему:

Проектирование и разработка обучающего тренажера для изучения волнового алгоритма

Студент 07.06 (подпись, дата) Исаев С.Д. (фамилия, инициалы)

Руководитель 07.06.24 (подпись, дата) Эпп В.В. (фамилия, инициалы)

Нормоконтролёр доц. каф. САПР ПГУ (должность, место работы) 07.06.24 (подпись, дата) Бождай А.С. (фамилия, инициалы)

Работа допущена к защите

Заведующий кафедрой (подпись) Бурукина И.П (фамилия, инициалы)

Работа защищена с отметкой хор (протокол заседания ГЭК от 19.06.24 № 2)

Секретарь ГЭК (подпись) Финогеев А.А. (фамилия, инициалы)

Пенза 2024

“Утверждаю”

Заведующий кафедрой САПР

_____ /Бурукина И.П. /

“11” 06 2024 г.

ЗАДАНИЕ

на выполнение бакалаврской работы

1. Студент гр. 20ВВС1 факультета ВТ специальности 09.03.01 «Информатика и вычислительная техника».

Исаев Сергей Дмитриевич

(фамилия, имя, отчество)

2. Руководитель проекта к.т.н., доцент кафедры САПР Эпп Виталина Викторовна

3. Время дипломного проектирования с 08.03 2024 г. по 18.05 2024 г.

4. Место преддипломной практики кафедра САПР ПГУ

5. Тема проекта: «Проектирование и разработка обучающего тренажера для изучения волнового алгоритма»

Тема утверждена приказом ПГУ № 53010 от “23” мая 2024 г.

6. Техническое задание на бакалаврскую работу

6.1 Предмет: «обучающий тренажер для изучения волнового алгоритма».

6.2 Назначение: улучшение качества изучения волнового алгоритма, повышения удобства и понятности изучаемого материала благодаря интерактивному подходу к осваиванию материала.

6.3 Функциональные характеристики: тренажер предоставляет пользователю возможность изучения основ волнового алгоритма по средством различных, сгенерированных заданий.

6.4 Технические характеристики: персональный компьютер с браузером Google Chrome версией 124.0.6367.60/61 для корректной работы элементов интерфейса.

7. Объем и содержание основной части проекта

7.1. Содержание пояснительной записки (перечень вопросов, подлежащих разработке, расчетов, обоснований, описаний)

7.1.1. Анализ предметной области.

7.1.3. Проектирование обучающего тренажера.

7.1.4. Реализация обучающего тренажера.

7.1.5. Проверка работоспособности.

8. Календарный график работ по выполнению проекта.

Наименование этапов работы	Объем работы	Срок выполнения	Подпись руководителя, консультанта
1. Анализ ТЗ	15 %	20.3-25.3	
2. Проектирование	25 %	25.3-10.4	
3. Разработка ПО	35%	20.4-10.5	
4. Проверка работоспособности	10%	10.5-14.5	
5. Оформление ПЗ	15 %	15.5-30.5	
6. Нормоконтроль		07.06.24	

Дата выдачи задания "18" марта 2024 г.

Руководитель дипломного проекта _____ / Фин В.Р

Задание к исполнению принял "18" марта 2024 г.

Студент _____ / Исаев С.Д.

Бакалаврскую работу к защите допустить

Декан ФВТ _____ / Л.Р. Фионова .

17.06.2024

Реферат

Пояснительная записка содержит страниц, рисунка, 9 использованных источников и 1 приложение.

ОБУЧАЮЩИЙ ТРЕНАЖЕР ВОЛНОВОГО АЛГОРИТМА, ВЕБ-САЙТ, MICROSOFT VISUAL STUDIO CODE 2022, JavaScript, HTML

В рамках данной работы была поставлена цель – разработать обучающий тренажер для изучения волнового алгоритма, который улучшит качество получаемых знаний при работе с волновым алгоритмом.

Объектами разработки являются:

Интернет-ресурс, на котором расположен тренажер.

Тренажер, включающий в себя практические задания для развития знаний пользователя.

Разработанный тренажер должен позволять пользователям решать интерактивные задачи, в процессе которых будут улучшаться их знания и возможность создавать свои упражнения для удобства обучения.

В результате проделанной работы разработан обучающий тренажер для изучения волнового алгоритма, который повышает качество усвоения изученного материала благодаря интерактивному подходу, что позволяет пользователю контролировать весь процесс работы волнового алгоритма. Проведена проверка работоспособности обучающего тренажера.

Язык программирования: JavaScript.

Среда разработки: Visual studio code 2022

Отзыв
на выпускную квалификационную работу

«Проектирование и разработка обучающего тренажера для изучения волнового алгоритма»
(тема выпускной квалификационной работы)

студента Исаева Сергея Дмитриевича
(ФИО)

Группа 20ВВС1 09.03.01 «Информатика и вычислительная техника»
Направленность (профиль подготовки) «Системы автоматизированного проектирования»

Актуальность темы ВКР, определение задач проектирования с учетом потребностей практики, полноты раскрытия темы: выпускная квалификационная работа направлена на разработку обучающего тренажера для изучения волнового алгоритма. Волновые алгоритмы активно используются в области искусственного интеллекта, робототехники, компьютерного зрения и других смежных областей, поэтому изучение работы самого алгоритма является актуальной задачей.

Анализ результатов: в результате выполнения ВКР разработан обучающий тренажер, который позволит решать интерактивные задачи по освоению работы волнового алгоритма, а также у пользователя имеется возможность создавать свои упражнения для проверки усвоенного материала.

Качество проработки выпускником отдельных разделов, применение выпускником передовых методов и новейших учебных достижений: выпускная квалификационная работа выполнена в полном объеме, на хорошем техническом уровне. Теоретическая и практическая значимость работы состоит в систематизации знаний по разработке приложений, а также углубления знаний по написанию качественного программного кода.

Уровень общей подготовленности выпускника, способность его к самостоятельной деятельности: хотелось бы отметить умение студента работать самостоятельно от постановки задачи до ее практического решения, а также активность в изучении новых технологий и подходов к разработке программного кода.

Выводы и рекомендации: считаю, что работа выполнена на хорошем уровне и Исаев С.Д. заслуживает присуждение ему квалификации бакалавра по специальности «Информатика и вычислительная техника», с оценкой хорошо.

Руководитель: Эпп Виталина Викторовна
(ФИО)
к.т.н., доцент кафедры «Системы автоматизированного проектирования»
ФГБОУ ВО «Пензенский государственный университет»
(ученая степень, должность)

«_07» __июня___2024_г. _____
(подпись)

АНТИПЛАГИАТ
ОБНАРУЖЕНИЕ ЗАИМСТВОВАНИЙ

СПРАВКА

Пензенский государственный университет

о результатах проверки текстового документа
на наличие заимствований

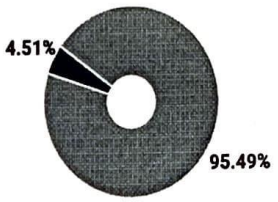
ПРОВЕРКА ВЫПОЛНЕНА В СИСТЕМЕ АНТИПЛАГИАТ.ВУЗ

Автор работы: Исаев С Д
Самоцитирование рассчитано для: Исаев С Д
Название работы: VKR_2.0_на Антиплагиат
Тип работы: Не указано
Подразделение: САПР

РЕЗУЛЬТАТЫ

СОВПАДЕНИЯ	4.51%
ОРИГИНАЛЬНОСТЬ	95.49%
ЦИТИРОВАНИЯ	0%
САМОЦИТИРОВАНИЯ	0%

ДАТА ПОСЛЕДНЕЙ ПРОВЕРКИ: 07.06.2024



Структура документа: Проверенные разделы: содержание с.5, основная часть с.1-4, 6-61

Модули поиска: Библиография; Коллекция НБУ; Патенты СССР, РФ, СНГ; Шаблонные фразы; СМИ России и СНГ; Переводные заимствования*; ИПС Адилет; Диссертации НББ; Публикации РГБ; Цитирование; СПС ГАРАНТ: аналитика; Перефразирования по коллекции IEEE; IEEE; Кольцо вузов; Публикации eLIBRARY (переводы и перефразирования); Переводные заимствования (RuEn); Перефразирования по Интернету (EN); Сводная коллекция ЭБС; Переводные заимствования по коллекции Интернет в русском сегменте; Перефразирования по Интернету; Переводные заимствования IEEE; Медицина; Переводные заимствования по Интернету (EnRu); Перефразирования по СПС ГАРАНТ: аналитика; Переводные заимствования по коллекции Гарант: аналитика; Переводные заимствования по коллекции Интернет в английском сегменте;

Работу проверил: Эпп Виталина Викторовна
ФИО проверяющего

Дата подписи: 07.06.2024

Подпись проверяющего

Чтобы убедиться
в подлинности справки, используйте QR-код,
который содержит ссылку на отчет.

Ответ на вопрос, является ли обнаруженное заимствование корректным, система оставляет на усмотрение проверяющего. Предоставленная информация не подлежит использованию в коммерческих целях.

..6
..7
...8
..8
11
.12
.12
.13
.15
15
.18
.18
.19
19
.26
.33
.34
34
34
.46
50
.51
.52

Перечень обозначений и сокращений

UML (Unified Modeling Language) – язык моделирования

САПР – системы автоматизированного проектирования

ПО – программное обеспечение

MySQL – система управления базами данных

SQL – язык программирования для управления информацией в базах данных

БД – база данных

HTML – язык разметки веб-страниц

CSS – язык стилей веб-страниц

ЭИОС – электронная информационно-образовательная среда

Введение

Процесс обучения всегда стремится к улучшению нескольких ключевых критериев: качество образования, доступность информации, понятность информации, удобство получения информации, наглядность информации.

Современный человек не представляет свою жизнь без каждодневного выхода в интернет, в связи с чем, привыкает к телефону или компьютеру, как к средству постоянного использования, а интернет становится средством получения информации. Из-за чего, сейчас происходит цифровизация “всего”. Все книги, журналы, отчеты, учетные ведомости переносятся в цифровой формат, так как, банально, благодаря такому формату хранения информации, она становится “вечной” – то есть исключается возможность утраты или потери физического источника, исключается любого вида коррозия физического источника. Также цифровой информацией гораздо проще управлять, благодаря функциональным возможностям вычислительных машин (телефон, компьютер)[6]. Даже взаимодействия Института и студента плавно переходит в цифровой формат, так ЭИОС – система института, которая включает в себя личную карточку студента, его успеваемость, список пройденных курсов, также выступает в роли электронной библиотеки, в которой хранятся методички, книги, работы, выполненные студентов, за годы обучения, научные статьи. Часть дисциплин содержат на своих страницах в ЭИОС задания для проверки успеваемости и повторения материала[8].

Таким образом, обучающий тренажер является современным способом, удобным и привычным способом изучения и усвоения различного рода материала.

1. Анализ предметной области

1.1. Актуальность

В настоящее время волновые алгоритмы привлекают все больше внимания специалистов в области искусственного интеллекта, робототехники, компьютерного зрения и других смежных областей.

Волновые алгоритмы широко применяются в задачах пути в графах, таких как поиск кратчайшего пути в лабиринтах, маршрутизация в сетях и многих других. Эти алгоритмы находят применение не только в теоретических исследованиях, но и в практических задачах, в которых требуется оптимальное решение с учетом различных ограничений и условий.

Обучающие тренажеры в современном образовании играют ключевую роль в процессе обучения, позволяя студентам и школьникам на практике изучать сложные темы и алгоритмы. Разработка такого тренажера для волнового алгоритма будет способствовать более глубокому и понятному усвоению материала, а также позволит студентам применять полученные знания на практике.

С учетом роста интереса к областям искусственного интеллекта и робототехники актуальность данной работы возрастает. Волновые алгоритмы часто используются в алгоритмах машинного обучения и робототехнике для решения задачи навигации и планирования движения. Понимание принципов работы волновых алгоритмов становится необходимым для специалистов, работающих в этих областях.

Также следует отметить, что современные методы обучения все больше ориентированы на интерактивные и визуальные методы. Обучающий тренажер для волнового алгоритма может представлять собой интуитивно понятный и визуально привлекательный инструмент, который поможет студентам лучше понять принципы работы алгоритма, а также научиться применять его в различных практических задачах.

Кроме того, разработка обучающего тренажера для волнового алгоритма может стать основой для дальнейших исследований в области разработки алгоритмов и методов искусственного интеллекта. Полученные в ходе работы результаты могут быть использованы в дальнейших исследованиях, направленных на улучшение и оптимизацию волновых алгоритмов, а также на создание новых методов и подходов в этой области.

Таким образом, разработка обучающего тренажера для изучения волнового алгоритма является актуальной и перспективной задачей, которая вносит вклад в развитие образования, науки и технологий. Обучающий тренажер позволит студентам глубже понять принципы работы волновых алгоритмов, и применять полученные знания в практических задачах.

Волновой алгоритм или алгоритм Ли – служит для поиска кратчайшего пути, выпуская волну от источника до приемника. Весь алгоритм можно разбить на две части [4]:

1. На рисунке 1 клетка с идентификатором ноль – стартовая позиция(источник), закрашенный красный квадрат – позиция, до которой нужно проложить путь(приемник). Волна из стартовой позиции распространялась в соседние клетки в 4 направлениях и нумеровала каждую клетку. Дойдя до красного квадрата, распространение прекратилось.

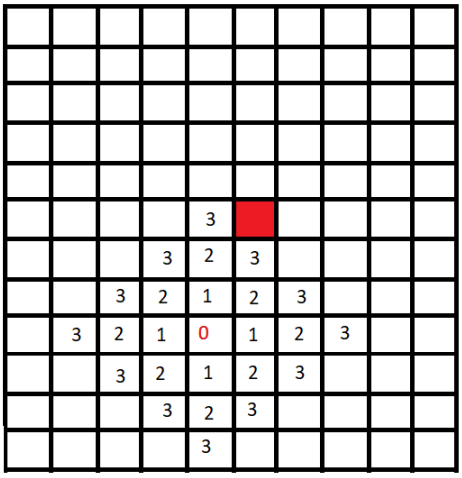


Рисунок 1 - Распространение волны

Площадь, на которой нужно проложить путь представляется в виде сетки с равными клетками. Из выбранной начальной позиции(источника), с идентификатором ноль, распространяется волна в соседние клетки. Достигнув новой клетки, волна присваивает клетке номер на один выше, чем у той из которой она пришла. Как только волна достигла точки, до которой нужно проложить маршрут(приемника) волна останавливается.

2. От приемника прокладывается трасса до источника, двигаясь по пронумерованным клеткам, выбирая клетку с наименьшим числовым значением.

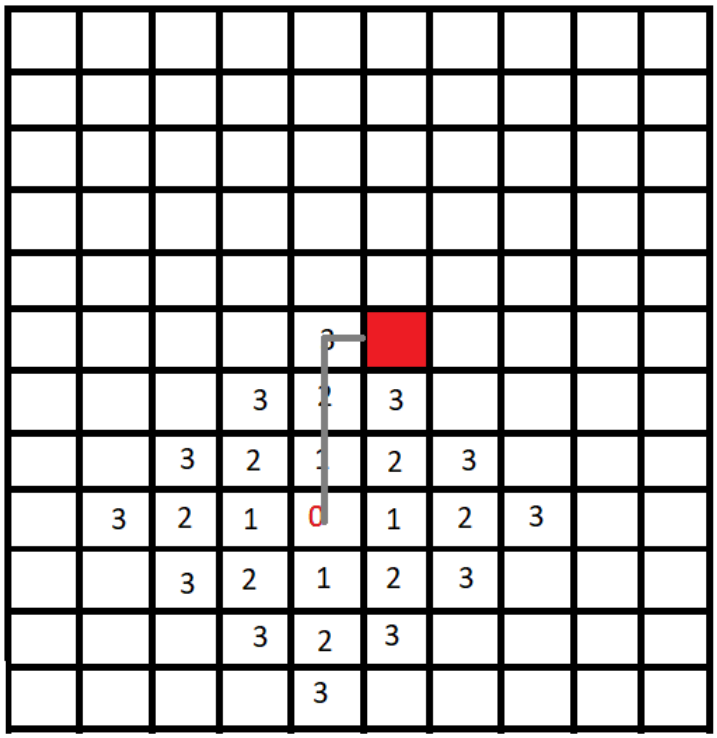


Рисунок 2 - Построение маршрута

На рисунке 2 отражено как трасса, обозначенная серым цветов, прокладывается от приемника к источнику, перемещаясь в соседнюю клетку с наименьшим номером.

1.2. Прикладное использование алгоритма Ли

Волновой алгоритм используется в таких областях как: разработка игр, распределенные системы и сети, робототехника, обработка изображений, трассировка печатных плат и тд.

Использование волнового алгоритма в играх обусловлено обширным набором задач, которые алгоритм Ли способен решить. В основном используется для поиска различных путей. Поиск пути в лабиринте, построение маршрута до указанной цели. Вторая область применения в играх – это создание реалистичных движений персонажа или врага. Принцип использования волнового алгоритма для создания движений: сначала, определяется точка, в которую должен прийти персонаж, затем пространство вокруг персонажа разбивается на ячейки, далее алгоритм ищет кратчайший путь до цели. Последний этап – анимация. Благодаря использованию алгоритма Ли вокруг персонажа образуется пронумерованное поле, из-за чего можно легко изменять положение персонажа в каждой отдельно взятой клетке. Поэтому движения получаются сглаженными и плавными.

В распределенных системах и сетях алгоритм Ли применяется для поиска оптимальных маршрутов передачи данных, из-за чего минимизируются задержки и повышается пропускная способность, а в синхронизации находит применение в следующем аспекте: каждый узел имеет свое локальное время, поэтому важно чтобы время на каждом узле синхронизировалось. Волновой алгоритм позволяет легко производить синхронизацию. Один узел выбирается в качестве координатора, а его время устанавливается глобально. Дальше распространяется волна, по которой в последствии отправляется сообщение с временем координатора и поправкой на задержку при передаче этого сообщения. После получения сообщения, узел пересылает пакет данных на соседний узел. Таким образом можно производить синхронизацию в сетях.

В робототехнике алгоритм Ли применяется для построения маршрута и движения робота. Позволяет ориентироваться в трудных условиях и обходить препятствия.

В работе с изображениями применяется для распознавания образов на картинке: лица, дорожные знаки, определенные предметы

В процессе трассировки печатных плат используется в САПР. Волновой алгоритм применяется для разводки печатных плат, производя размещение компонентов микросхемы, а также прокладку соединений между элементами платы.

1.3. Целевая аудитория обучающего тренажера для изучения волнового алгоритма

1.3.1. Субъекты изучения волнового алгоритма

Волновой алгоритм позволяет решать большой набор задач, поэтому круг пользователей довольно обширен. Для лучшего отображения была составлена таблица субъектов и целей использования (табл. 1). В которой рассматриваются такие субъекты использования как: студент, преподаватель, инженер и разработчик.

Инженеры и разработчики используют большое количество готовых программных продуктов, в которых реализован волновой алгоритм, в то время как для изучения алгоритма Ли все еще используются старые нецифровые методы обучения, которые имеют ряд недостатков, таких как: непонятность, неудобность использования, обязательное присутствие наблюдателя, который может производить оценку.

Таблица 1 – субъекты и цели использования

Субъект	**Цель использования**
Студент	Познакомиться с волновым алгоритм. Изучить основные принципы действия алгоритма.
Преподаватель	Познакомить студентов с волновым алгоритмом. Показать разные области применений.
Инженер	Поиск пути для маршрутизации, для построения трасс и размещения элементов на печатных платах.
Разработчик игр	Поиск пути, создание анимации.

Из таблицы 1 можно сделать вывод, что студентам и преподавателям важно изучить базовые возможности волнового алгоритма, чтобы иметь о нем представление. Инженеры и разработчики игр изучают алгоритм Ли более углубленно, ведь им важно решать практические задачи, ведь каждый из них стремится к глобальной оптимизации своей работы, а этого невозможно добиться, опираясь лишь на базовые знания. Поэтому целью ВКР является разработка обучающего тренажера, который будет предназначен для студентов.

1.3.2. Сравнение средств аналогов для изучения волнового алгоритма

Для более наглядного отображения сравнения была составлена таблица, в которой были рассмотрены следующие ПО, позволяющие работать с волновым алгоритмом: Altium designer, KiCad, Autodesk EAGLE, PADS PCB Design

Таблица 2 – Обзор программных продуктов для трассировки печатных плат

Критерии / ПО	Модульная трассировка	Редактирование соединений	Редактирование проложенных трасс
Altium designer	Только полная трассировка, с ручной корректировкой	Позволяет легко изменять и добавлять соединения между компонентами	Настройка изгибов трассы под любым углом, возможность замены изломов на дуги, изменение ширины канала.
KiCad	Полная, а также трассировка отдельных участков платы и копирование шаблона трассировки на другие участки	Позволяет легко изменять и добавлять соединения между компонентами	Настройка изгибов трассы под любым углом, возможность переключиться на режим закруглённый режим
Autodesk EAGLE	Только полная трассировка, с ручной корректировкой	Позволяет легко изменять и добавлять соединения между компонентами	Настройка изгибов трассы под любым углом. Ширина каналов подбирается автоматически под каждый вход\выход
PADS PCB Design	Полная, а также трассировка отдельных участков платы и копирование шаблона трассировки на другие участки	Позволяет легко изменять и добавлять соединения между компонентами	Настройка изгибов трассы под любым углом. Поддерживает технологию “push and shove” позволять расталкивать проложенные проводники.

В таблице 2 представлены популярные ПО, в которых реализован волновой алгоритм, которым можно решить множество задач трассировки и компоновки.

Сам процесс работы алгоритма скрыт. Пользователь видит только входные данные и выходной результат, поэтому изучить работу алгоритма не представляется возможным[3]. Исходя из этого было принято решение разработки тренажера, который наглядно будет отображать работу алгоритма Ли и позволит на практике с ним познакомиться.

1.4. Постановка задачи

Обучающий тренажер для изучение волнового алгоритма, размещенный в браузере, как интернет-ресурс, будет служить отличном инструментом для студентов, которой позволит студентам в рамках самостоятельной работы изучить принцип работы с алгоритмом Ли.

Система должна иметь личный кабинет, в котором будут находится задания. Эти задания можно создать самостоятельно, выбирая уровень сложности. Содержать в себе теоретическую информацию, которую студенты смогут изучать для применения этой информации в практический заданиях. Задания должны автоматически проверяться, показывая студенту правильно или неправильно решено задание

1.4.1. Функциональные требования

Разрабатываемый продукт должен выполнять следующие функции:

-Наличие теоретического материала про волновой алгоритм

-Наличие тренировочных заданий для изучения и повторения теории о волновом алгоритме

-Наличие личного кабинета для сохранения информации о пройденных задачах

-Возможность изменять задания

-Задания должны быть нескольких уровней сложности

-Задания выполняются интерактивно в браузере

-Автоматическая проверка выполнения

В настоящее время изучение волнового алгоритма происходит в традиционном формате с использованием методических указаний, пособий и

учебников. Из современных технологий используется разве что оцифрованные книги, которые можно найти в браузере и различного рода статьи. Тренажер привнес бы новые функции и методы в изучение волнового алгоритма. Для подтверждения этого тезиса была составлена таблица, отображающая недостатки и преимущества подходов в изучении волнового алгоритма, в которой сравниваются основные характеристики двух подходов по десяти-бальной шкале.

Таблица 3 – Особенности подходов к изучению волнового алгоритма

Наименование характеристики	**Изучение алгоритма с помощью различных методических указаний и технической литературы**	**Изучение алгоритма с использованием видео-уроков**	**Изучение алгоритма с использованием аудиофайлов**	**Изучение алгоритма с использованием обучающего тренажера**
Доступность	10	10	10	10
Использование технического оборудования	5	5	5	10
Обширность тренировочных заданий	7	1	3	10
Возможность проверки результатов выполненных упражнений	3	6	6	10
Сохранение информации о пройденный упражнениях	10	10	10	10

Продолжение таблицы 3

Наименование характеристики	**Изучение алгоритма с помощью различных методических указаний и технической литературы**	**Изучение алгоритма с использованием видео-уроков**	**Изучение алгоритма с использованием аудиофайлов**	**Изучение алгоритма с использованием обучающего тренажера**
Удобство использования	6	9	4	8
Наглядность информации	5	8	0	7
Наличие интерактива	0	4	2	10
Возможность управления сложностью упражнения	0	0	0	10
Возможность генерации новых тестовых заданий	0	0	0	10

Из таблицы 3 видно, что тренажер для волнового алгоритма имеет преимущество по многим характеристикам и лишь в некоторых аналоговые способы изучения волнового алгоритма не уступают разрабатываемому тренажеру. Система имеет огромное количество заданий, так как пользователь/студент имеет возможность генерировать упражнения, что минимизирует вероятность встретить одно и то же упражнение. Так же используя аналоговые источники упражнений такие как видео-уроки или аудио-уроки

студенту требуется время для подготовки задания к решению. Тренажер позволяет решать упражнение сразу и в той же среде, где задание было получено. Только бумажная книга/учебник может так же позволить обучающемуся сразу преступить к решению задания, цифровая книга аналогично требует подготовки к началу решения задания. Проверку правильности решения имеет книга с заданиями, если автор включил такой раздел, аудио- видео- уроки имеют в наличии ответ на представленное задание, но в таких методах обучения обычно представляются одно/два упражнения и на этом урок заканчивается. В свою очередь тренажер имеет ответ на все задания, которые были сгенерированы на веб-сайте, что позволяет студенту сразу понять правильно ли решено задание.

1.5. Аппаратное обеспечение

Веб-сервис разработан под настольную версию браузера Google Chrome 124.0.6367.60/61.

Минимальные технические требования:

- Процессор – intel Pentium 4 или более поздней версии, 2,3 ГГц или выше
- Оперативная память – 1024 Мб или выше
- Свободное дисковое пространство – 1000 Мб
- Разрешение дисплея 1024x768 точек или выше
- Windows 10 или более поздней версии

1.6. Программное обеспечение

Программное обеспечение состоит из нескольких модулей HTML, CSS и JavaScript файлов и браузера Google Chrome, который служит для визуализации системы

2. Проектирование системы

2.1. Структура системы

Для отображения того, как функционирует система были разработаны некоторые диаграммы UML:

- Диаграмма использования [1].
- Диаграмма действий (активности)[2].
- Диаграмма компонентов[7].
- Диаграмма последовательности.

А также для отображения работы алгоритмов были составлены несколько блок схем.

На рисунке 3 представлена Диаграмма использования системы.

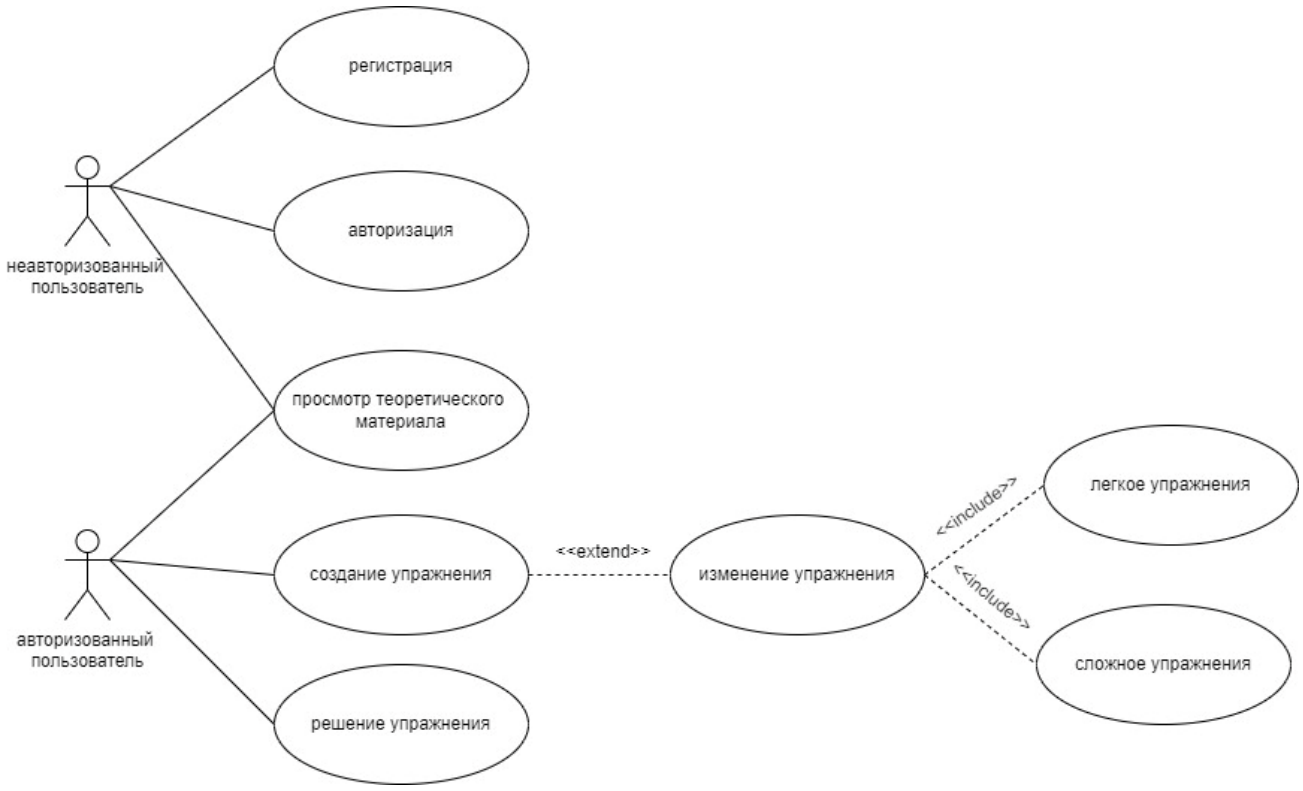


Рисунок 3 – Диаграмма использования системы

Диаграмма использования показывает возможные действия пользователя с системой.

Неавторизованный пользователь имеет возможность пройти регистрацию, выполнить авторизацию и просматривать теоретический материал.

Авторизованный пользователь имеет возможность решать упражнения, просматривать теоретический материал и создавать упражнения.

В свою очередь создание упражнения расширяется блоком изменение упражнения, который включается в себя выбор сложности упражнения: легкое упражнение соответствует простому уровню сложности, сложное упражнение соответствует тяжелому уровню сложности.

На рисунке 4 представлена Диаграмма действий (активности) для незарегистрированного пользователя.

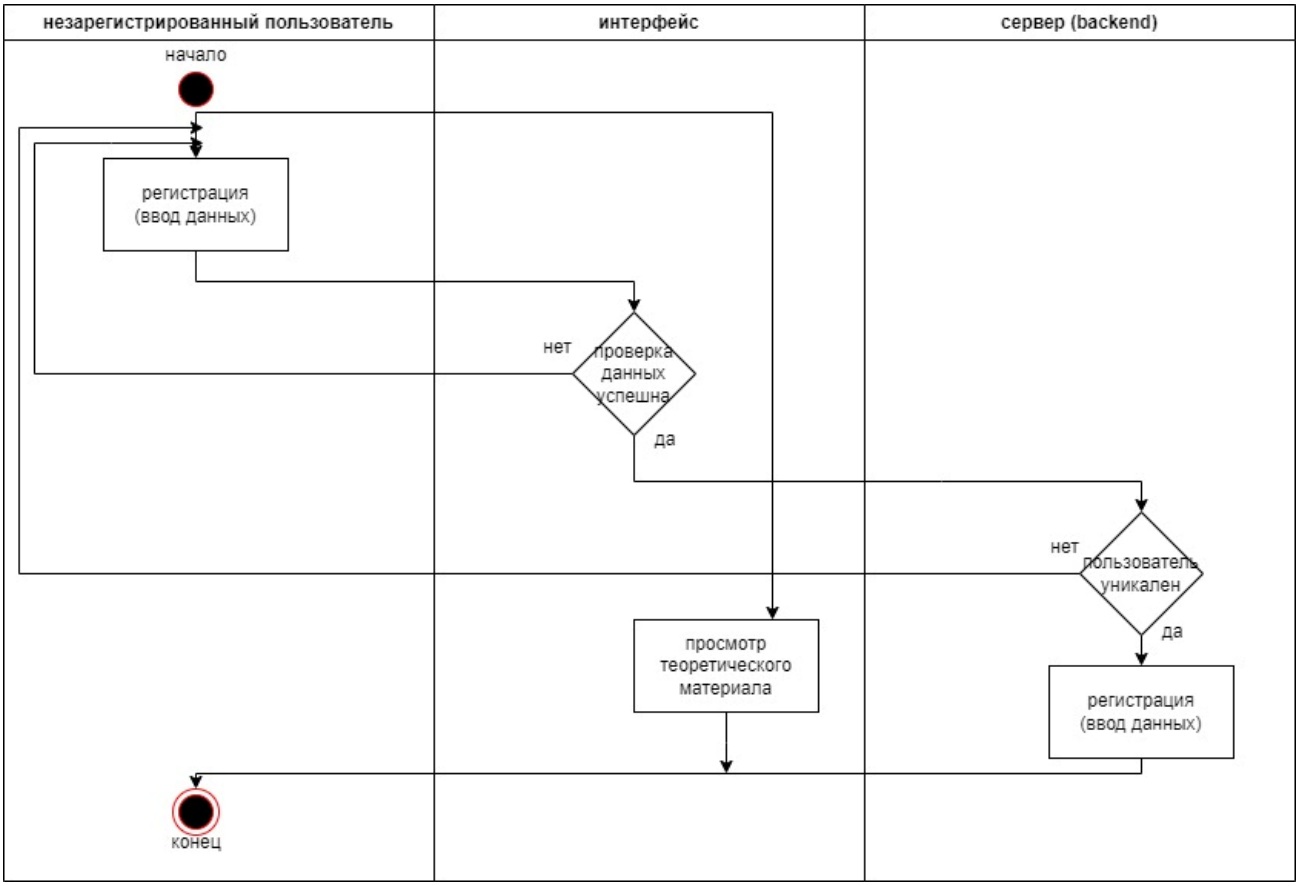


Рисунок 4 – Диаграмма действий для незарегистрированного пользователя

Диаграмма действий отражает возможные действия незарегистрированного пользователя при работе с тренажером:

У незарегистрированного пользователя есть две опции действий, либо зарегистрироваться, где после ввода данных произойдет проверка на корректность введённых данных и проверка на уникальность логина для того, чтобы исключить перезапись данных пользователей и дублирование записей в БД, либо без регистрации просматривать теоретический материал.

На рисунке 5 представлена диаграмма действий (активности) создания упражнения для зарегистрированного пользователя.

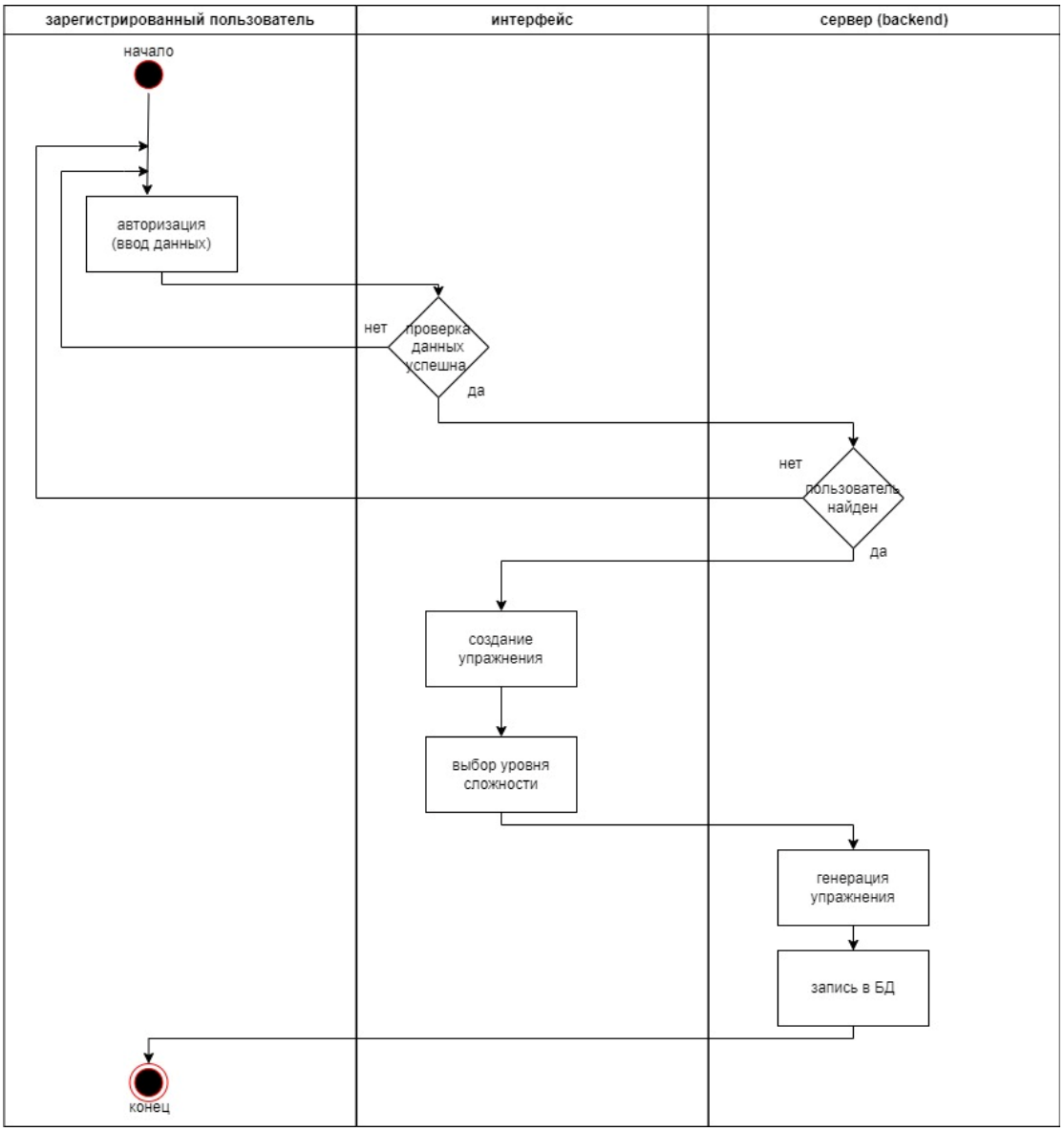


Рисунок 5 – Диаграмма действий для зарегистрированного пользователя

Для того, чтобы создать упражнение пользователю необходимо авторизоваться, иначе возможности создавать упражнения у неавторизованного пользователя не будет.

После ввода данных для авторизации происходит проверка данных на корректность, если проверка не пройдена, то пользователю нужно повторить ввод данных, если проверка пройдена, то дальше следует вторая проверка на наличие такого пользователя в БД, если пользователь не найден, то нужно повторно ввести данные для авторизации, если найден, то теперь авторизованный пользователь может создать упражнение. В таком случае авторизованному пользователю предлагается выбор уровня сложности упражнения, после чего происходит генерация упражнения и запись этого упражнения в БД.

На рисунке 6 представлена Диаграмма компонентов.

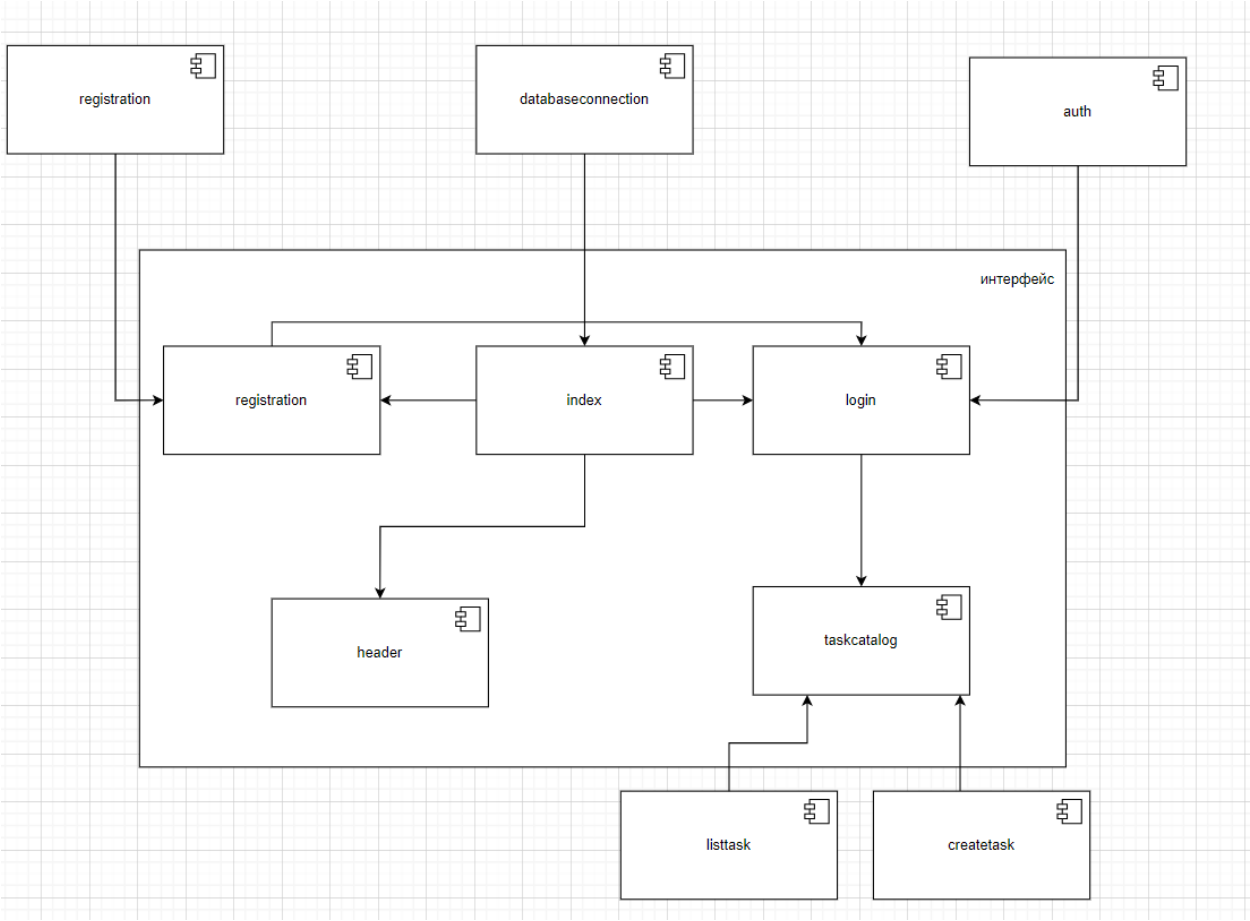


Рисунок 6 - Диаграмма компонентов

В объекте интерфейс представлены компоненты интернет-ресурса:

- Index - предназначена для отображения главной или начальной страницы веб-сайта.
- Registration – отображает страницу с полями для ввода данных, служит для регистрации пользователя.
- Login – отображает страницу с полями для ввода данных, служит для авторизации зарегистрированных пользователей.
- Header – отображает часть страницы - заголовок, в котором включены блоки переходов по страницам ресурса.
- Taskcatalog – отображает страницу с каталогом заданий, на этой странице можно выполнять упражнения.

Вне объекта интерфейс представлены функциональные компоненты:

- Databaseconnection – функция отвечающая за подключение веб-сайта к базе данных.
- Registration – функция для регистрации пользователя на веб-сайте. Отправляет POST запрос к базе данных с введенными пользователем данными и создает новую запись.
- Auth - функция для авторизации пользователя на веб-сайте. Отправляет GET запрос к базе данных с введенными пользователем данными. Не создает новых записей в базе данных.
- Listtask – функция отправляет GET запрос к базе данных для получения всех упражнений.
- Createtask – функция отправляет POST запрос к базе данных с введённой информацией об уровне сложности задания и генерирует упражнение.

На рисунке 7 представлена Диаграмма последовательности регистрации пользователя.

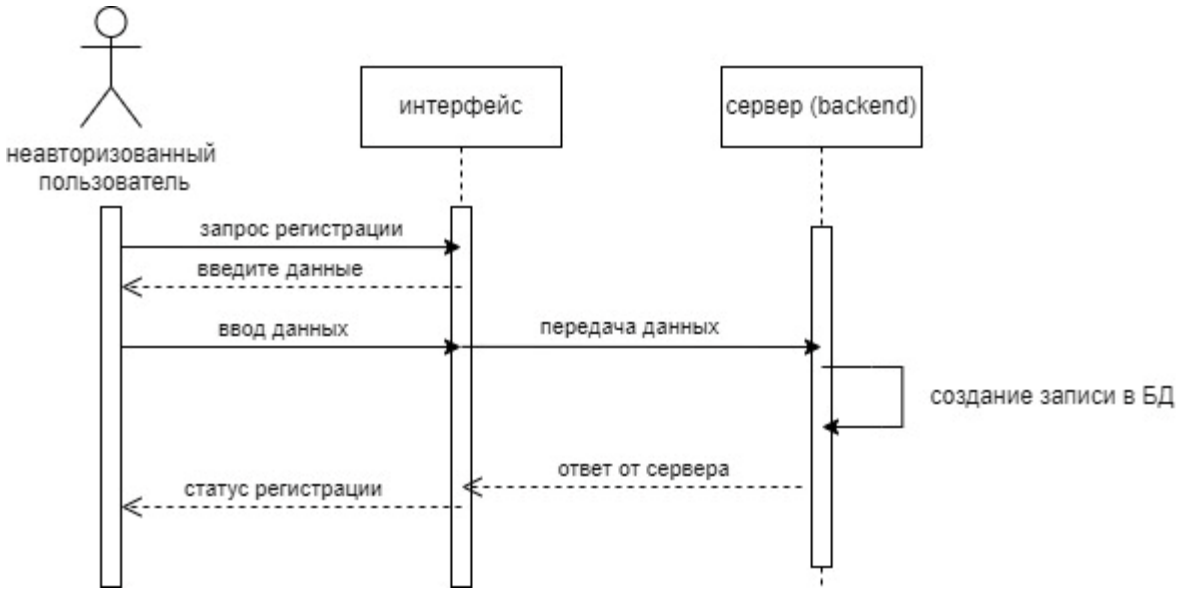


Рисунок 7 – Регистрация пользователя

На рисунке 7 видно, как ведет себя система при регистрации пользователя и отражено какое действие за каким следует (прямая стрелка – запрос, пунктирная стрелка – ответ):

Сначала пользователь инициирует процесс регистрации запросом регистрации через интерфейс. После система требует от пользователя ввода данных, пользователь заполняет поля для ввода, и система отправляет данные на сервер. На сервере происходит проверка на уникальность введенных данных, после чего в случае прохождения проверки создается запись в базу данных, после записи или в случае не уникальности данных отправляется ответ от сервера и до пользователя доходит статус регистрации, если запись создана, то статус регистрации - успешно, если запись не создана, то статус регистрации - ошибка.

На рисунке 8 представлена Диаграмма последовательности создания упражнения

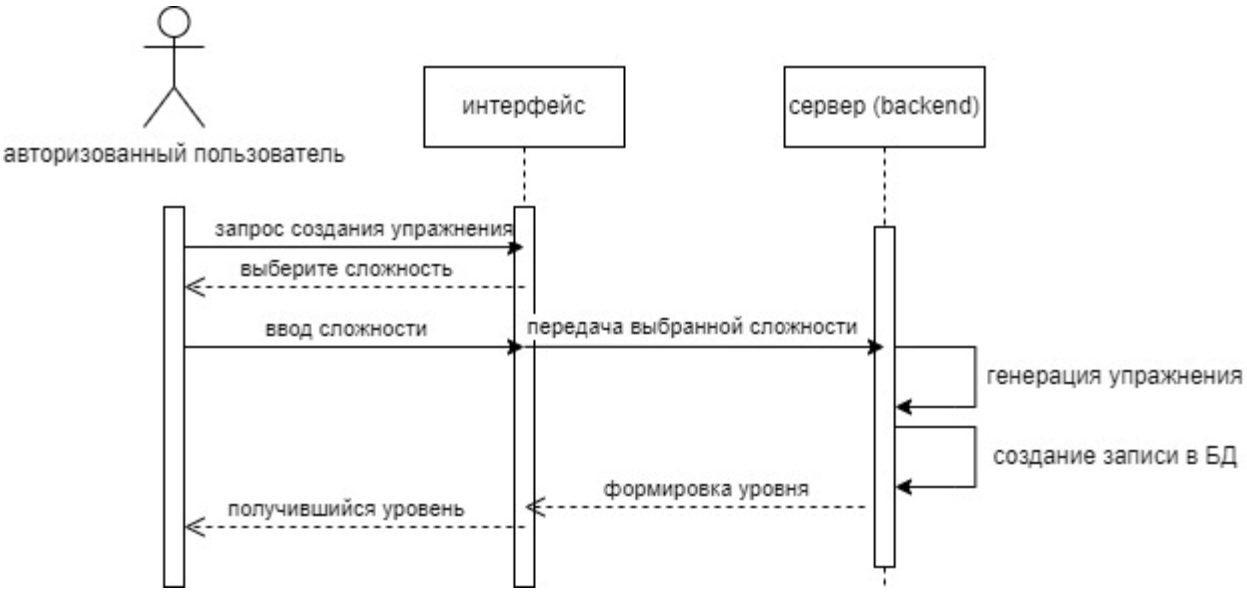


Рисунок 8 – Создание упражнения

На рисунке 8 отображена последовательность действий авторизованного пользователя для создания упражнения.

Сначала пользователь инициирует процесс создания упражнения запросом создания через интерфейс. После, система требует от пользователя выбрать уровень сложности, пользователь вводит сложность, и система отправляет данные на сервер. На сервере происходит генерация упражнения, после чего производится запись в базу данных, после записи отправляется ответ от сервера с формированием упражнения, где происходит обработка полученного ответа, распределение объектов упражнения по переменным для отрисовки на html странице и пользователю доходит сформированное упражнение.

На рисунке 9 представлена диаграмма последовательности загрузки упражнения из БД.

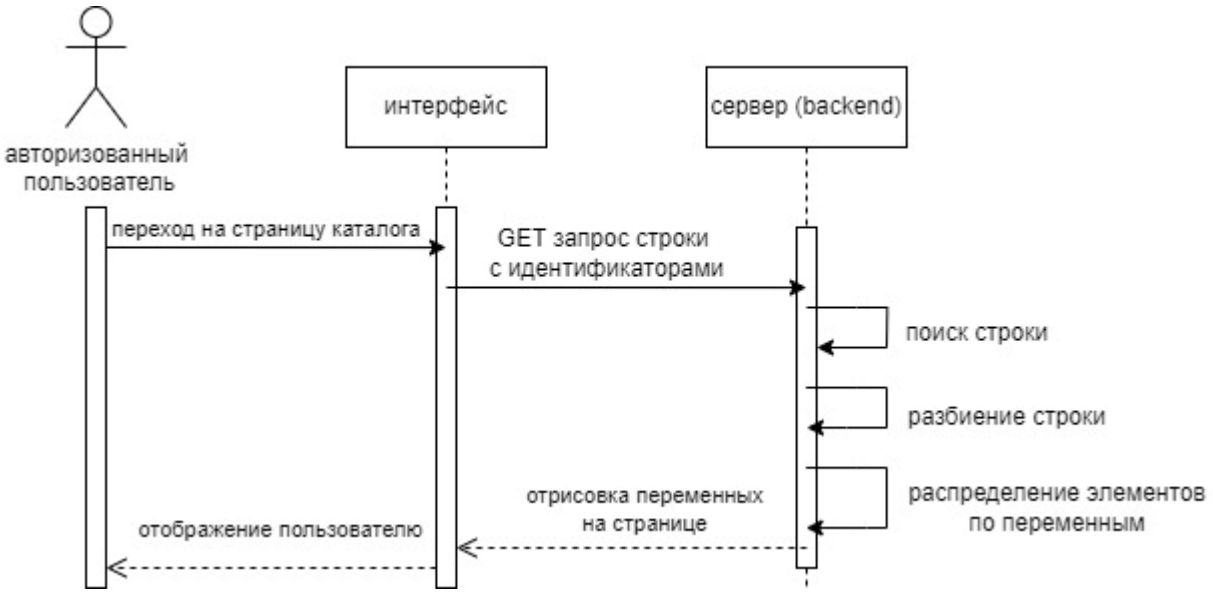


Рисунок 9 – Диаграмма последовательности загрузки упражнения

Пользователь переходит на страницу с каталогом заданий, далее отправляется GET-запрос к БД, следующим шагом происходит поиск строки в БД, затем скрипт принимает строку, разбивает ее на идентификаторы, распределяет элементы по блокам и последним шагом скрипта является отрисовка элементов на html-странице, а последним шагом на диаграмме является получение пользователем упражнение.

2.2. Алгоритм системы

Функциональные взаимодействия пользователя с обучающим тренажером для волнового алгоритма описаны блок схемами, которые отражают алгоритмы работы отдельных модулей системы.

На рисунке 10 представлен алгоритм выполнения регистрации пользователем

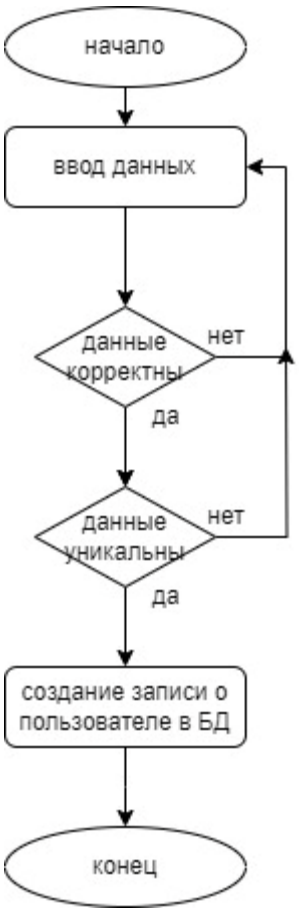


Рисунок 10 – Блок схема алгоритма регистрации

Алгоритм на рисунке 10 выполняется следующим образом: пользователь вводит свои данные (логин и пароль), после чего производится проверка данных на корректность (отсутствуют недопустимые символы), если проверка не пройдена, тогда следует по новой ввести данные, если пройдена, тогда происходит проверка данных на уникальность, то есть идет поиск записей в базе данных, если такие записи есть, значит пользователь не считается уникальным и не может быть записан повторно, следовательно, нужно перезаполнить данные для регистрации, если записей с таким логином нет, то значит введенные данные уникальные и они записываются в базу данных.

На рисунке 11 представлен алгоритм авторизации пользователя.

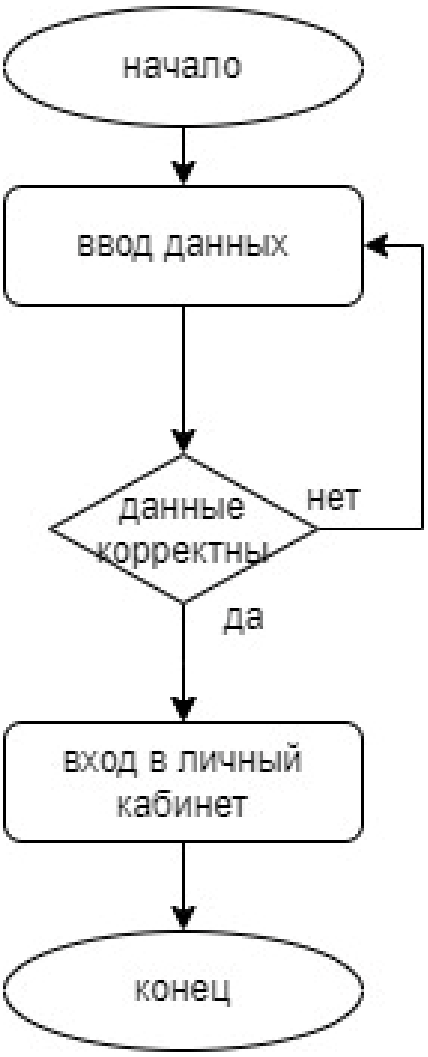


Рисунок 11 – Блок схема алгоритма авторизации

Алгоритм на рисунке 11 выполняется следующим образом: пользователь вводит свои данные, которые уже записаны в базу данных, после происходит проверка данных на корректность, где проверяется те ли символы введены, нет ли пробелов, пустых строк, если проверка не пройдена, то следует ввести данные повторно, а если проверка пройдена, то происходит вход в личный кабинет.

На рисунке 12 представлен алгоритм создания упражнения.

На рисунке 13 представлен алгоритм проверки решения упражнения.

Рисунок 12 – Блок схема алгоритма создания

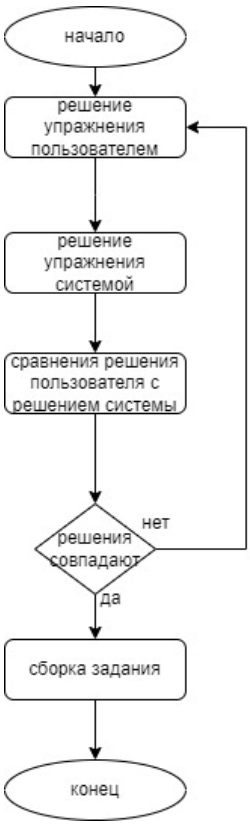


Рисунок 13 – Блок схема алгоритма проверки решения

Алгоритм на рисунке 12 выполняется следующим образом: пользователь выбирает уровень сложности упражнения, после чего происходит генерация упражнения, затем следует запись в базу данных об этом упражнении, после чего происходит сборка упражнения.

Алгоритм на рисунке 13 выполняется следующим образом: пользователь решает упражнение, после чего система решает упражнение. Решение пользователя сравнивается с решением системы, и если оно совпадает с системным решением, то упражнению присваивается новый статус – решено.

На рисунке 14 изображен алгоритм отрисовки упражнение на веб-странице

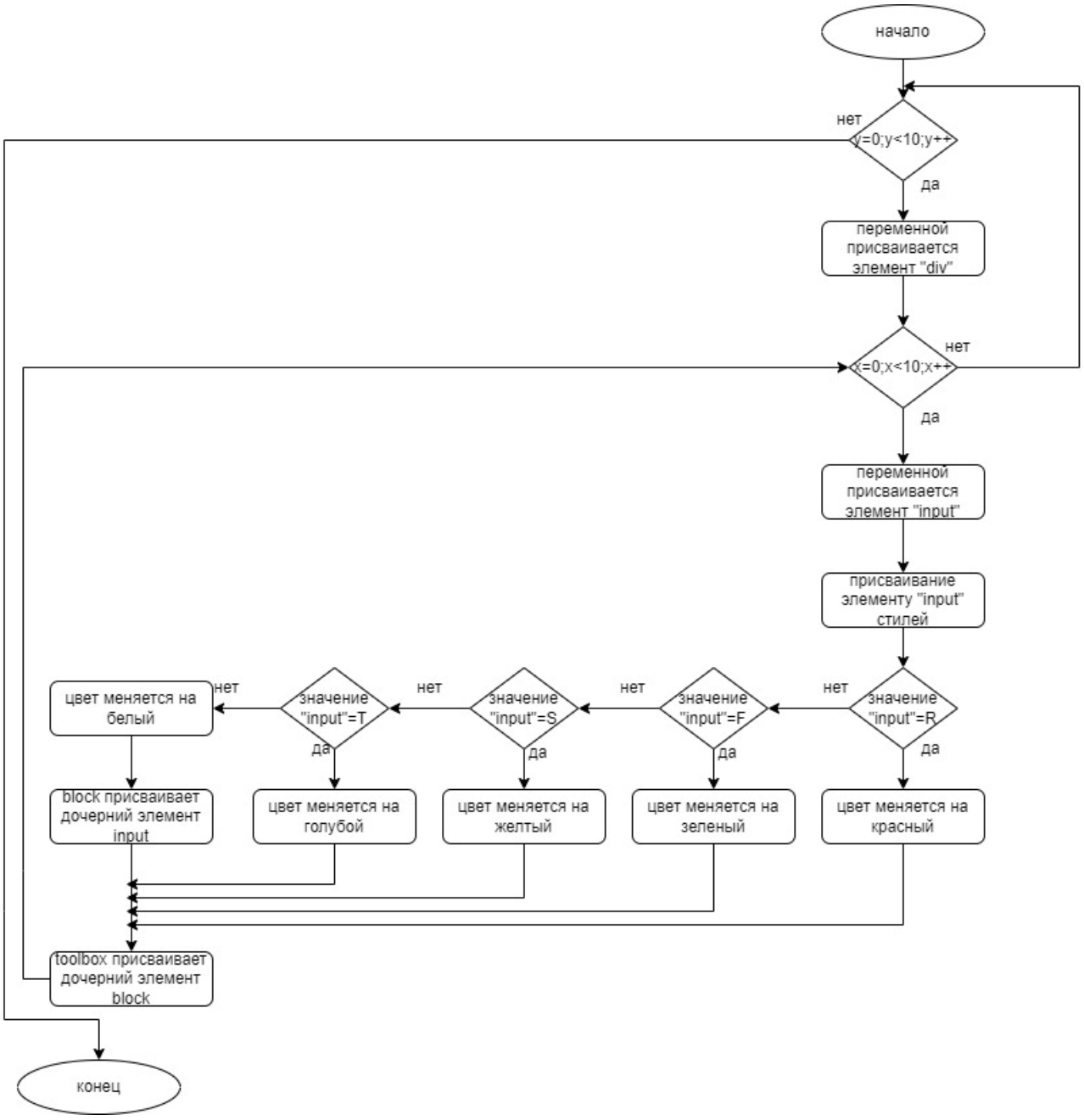


Рисунок 14 – Блок схема алгоритма отрисовки упражнения

На рисунке 14 происходит инициация цикла for, в котором шаг +1, от нуля до десяти присваиваем переменной элемент div, после начинается вложенный цикл for с шагом +1, от нуля до десяти, в котором переменной присваивается элемент input, после, присваиваются стили элементу input, следующим шагом

идет проверка на значение внутри поля input, если значение равно R, то цвет заднего фона становится красным, если нет, то проверяется значение равное F, тогда, если да, то цвет заднего фона становится зеленым, если нет, то проверяется значение, равным S, если да, то цвет заднего фона меняется на желтый, потом, если нет, то проверяется на значение равное T, если да, то цвет заднего фона меняется на голубой, если нет, то цвет заднего фона устанавливается белым, переменная block принимает дочерний элемент input, после возвращается в начало вложенного цикла и действия повторяются, как только значения вложенного цикла превышают десять, то происходит возврат в первый цикл, тогда начинается второй шаг первого цикла и происходят те же действия, после того как и первый цикл превысил порог – десять, тогда элемент toolbox принимает все значения.

На рисунке 15 представлен алгоритм загрузки упражнения из базы данных для того, чтобы пользователь видел свои выполненные задания

На рисунке 15 видно, что сначала пользователь переходит в каталог, что, в свою очередь, провоцирует отправку GET-запроса к БД, после чего приходит ответ от БД в виде строки, в которой закодированы зашифрованные индексы поля упражнения, следующем шагом идёт разбиение полученной строки на отдельные индексы, далее идет подсчет количества индексов для представления о количестве отрисованных элементов, следующий шаг – это цикл for в котором происходит разбор полученных индексов. Индекс может быть с пометкой или без, если с пометкой, то переменная присваивает элемент <inputfield> со значением метки и задний фон перекрашивается в цвет обозначенный индексом, если индекс без метки, тогда присваивается элемент <inputfield> с пустым значением и белым задним фоном, после чего элемент <inputfield> добавляется на html-страницу и все повторяется столько раз, пока в цикле for переменная count не выйдет за пределы условия и тогда цикл завершается.

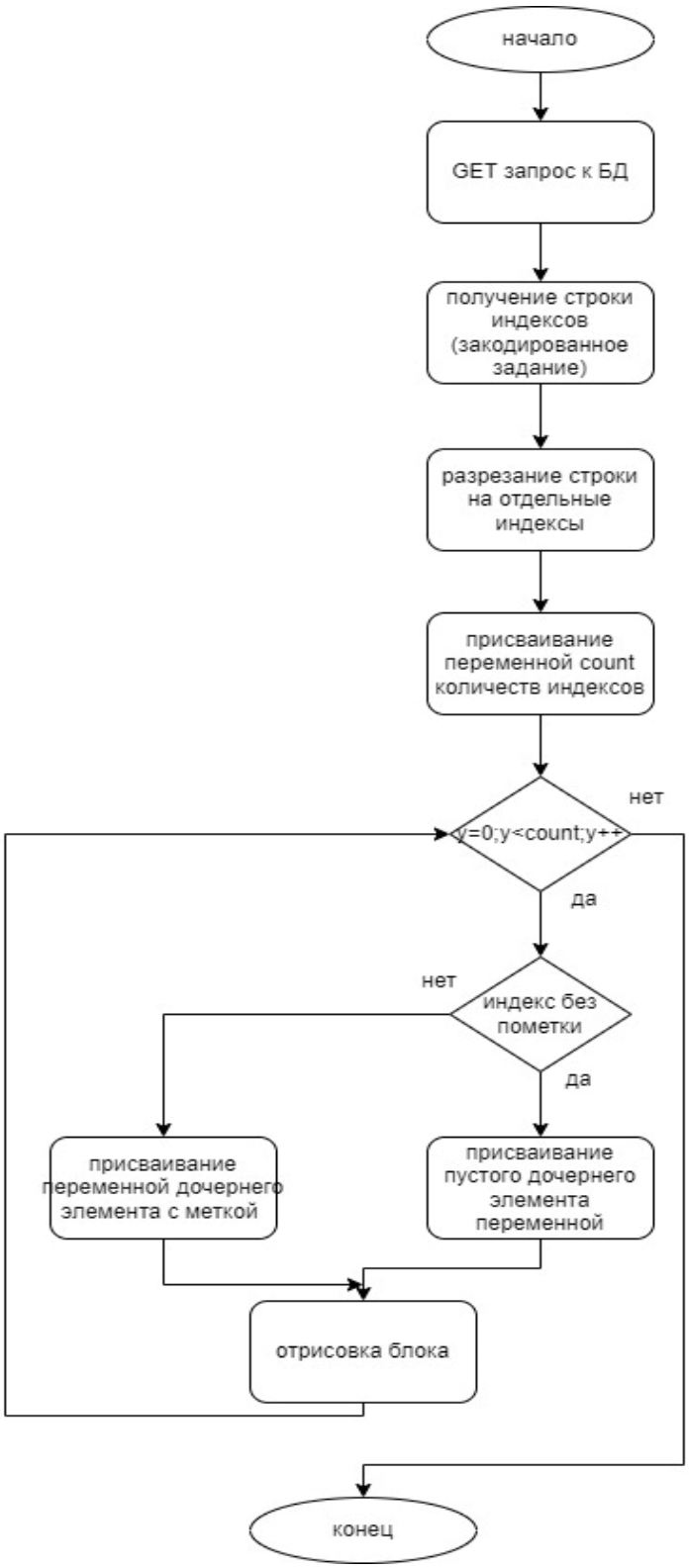


Рисунок 15 – блок схема алгоритма загрузки упражнения

Благодаря всем спроектированным блок схемам и UML диаграммам можно однозначно понять, как ведет себя система при различных действиях

пользователя, какой ответ ожидать, после определенного взаимодействия, как функционируют отдельные модули разрабатываемой системы, например, загрузка заданий или проверка правильности его решения.

2.3. Средства разработки

Средствами разработки стали: phpMyAdmin, Javascript, html и css.

JavaScript — это язык программирования, который используется для разработки большинства интерактивных веб-страниц.

Используется для функционирования логических элементов, создания функций, анимаций на странице. Выбору JavaScript способствовала возможность манипулировать DOM-элементами - прямое управление HTML-структурой и элементами страницы. Открывает безграничные возможности для создания новых html элементов (упражнения для тренировки) [5].

HTML и **CSS** предназначены для графического отображения информации, блоков, картинок на html странице.

phpMyAdmin – позволяет через браузер и не только осуществлять администрирование сервера MySQL, запускать команды SQL и просматривать содержимое таблиц и баз данных [9].

3. Разработка обучающего тренажера

3.1. Среда разработки

Первая задача – выбор среды разработки. Есть множество различных сред, которые подходят для решения разных задач. Выбором стала среда разработки Visual Studio Code так как имеет ряд преимуществ: Быстрая и легкая смена языка разработки, что является весомым аргументом при разработке веб-сервисов и веб-сайтов, так как такой процесс создания затрагивает несколько языков программирования, таких как JavaScript и его различные фреймворки, php, язык разметки - html и язык стилей – css. Visual Studio Code позволяет установить нужную библиотеку любого языка и сразу же начать работу, что и стало причиной выбора этой среды разработки.

3.2. Процесс разработки

Любой разрабатываемый продукт содержит интерфейс, через который пользователь может взаимодействовать с системой. Спроектирован следующий макет веб-сайта: на рисунке 16 представлена заглавная страница

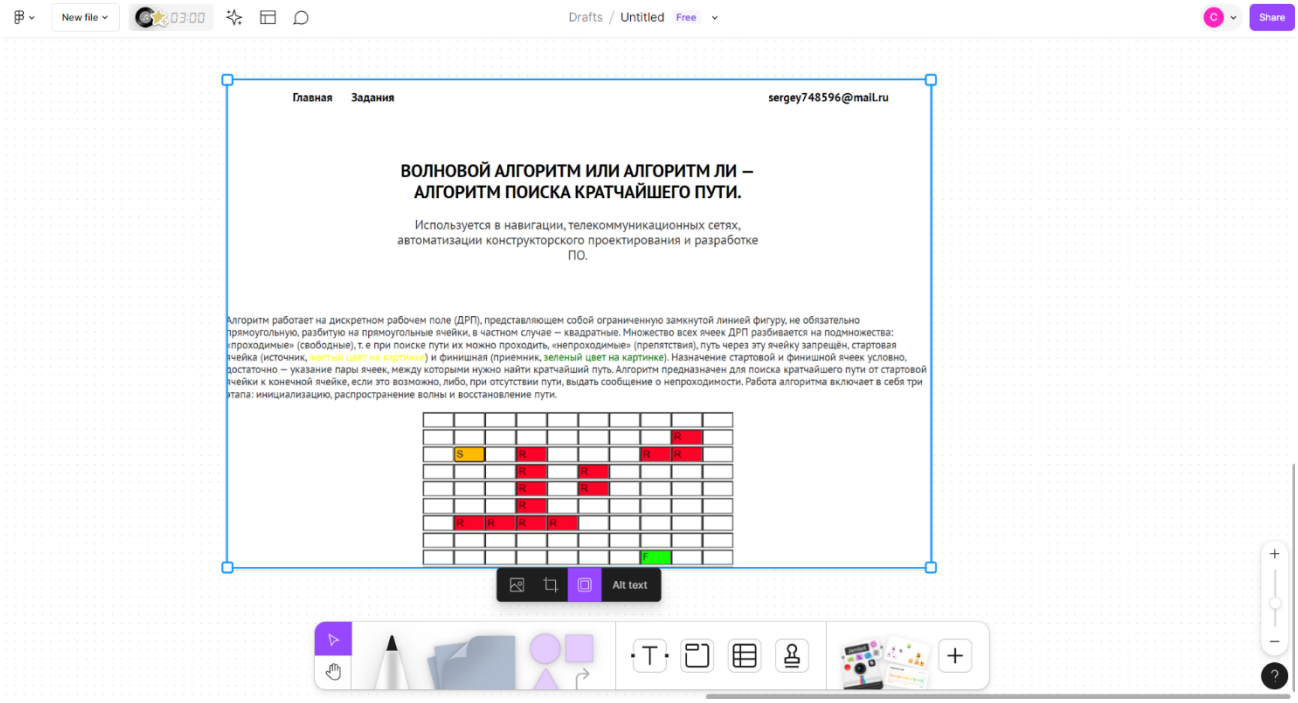


Рисунок 16 – Первая страница веб-сайта

На рисунке 16 представлена главная страница веб-ресурса, которая включает в себя заголовок (header), в котором находятся 3 кнопки: Главная, Задания, Регистрация. Основное наполнение страницы представляет собой теоретические сведения об алгоритме Ли. Низ страницы – футер (footer), который наполнен справочной информацией: ссылками на интернет-ресурсы.

На рисунке 17 представлена вторая страница сайта (Задания)

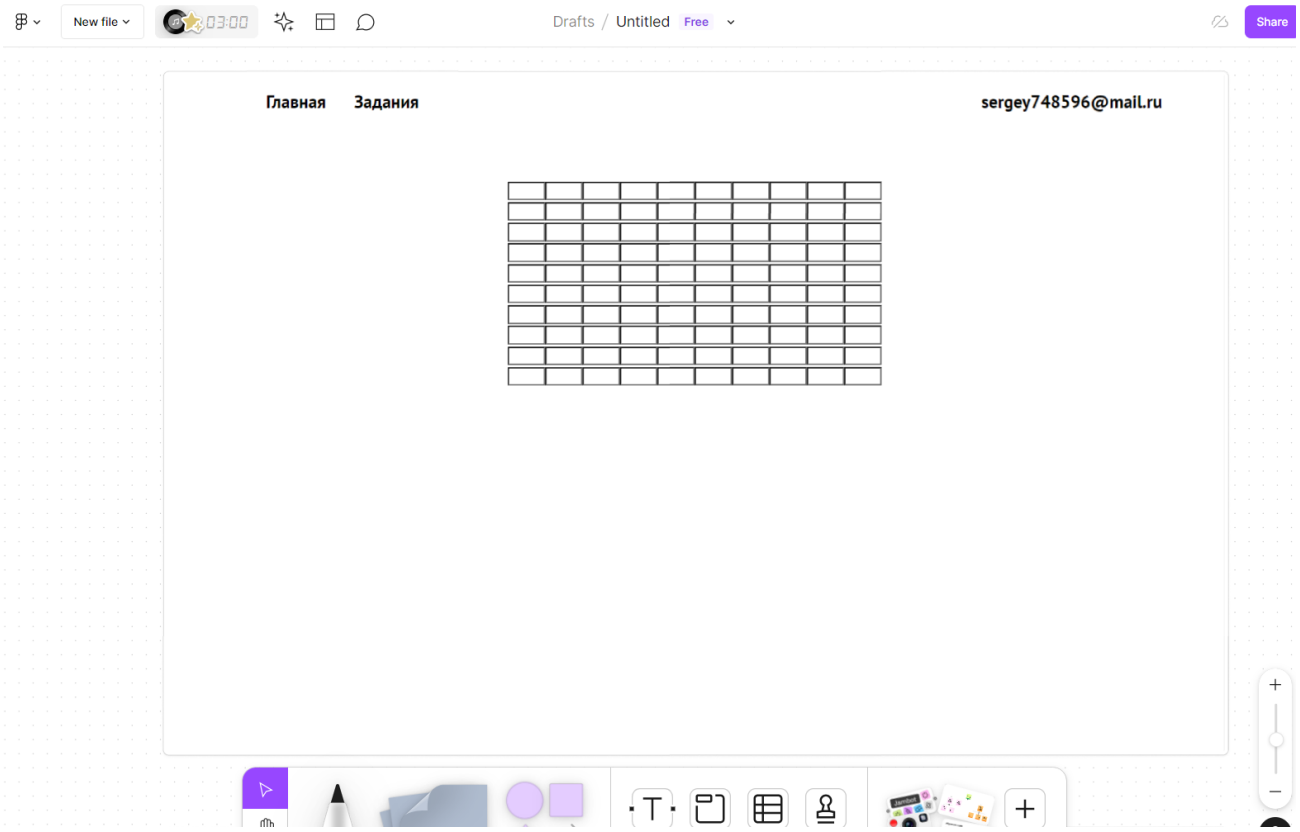


Рисунок 17 – Страница Задания

Рисунок 17 содержит наполнение второй страницы: header, состоящий из 3 кнопок Главная, Задания, адрес электронной почты, который говорит о том, что пользователь авторизован, тело страницы – само задание и footer.

На рисунке 18 представлена страница регистрации

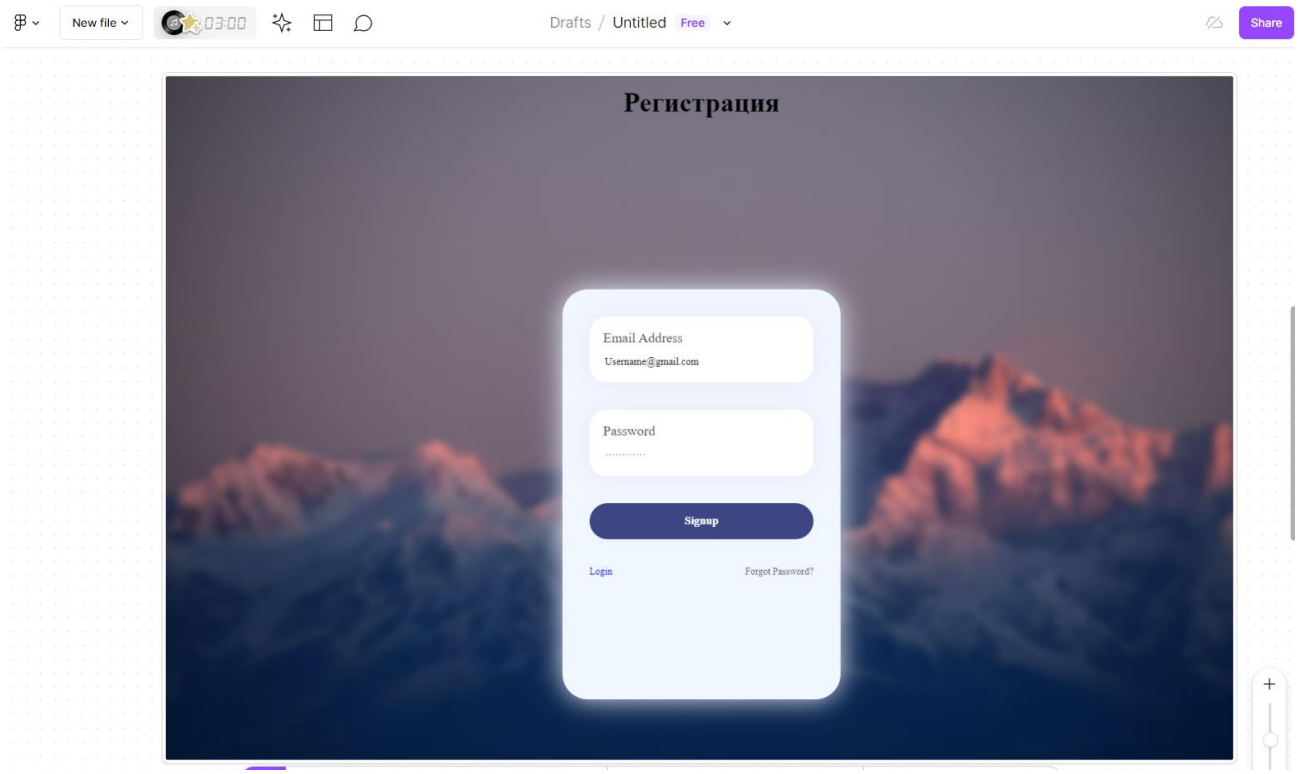


Рисунок 18 – Страница регистрации

На рисунке 19 представлена страница для авторизации

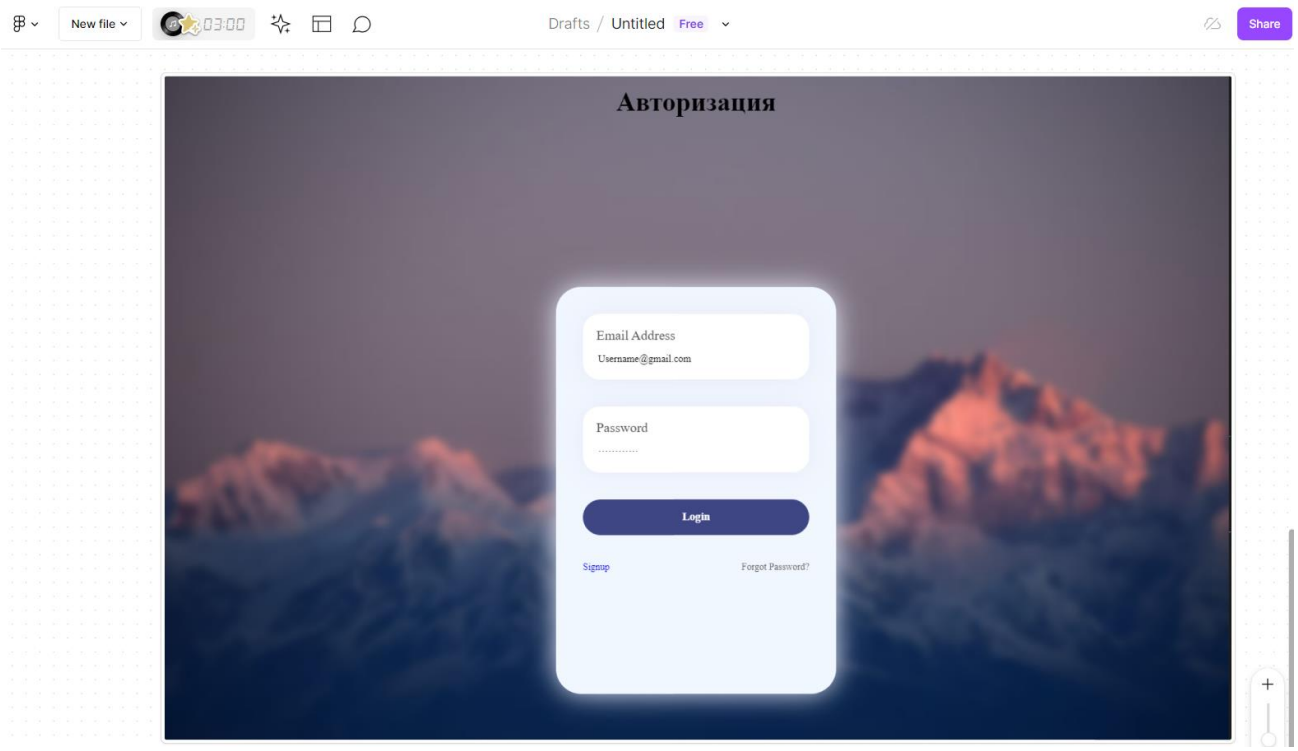


Рисунок 19 – Страница авторизации

На рисунке 18 отображено наполнение страницы для регистрации: форма, которая содержит в себе два поля для ввода данных (почта и пароль), одну кнопку вход и две ссылки (login и Forgot Password?) для перехода на страницу авторизации и смены пароля.

Рисунок 19 состоит из тех же элементов, что и страница регистрации (см. рис. 18) за исключением кнопки Login для авторизации и ссылки Signup для перехода на страницу регистрации

На рисунке 20 представлены кнопки, используемые на сайте

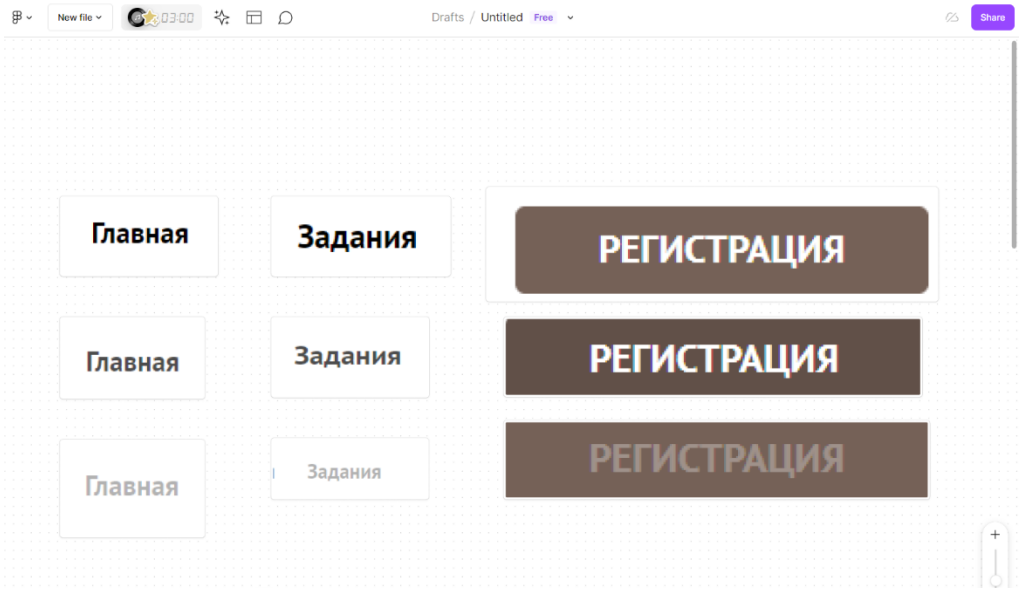


Рисунок 20 – Макет кнопок веб-сайта

На рисунке 20 отображены кнопки Главная, Задания и Регистрация. Каждая кнопка может находится в трех состояниях: покой, фокус, нажатие. Покой, когда элемент просто отображается, без воздействия со стороны пользователя. Фокус, когда пользователь наводит курсор на элемент на странице. Нажатие, когда пользователь производит клик по кнопке. У каждого из этих положений различные стили для удобства пользователя.

На рисунке 21 представлено Упражнение.

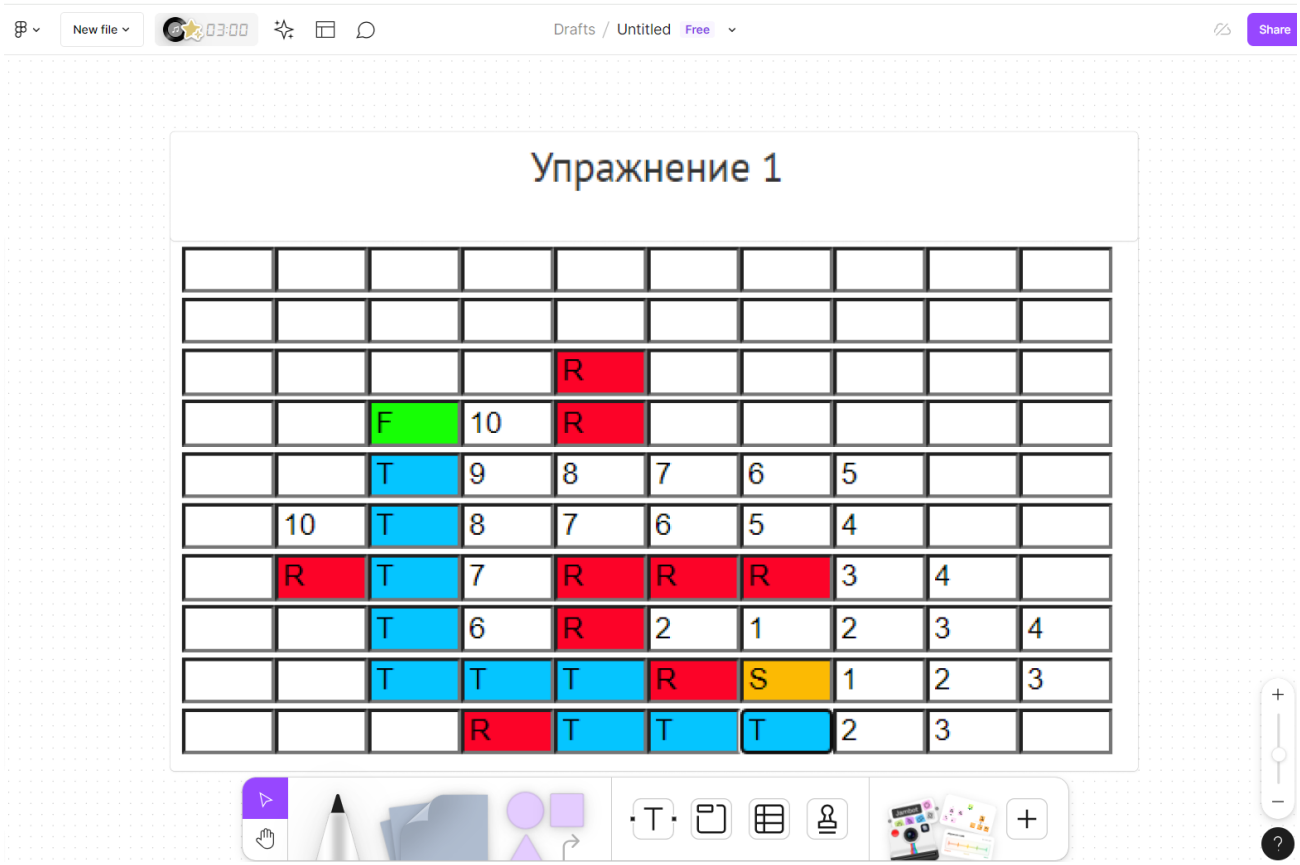


Рисунок 21 – Упражнение

На рисунке 21 представлен образ упражнения. Упражнение состоит из блоков поле для ввода, где белые поля без цифр – пустые клетки, белые поля с цифрами - распространяема волна, красные клетки с буквами – преграда, синие клетки с буквами – трасса, желтая клетка с буквой – начальная позиция, зеленая клетка с буквой – конечная позиция. Цвета выбраны в соответствии с согласованностью интерфейса, когда интуитивно понятно, что красный цвет стоит избегать, поэтому преграда красного цвета, зеленый ассоциируется с сигналом к побуждению действий, из-за чего финиш был выбран зеленого цвета.

На рисунке 22 представлен весь макет целиком.

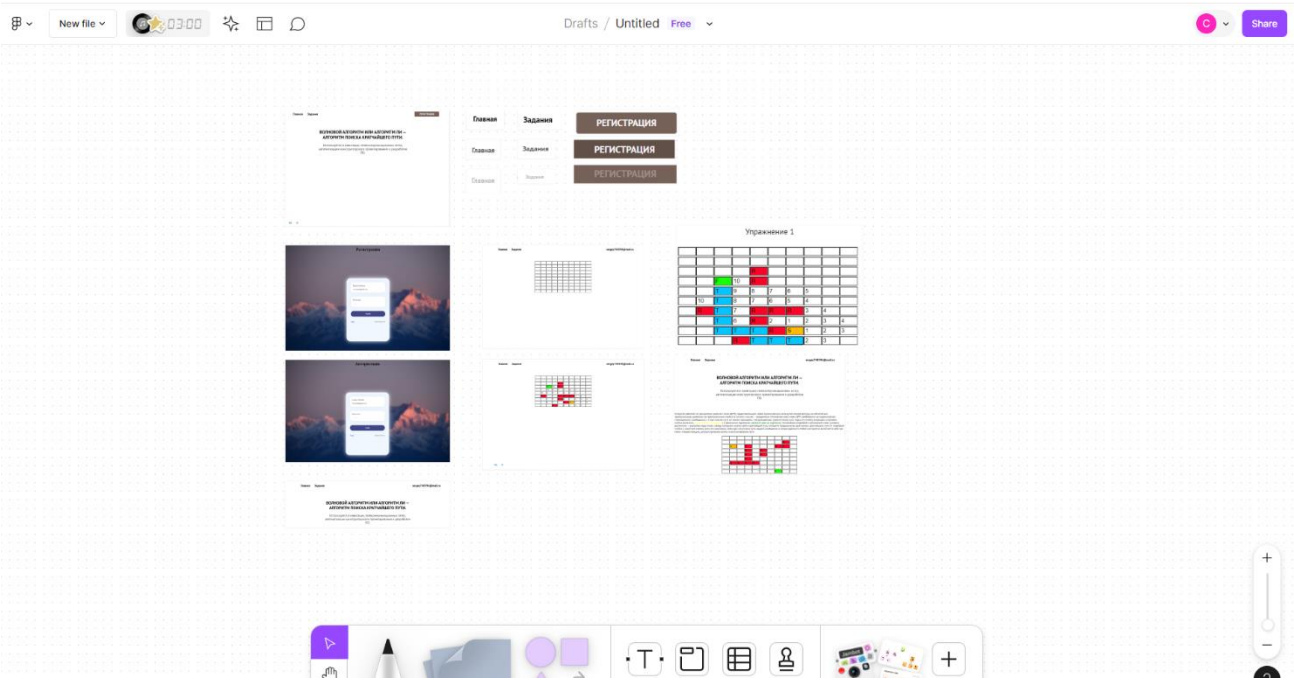


Рисунок 22 – Полный макет веб-сайта

Все элементы интерфейса на рисунке 22 разработаны на языке html. Заголовочный элемент – закрывающийся тег <header>, содержимое страницы – закрывающийся тег <body>, низ страницы – закрывающийся тег <footer>, кнопки – тег <button>, ссылки – тег <a>, текстовые блоки – закрывающийся теги , , <p>, упражнение – тег , состоящий из блоков - тег <inputfield> и названия упражнения – тег <p>.

Для того, чтобы была возможность регистрироваться, авторизовываться, создавать и сохранять задания, необходимо хранить информацию о пользователях и их заданиях в базе данных. Для создания и администрирования базы данных используется среда XAMPP, которая предоставляет сервер, на котором размещается веб-сайт и средство управления базой данных, где размещена таблица, рисунок 23.

Для подключения веб-сайта к базе данных, представленной на рисунке 23, используется модуль DataBaseConnection.php.

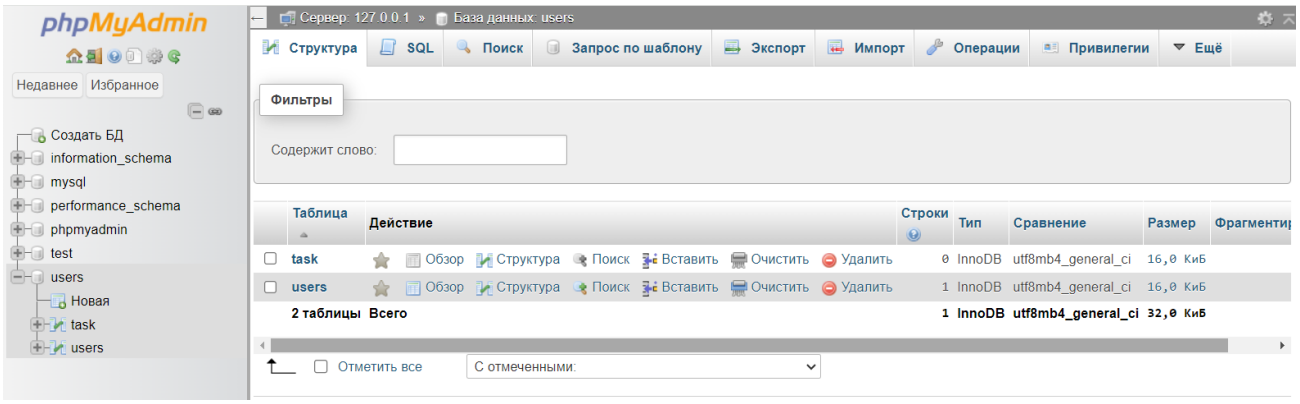


Рисунок 23 – Панель управления базой данных

В этом модуле использована следующая встроенная функция: $link = mysqli_connect($hostname, $username, $password, $databasename);

mysqli_connect принимает четыре параметра: $hostname – адрес расположения базы данных, $username – стандартизированное имя пользователя, используемое для предоставления доступа к базе данных, $password – стандартизированный пароль, используемый для предоставления доступа к базе данных, $databasename – имя созданной базы данных.

Для регистрации пользователя на веб-сайте используется модуль reg.php. Внутри него первым шагом идет подключение модуля databaseconnection.php с помощью операции include(); далее следует процедура формирования пост запроса для отправки данных пользователя в базу данных. Первым шагом проверяется, что поля заполнены: if(!isset($_POST["email"]) || !isset($_POST["password"])) {

return;

}

Внутри круглых скобок находятся условия, которые проверяются: (!isset($_POST["email"]) означает – если значение email пустое или, дословно, не установлено, то происходит ожидание ввода символов. Это сделано для того, чтобы база данных не заполнялась пустыми полями. Следующим шагом является запись вводимых значений в переменные. После чего с помощью встроенной

функции md5(); производится хеширование пароля и уже после того, как пароль захешировался, отправляется запрос INSERT INTO, который меняет поля email и password в таблице users.

За авторизацию пользователя отвечает модуль auth.php, который также включает в себя операцию include(); для подключения к базе данных, после чего, ожидает ввода данных от пользователя, записывает полученные данные в переменные, так как пароль в базе данных захеширован, то и при авторизации, введенный пароль хешируется, после чего сравнивается с хешированным паролем в таблице, найденным по полю email, введенным пользователем и, если пароли совпадают, то пользователя переносит на главную страницу, если поля не заполнены или совпадения с базой данных не найдены, тогда пользователя уведомит об этом, и ему нужно будет снова ввести свои данные для повторной попытки авторизации.

За создание упражнения отвечает модуль createtask, внутри которого реализован конструктор html элементов с помощью языка программирования JavaScript. Код представлен ниже:

```
for (let y=0;y<10;y++){
    block = document.createElement('div');
    for (let x=0;x<10;x++){
        inp = document.createElement('input');
        inp.type = "text";
        inp.style.background = "#FFFFFF"
        block.margin_left = "auto";
        block.margin_right = "auto";
        inp.id = y.toString() + x.toString();
        inp.size = "1"
        inp.addEventListener('input', function(){
            if (this.value === "R"){
```

```
                this.style.background = "#fc0328"
            }
            else if(this.value === "F") {
                this.style.background = "#16ff05"
            }
            else if(this.value === "S") {
                this.style.background = "#fcba03"
            }
            else if(this.value === "T")
                this.style.background = "#05c5ff"
            else {
                this.style.background = "#FFFFFF"
            }
        })
        block.appendChild(inp);
    }
    toolb = document.getElementById("toolbox")
    toolb.appendChild(block)
    }
document.getElementById('savebtn').addEventListener('click', function(event){
    let input_arr = [];
    for (let y=0;y<10;y++){
        let linetext = ""
        for (let x=0;x<10;x++){
            linetext = linetext + document.getElementById(y.toString() +
x.toString()).value + ','
        }
        input_arr.push(linetext);
```

```
    }
})
```

Сначала создается цикл for, который работает с шагом +1, от нуля до десяти, и создает блок , с помощью встроенной функции в JavaScript - document.createElement('div'); далее производится вход во второй цикл for, который также шагает с шагом +1, от нуля до десяти, и рисует блок <inputfield>, с помощью встроенной функции в JavaScript - inp = document.createElement('input'); таким образом с помощью вложенного цикла for получается двумерный массив 10x10 элементов , с элементами <inputfield> в каждом из элементов . Следующим шагом является присвоение атрибутов к <inputfield>: inp.type = "text"; - в поле для ввода поддерживается текстовый формат данных, inp.style.background = "#FFFFFF" – задний фон поля для ввода – белый, block.margin_left = "auto"; и block.margin_right = "auto"; нужны для удаления пробелов между блоками, чтобы поле упражнения выглядело как одна система, inp.id = y.toString() + x.toString(); служит для присваивания идентификационного номера клетки поля, чтобы его можно было сохранить в базе данных, inp.size = "1" – размер поля ввода, inp.addEventListener('input', function() – добавление слушателя событий, который будет реагировать на взаимодействия с полем, if (this.value === "R") – проверка условия, если значение блока равно R, this.style.background = "#fc0328". – то значение цвета заднего фона устанавливается на красный, else if(this.value === "F") – иначе, если значение поля равно F, this.style.background = "#16ff05" – то значение цвета заднего фона устанавливается на зеленый, else if(this.value === "S") – инча, если значение равно S, this.style.background = "#fcba03" – то значение цвета заднего фона устанавливается на желтый, else if(this.value === "T") – иначе, если значение равно T, this.style.background = "#05c5ff" - значение цвета заднего фона устанавливается на голубой, block.appendChild(inp); - устанавливает дочерним элементом блок <inputfield> в блок block, toolb =

document.getElementById("toolbox") – находим на html странице элемент с идентификатором toolbox, toolb.appendChild(block) – устанавливаем в конец элемента toolb, элемент block.

Следующая часть алгоритма отвечает за сохранение поля для отправки в базу данных, document.getElementById('savebtn').addEventListener('click', function(event) – находим элемент по идентификатору savebtn, добавляет слушателя событий на нажатие кнопки, let input_arr = []; - объявление массива, for (let y=0;y<10;y++) начинается цикл for с шагом +1, от нуля до десяти, let linetext = "" – объявляется переменная linetext с пустым значением, for (let x=0;x<10;x++) – начинается второй цикл for с шагом +1, от нуля до десяти, linetext = linetext + document.getElementById(y.toString() + x.toString()).value + ',' – переменная linetext присваивает значение самой себя, сложенной идентификаторами блока inputfield, разделяя значения в строке пробелами. input_arr.push(linetext); - производится запись в массив. Аналогичным образом, только в обратной последовательности, происходит отрисовка поля на веб-странице из базы данных с помощью модуля tasklist, сначала с помощью GET запроса считывается массив со строкой идентификаторов, строка делится на отдельные идентификаторы, а дальше происходит отрисовка каждого элемента согласно идентификационному номеру, как отображено на рисунке 15, где в зависимости от индекса переменной присваивается элемент <inputfild> и устанавливается значение для поля внутри элемента <inputfield>, либо пустое значение, либо цифровое значение, либо буквенное значение, где каждому из этих значений принадлежит свой цвет заднего фона, что отображено на рисунке 24.

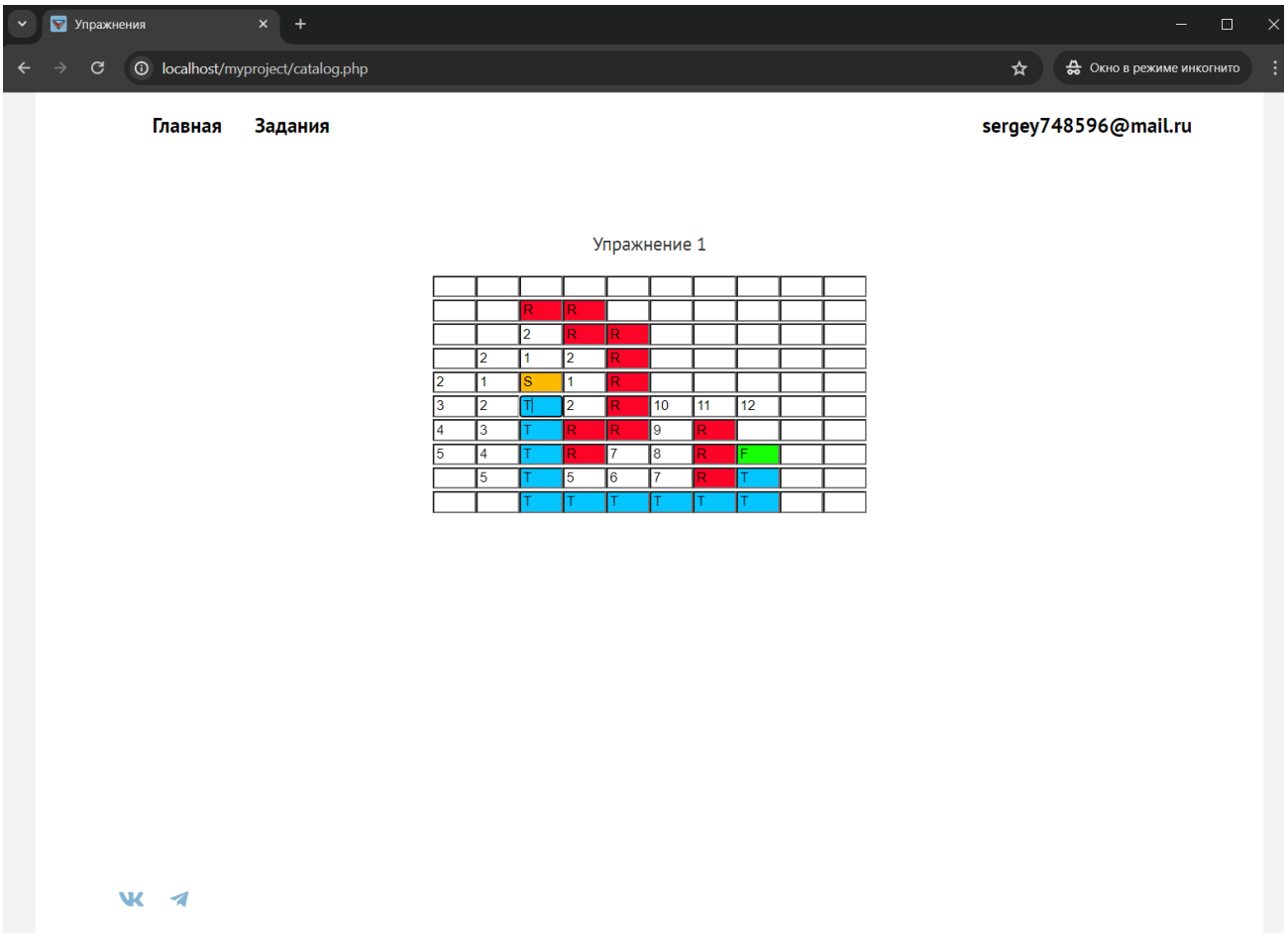


Рисунок 24 – Отображение упражнения

На рисунке 24 представлено отрисованное задание, где пустые клетки – это <inputfield> с пустым значением, клетки с буквенными обозначениями и цветным задним фона - <inputfield> со значением указанных в них буквах, а клетки с цифровыми обозначениями - <inpurfield> с цифровым значением.

4. Проверка работоспособности

При первичном запуске веб-сайта таблица базы данных с пользователями пустая, что отражено на рисунке 25.

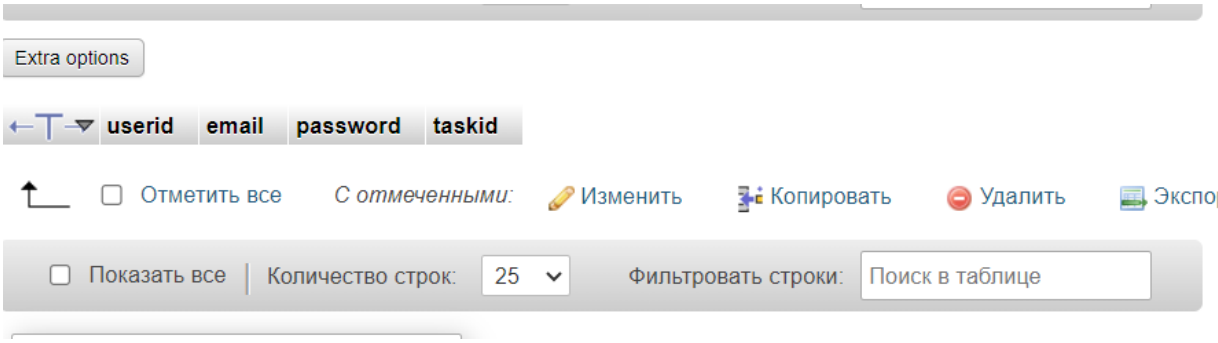


Рисунок 25 – Пустая таблица базы данных

На рисунке 25 видно, что записи о пользователях отсутствуют.

Следующий шаг – регистрация. Процесс показан на рисунках 26, 27, 28, 29.

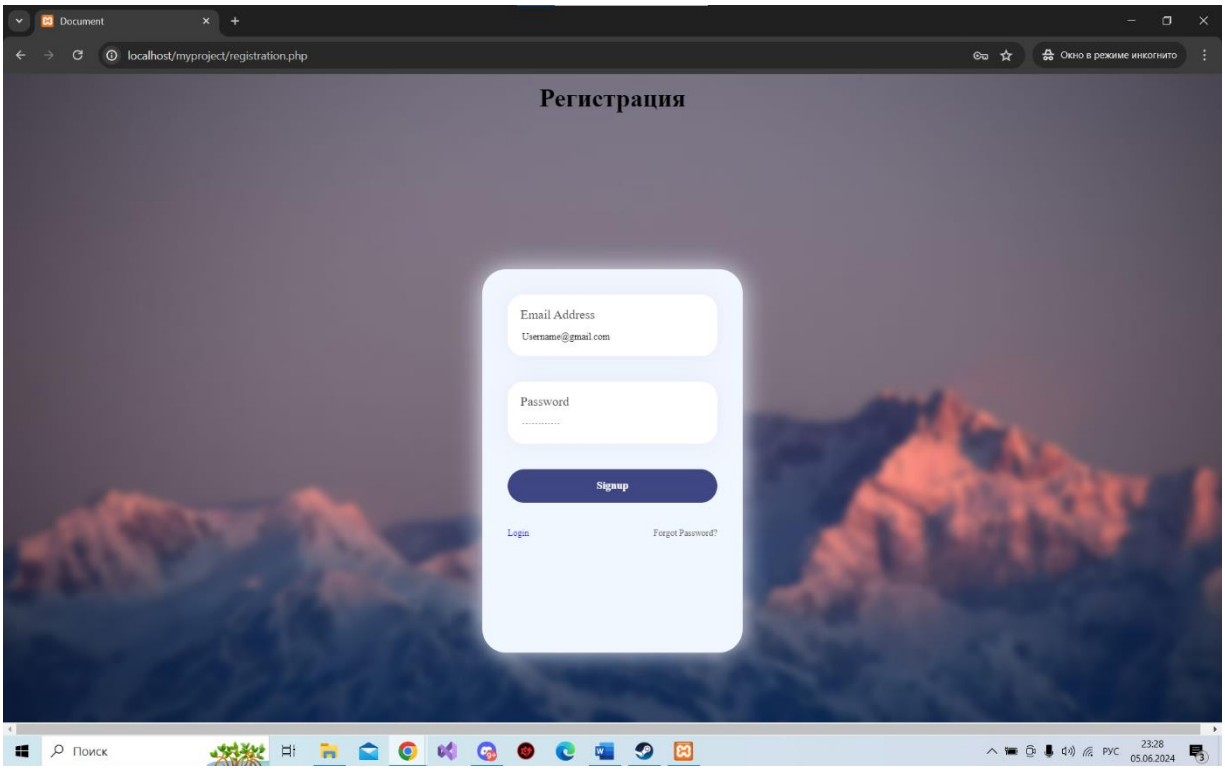


Рисунок 26 – Нажатие на кнопку регистрации

На рисунке 26 отражено, что после нажатия кнопки регистрации происходит перенос пользователя на страницу регистрации, где находится форма для заполнения.

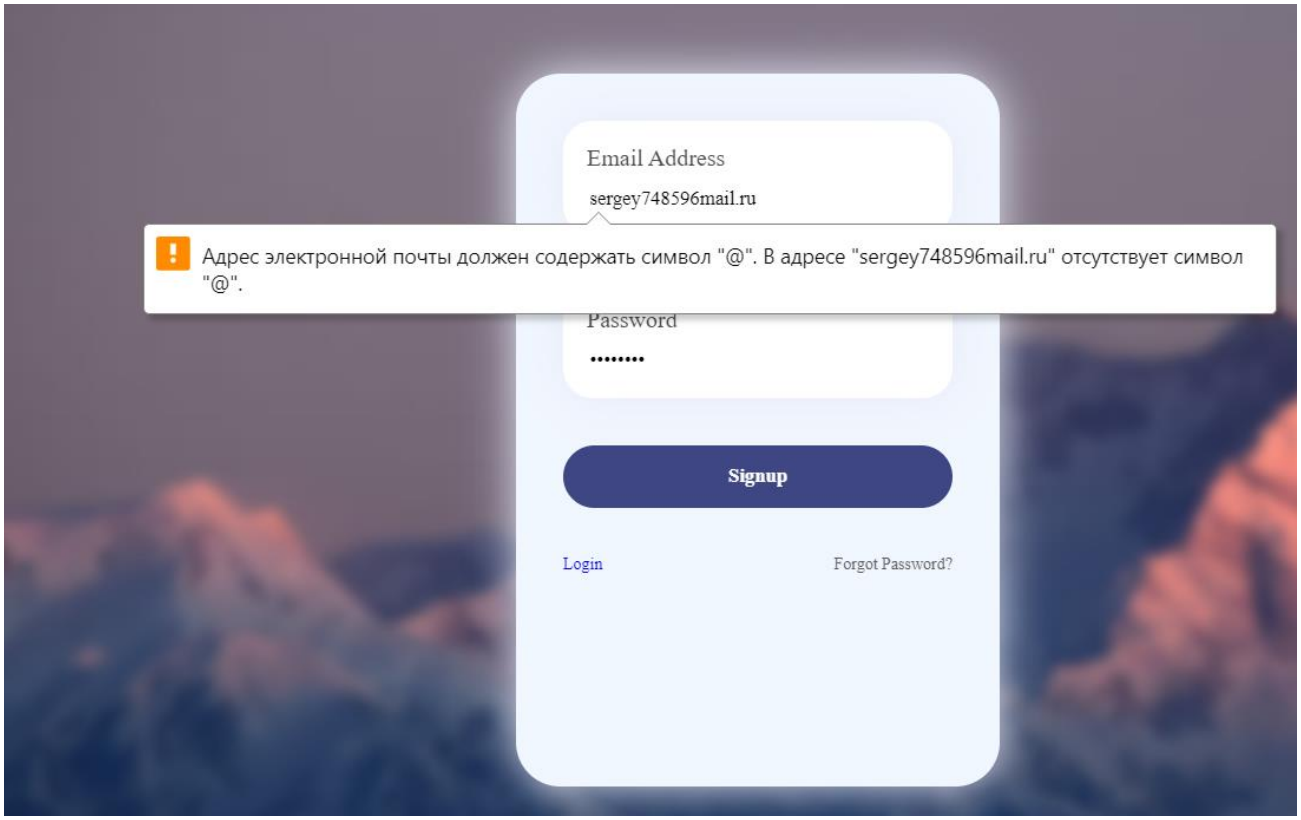


Рисунок 27 – Уведомление об ошибке

При попытке неправильно ввести данные в форму пользователю не позволяют зарегистрировать аккаунт, так как почта введена без специального символа, о чем уведомляют на рисунке 27.

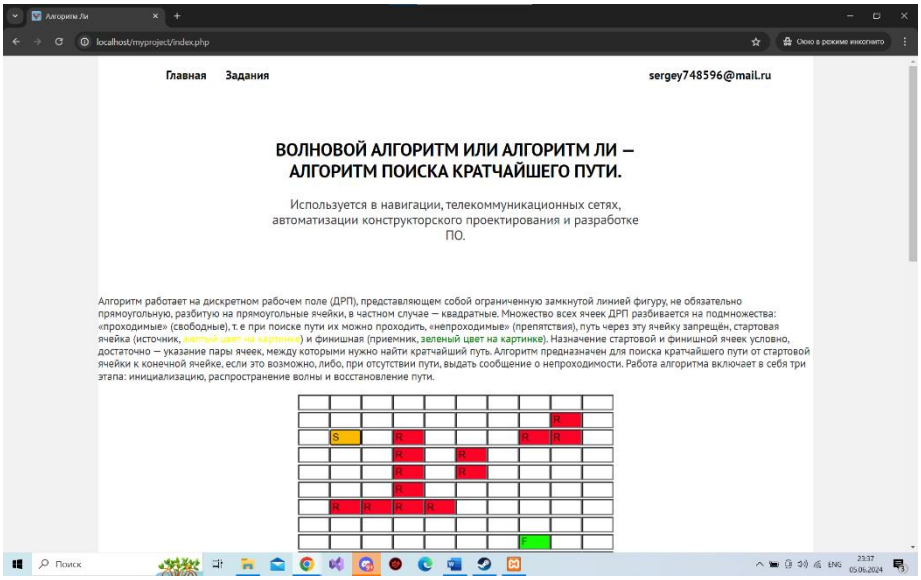


Рисунок 28 – Успешная регистрация

На рисунке 28 видно, что после успешной регистрации на сайте в правом верхнем углу, где была кнопка регистрации теперь находится почта-логин авторизованного пользователя.

После успешной регистрации ожидается появление записи в таблице с пользователями, рисунок 29.

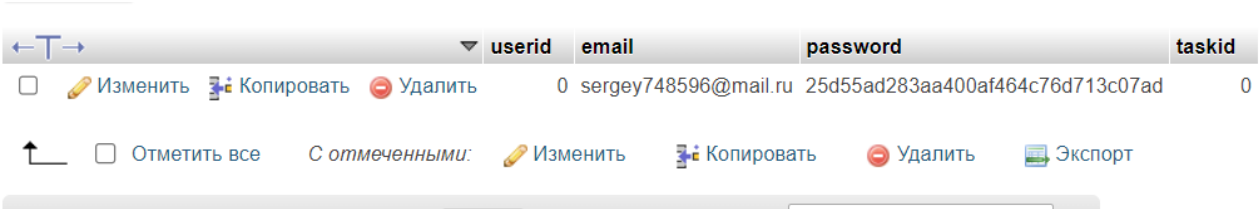


Рисунок 29 – Состояние таблицы

На рисунке 29 видно, что в таблице появилась новая запись о пользователе с почтой, такой же, как и на рисунке 28, с присвоенным идентификатором, а также видно, что в столбце password находится зашифрованный пароль, что способствует безопасному хранению данных пользователей.

На рисунке 30 отображено, что происходит при нажатии на кнопку задания.

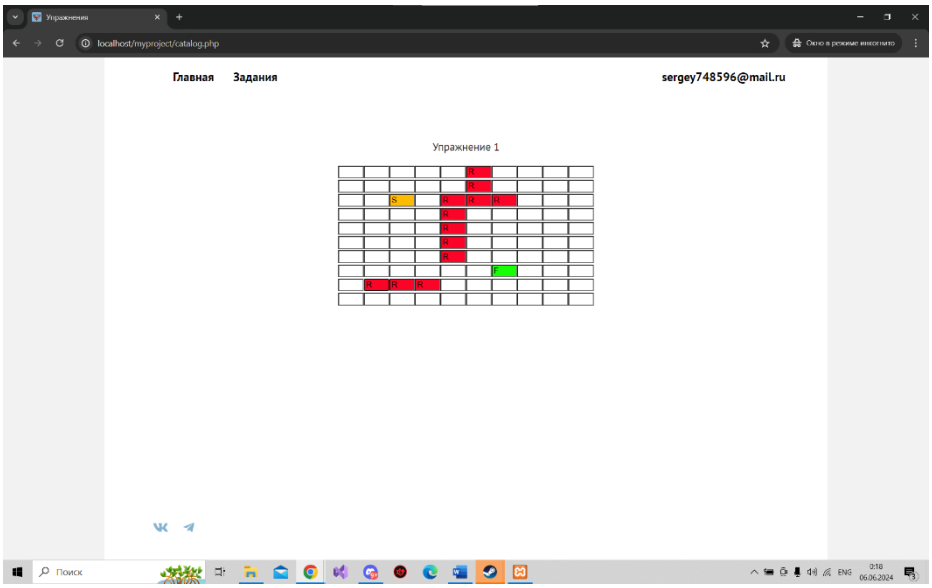


Рисунок 30 – Отображение упражнения

При переходе в каталог заданий подгружаются упражнения пользователя, которые можно начать решать, рисунок 30.

На рисунке 31 представлено, что функция распространения волны работает исправно.

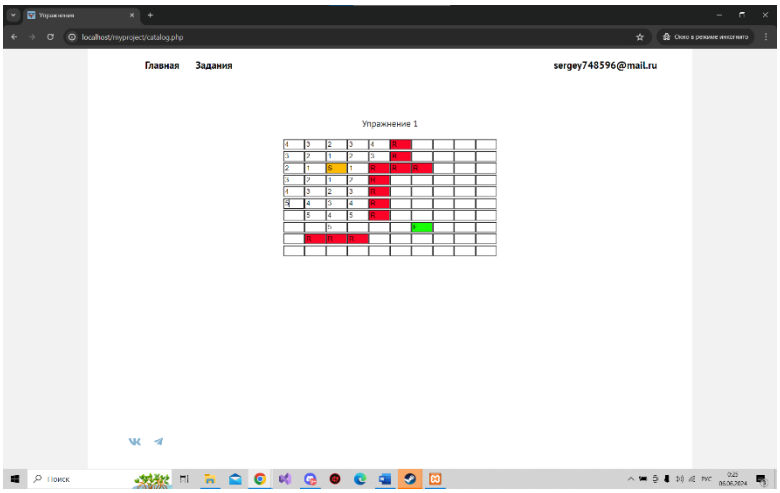


Рисунок 31 – Распространение волны

На рисунке 31 видно, что пользователь может задавать значения клеткам и прописывать индекс для каждой из них, что соответствует номеру волны.

На рисунке 32 отображено, что пользователь может прокладывать трассу.

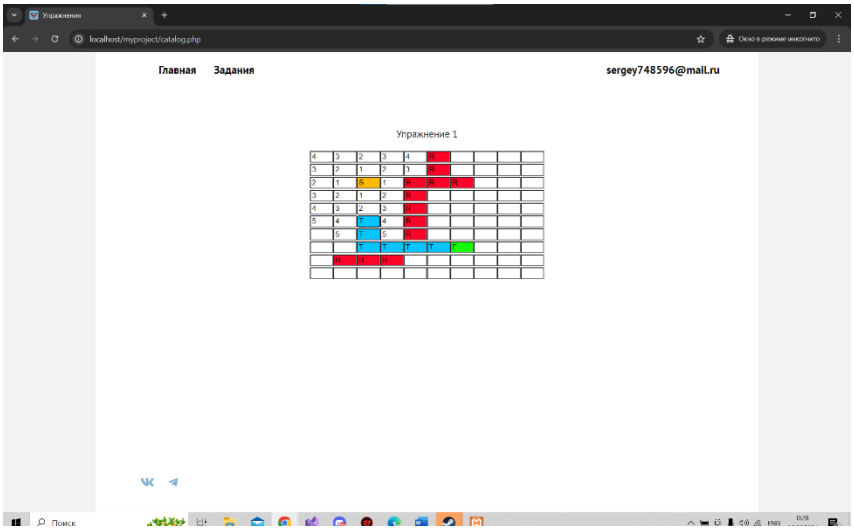


Рисунок 32 – Прокладка трассы

На рисунке 32 представлено, что пользователю доступна возможность прокладки трассы.

Таким образом, основные функции проверены, что свидетельствует о работоспособности тренажера для изучения волнового алгоритма.

Заключение

Результатом работы является обучающий тренажер для изучения волнового алгоритма с интуитивно понятным пользовательским интерфейсом. Разработанная система позволяет решать различные сгенерированные упражнения и сразу получать результат проверки, проделанной работы.

Чтобы пользователь мог постоянно проходить разные упражнения, реализована функция добавления новых заданий.

В дальнейшем систему можно легко расширить различными усложнениями имеющегося генератора упражнений или создать новый конструктор других упражнений, как и для волнового алгоритма, так и для других дисциплин в целом. Так же, возможно, расширение взаимодействий пользователя с тренажером, путем включения в эти взаимодействия 3-их лиц, к примеру, преподавателей, у которых будет возможность, создавать задания для студентов и следить за их результатами.

Таким образом требования технического задания выполнены в полном объёме, в соответствии с календарным планом.

Список используемой литературы.

1. Глотова Т. В. Объектно-ориентированная методология разработки сложных систем [Электронный ресурс] // Электронная библиотека ПГУ – URL: https://elib.pnzgu.ru/library/1715175271 (дата обращения: 08.05.2024)

2. Глотова Т. В. Объектно-ориентированная методология разработки сложных систем [Электронный ресурс] // Электронная библиотека ПГУ – URL: https://elib.pnzgu.ru/library/1715175271 (дата обращения: 12.05.2024)

3. Исаев С.Д., Гармаш Р.П., Эпп В. В. Обзор программных средств для изучения алгоритмов трассировки печатных плат // Новые информационные технологии и системы. – М.: НИТИС, 2023. – С. 26-27

4. Обход препятствий: волновой алгоритм (Алгоритм Ли) [Электронный ресурс]. – Режим доступа: https://suvitruf.ru/2012/05/13/1176/volnovoj-algoritm-algoritm-li/ (дата обращения: 7.04.2024)

5. Руководство по JavaScript [Электронный ресурс]. – Режим доступа: https://metanit.com/web/javascript/ (дата обращения: 16.05.2024)

6. Торотоева А. М. Цифровизация общества: проблемы реализации и новые возможности. (Обзор) // Социальные и гуманитарные науки. Отечественная и зарубежная литература. Серия 11: Социология. -2021. - № 1. – С. 118 – 132. DOI: 10.31249/rsoc/2021.01.07

7. Drawio [Электронный ресурс]. – Режим доступа: https://app.diagrams.net/ (дата обращения: 14.05.2024)

8. ЭИОС ПГУ [Электронный ресурс]. – Режим доступа: https://moodle.pnzgu.ru/my/ (дата обращения: 20.05.2024)

9. Requirements [Электронный ресурс]. – Режим доступа: https://docs.phpmyadmin.net/en/latest/require.html#php (дата обращения: 10.05.2024)

Приложение А

Разработка обучающего тренажера для изучения волнового алгоритма

Листинг программы

Файл DataBaseConnection.php

```
<?php

$hostname = "localhost";
$username = "root";
$password = "";
$databaseName = "users";

$link = mysqli_connect($hostname, $username, $password, $databaseName);

if ($link == false){

   echo("DATABASE ERROR");

}
 ?>
```

Файл auth.php

```
<?php


include("DataBaseConnection.php");

session_start();

$email = $_POST["email"];
$password = $_POST["password"];

if($email == "" || $password == "") {
        echo("Установите email или password");
        return;
}

if ($link){

        $hashedPassword = md5($password);

        // Выполняем запрос к базе данных
        $query = "SELECT * FROM users WHERE email='$email' AND
password='$hashedPassword'";
```

```
        $result = mysqli_query($link, $query);

        // Проверяем результат запроса
        if (mysqli_num_rows($result) == 1) {
            // Пользователь найден
            $user = mysqli_fetch_assoc($result);
            // Создаем сессию
            session_start();
            // Сохраняем в сессию данные пользователя
            $_SESSION['user_id'] = $user['id'];
            $_SESSION['email'] = $user['email'];
            // Записываем куки
        setcookie("user", $_POST['email'], time() + 3600 * 24 * 100, "/", "", 0);
            // Перенаправляем пользователя на главную страницу
            header("Location: ../index.php");
            die();
        } else {
            // Пользователь не найден
            echo "Неверный логин или пароль";
        }


    // Перенаправляем на главную страницу
    header("Location: ../login.php");
    exit();


}
else {
    echo("3");

}



?>
```

Файл createtask.js

```
        for (let y=0;y<10;y++){
            block = document.createElement('div');

            for (let x=0;x<10;x++){
```

```
        inp = document.createElement('input');
        inp.type = "text";
        inp.style.background = "#FFFFFF"
        block.margin_left = "auto";
        block.margin_right = "auto";
        inp.id = y.toString() + x.toString();
        inp.size = "1"
        inp.addEventListener('input', function(){
            if (this.value === "R"){
                this.style.background = "#fc0328"
            }
            else if(this.value === "F") {
                this.style.background = "#16ff05"
            }
            else if(this.value === "S") {
                this.style.background = "#fcba03"
            }
            else if(this.value === "T")
                this.style.background = "#05c5ff"
            else {
                this.style.background = "#FFFFFF"
            }
        })

        block.appendChild(inp);
    }
    toolb = document.getElementById("toolbox")
    toolb.appendChild(block)

}



document.getElementById('savebtn').addEventListener('click', function(event){
    let input_arr = [];
    for (let y=0;y<10;y++){
        let linetext = ""
        for (let x=0;x<10;x++){
            linetext = linetext + document.getElementById(y.toString() +
x.toString()).value + ','
        }
```

```
            input_arr.push(linetext);
            // console.log(linetext)
            // console.log(linetext.split(',').slice(0,10))
          }
          // console.log(input_arr);
})
```

Файл listcatalog.php

```
<!DOCTYPE html>
<html lang="ru">
 <head>
  <meta charset="utf-8">
  <title>Упражнения</title>
  <link rel="stylesheet" href="styles/styles.css">
  <link rel="icon" href="favicon.ico">
  <link rel="icon" type="image/png" href="images/favicon.png">



 </head>

 <body>
  <?php
  include "Header.php";
   ?>

  <main data-test="main" class="main-container main-inner">


   <section data-test="catalog" class="catalog-hotels">

    </header>




    <!-- <ul  id="hotel-list-id"> -->
     <center><p>Упражнение 1</p></div></center>
    <center><div id="toolbox"></div></center>
```

```
      <!-- </ul> -->

    </section>


  </main>

  <footer data-test="footer" class="page-footer">
    <ul class="socials-list">
      <li class="socials-item">
        <a class="socials-link" href="#">
          <span class="visually-hidden">Вконтакте.</span>
          <svg aria-hidden="true" focusable="false"
xmlns="http://www.w3.org/2000/svg" width="25" height="15" viewBox="0 0
25 15" fill="none"><path fill="#83B3D3" fill-rule="evenodd" d="M24.12
1.8c.16-.54 0-.94-.8-.94H20.7c-.67 0-.98.34-1.15.73 0 0-1.33 3.2-3.22 5.27-
.61.6-.9.8-1.23.8-.16 0-.41-.2-.41-.75v-5.1c0-.66-.19-.95-.74-.95H9.82c-.42 0-
.67.3-.67.59 0 .62.95.76 1.04 2.51v3.8c0 .83-.15.98-.48.98-.9 0-3.06-3.2-4.34-
6.88-.25-.72-.5-1-1.18-1H1.57c-.75 0-.9.34-.9.73 0 .68.89 4.07 4.14 8.55 2.17
3.06 5.23 4.72 8 4.72 1.68 0 1.88-.37 1.88-1v-2.32c0-.74.16-.88.7-.88.38 0
1.05.19 2.6 1.66 1.79 1.75 2.08 2.54 3.08 2.54h2.63c.75 0 1.12-.37.9-1.1-.23-
.72-1.08-1.77-2.2-3.02-.62-.71-1.54-1.48-1.82-1.86-.39-.5-.28-.71 0-1.15 0 0
3.2-4.42 3.54-5.93Z" clip-rule="evenodd"/></svg>
        </a>
      </li>
      <li class="socials-item">
        <a class="socials-link" href="#">
          <span class="visually-hidden">Telegram.</span>
          <svg aria-hidden="true" focusable="false"
xmlns="http://www.w3.org/2000/svg" width="18" height="17" viewBox="0 0
18 17" fill="none"><path fill="#83B3D3" d="M16.79.96.84 7.1c-1.09.44-1.08
1.05-.2 1.32l4.1 1.28 9.47-5.98c.44-.27.85-.13.52.17l-7.68 6.93-.28 4.22c.41 0
.6-.2.83-.41l1.99-1.94 4.13 3.06c.77.42 1.31.2 1.5-.71l2.72-12.8c.28-1.1-.43-
1.61-1.16-1.28Z"/></svg>
        </a>
      </li>

    </ul>


  </footer>
```

```
  </body>


  <script src="Ajax_Jerk/ajaxRequest.js"></script>
  <script src="scripts/CTRL_F.js"></script>
  <script src="алгоритм.js"></script>
</html>
```

Файл index.php

```
<!DOCTYPE html>
<html lang="ru">
  <head>
    <meta charset="utf-8">
    <title>Алгоритм Ли</title>
    <link rel="stylesheet" href="styles/styles.css">
    <link rel="icon" href="favicon.ico">
    <link rel="icon" type="image/png" href="images/favicon.png">
  </head>
  <body>
    <?php
    include "Header.php";
     ?>
    <main data-test="main" class="main-container main-index">
      <h1 class="visually-hidden">Волновой алгоритм.</h1>


      <section data-test="advantages" class="advantages">
        <div class="advantages-start-wrapper">
          <p class="lead advantages-lead">Волновой алгоритм или алгоритм ли
— алгоритм поиска кратчайшего пути.</p>
          <p class="desc">Используется в навигации, телекоммуникационных
сетях, автоматизации конструкторского проектирования и разработке
ПО.</p>
        </div>
        <div>
        <p>Алгоритм работает на дискретном рабочем поле (ДРП),
представляющем собой ограниченную замкнутой линией фигуру, не
обязательно прямоугольную,
        разбитую на прямоугольные ячейки, в частном случае — квадратные.
Множество всех ячеек ДРП разбивается на подмножества: «проходимые»
(свободные),
```

т. е при поиске пути их можно проходить, «непроходимые» (препятствия), путь через эту ячейку запрещён, стартовая ячейка

(источник, желтый цвет на картинке) и финишная (приемник, зеленый цвет на картинке).

Назначение стартовой и финишной ячеек условно, достаточно — указание пары ячеек, между которыми нужно найти кратчайший путь.

Алгоритм предназначен для поиска кратчайшего пути от стартовой ячейки к конечной ячейке, если это возможно, либо, при отсутствии пути, выдать сообщение

о непроходимости.

Работа алгоритма включает в себя три этапа: инициализацию, распространение волны и восстановление пути.

<img src = "images/catalog/1img.png">

Во время инициализации строится образ множества ячеек обрабатываемого поля, каждой ячейке приписываются атрибуты проходимости/непроходимости,

запоминаются стартовая и финишная ячейки.

Далее, от стартовой ячейки порождается шаг в соседнюю ячейку, при этом проверяется, проходима ли она, и не принадлежит ли ранее меченной в пути ячейке.

Соседние ячейки принято классифицировать двояко: в смысле окрестности Мура и окрестности фон Неймана, отличающийся тем, что в окрестности

фон Неймана соседними ячейками считаются только 4 ячейки по вертикали и горизонтали, в окрестности Мура — все 8 ячеек, включая диагональные.

<img src = "images/catalog/2img.png">

При выполнении условий проходимости и непринадлежности её к ранее помеченным в пути ячейкам, в атрибут ячейки записывается число, равное

количеству шагов от стартовой ячейки, на первом шаге это будет 1. Каждая ячейка, меченная числом шагов от стартовой ячейки, становится

стартовой и из неё порождаются очередные шаги в соседние ячейки. Очевидно, что при таком переборе будет найден путь от начальной ячейки

к конечной, либо очередной шаг из любой порождённой в пути ячейки будет невозможен.

Восстановление кратчайшего пути происходит в обратном направлении: при выборе ячейки от финишной ячейки к стартовой на каждом шаге

выбирается ячейка, имеющая атрибут расстояния от стартовой на единицу меньше текущей ячейки. Очевидно, что таким образом находится

кратчайший путь между парой заданных ячеек.

<img src = "images/catalog/3img.png">

Трасс с минимальной числовой длиной пути, как при поиске пути в окрестностях Мура,

так и фон Неймана может существовать несколько. Выбор окончательного пути в приложениях диктуется другими соображениями, находящимися

вне этого алгоритма. Например, при трассировке печатных плат — минимумом линейной длины проложенного проводника.

</main>

<footer data-test="footer" class="page-footer">

<ul class="socials-list">

<li class="socials-item">

Вконтакте.

<svg aria-hidden="true" focusable="false" xmlns="http://www.w3.org/2000/svg" width="25" height="15" viewBox="0 0 25 15" fill="none"><path fill="#83B3D3" fill-rule="evenodd" d="M24.12 1.8c.16-.54 0-.94-.8-.94H20.7c-.67 0-.98.34-1.15.73 0 0-1.33 3.2-3.22 5.27-.61.6-.9.8-1.23.8-.16 0-.41-.2-.41-.75v-5.1c0-.66-.19-.95-.74-.95H9.82c-.42 0-.67.3-.67.59 0 .62.95.76 1.04 2.51v3.8c0 .83-.15.98-.48.98-.9 0-3.06-3.2-4.34-6.88-.25-.72-.5-1-1.18-1H1.57c-.75 0-.9.34-.9.73 0 .68.89 4.07 4.14 8.55 2.17 3.06 5.23 4.72 8 4.72 1.68 0 1.88-.37 1.88-1v-2.32c0-.74.16-.88.7-.88.38 0 1.05.19 2.6 1.66 1.79 1.75 2.08 2.54 3.08 2.54h2.63c.75 0 1.12-.37.9-1.1-.23-.72-1.08-1.77-2.2-3.02-.62-.71-1.54-1.48-1.82-1.86-.39-.5-.28-.71 0-1.15 0 0 3.2-4.42 3.54-5.93Z" clip-rule="evenodd"/></svg>

<li class="socials-item">

<a class="socials-link socials-link-tg" href="https://t.me/htmlacademy">

Telegram.

```
            <svg aria-hidden="true" focusable="false"
        xmlns="http://www.w3.org/2000/svg" width="18" height="17" viewBox="0 0
        18 17" fill="none"><path fill="#83B3D3" d="M16.79.96.84 7.1c-1.09.44-1.08
        1.05-.2 1.32l4.1 1.28 9.47-5.98c.44-.27.85-.13.52.17l-7.68 6.93-.28 4.22c.41 0
        .6-.2.83-.41l1.99-1.94 4.13 3.06c.77.42 1.31.2 1.5-.71l2.72-12.8c.28-1.1-.43-
        1.61-1.16-1.28Z"/></svg>
              </a>
            </li>
          </ul>
        </footer>



          </div>
        </div>
        <script src="scripts/script.js"></script>
      </body>
</html>
```