

**Министерство науки и высшего образования Российской Федерации**  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
**АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**  
**(ФГБОУ ВО «АмГУ»)**

Институт компьютерных и инженерных наук  
Кафедра информационных и управляющих систем  
Направление подготовки 09.03.01 – Информатика и вычислительная техника

ДОПУСТИТЬ К ЗАЩИТЕ

Зав. кафедрой

\_\_\_\_\_ А.В. Бушманов

« \_\_\_\_\_ » \_\_\_\_\_ 2024 г.

**БАКАЛАВРСКАЯ РАБОТА**

на тему: Разработка модулей хранения и интерфейса для образовательного web-сервиса «ActiveTemplate»

Исполнитель \_\_\_\_\_ Н.Е. Марков  
студент группы 0103-об (подпись, дата)

Руководитель \_\_\_\_\_ Т.А. Галаган  
доцент, канд. техн. наук (подпись, дата)

Консультант:  
по безопасности и экологичности  
доцент, канд. техн. наук \_\_\_\_\_ А.Б. Булгаков  
(подпись, дата)

Нормоконтроль \_\_\_\_\_ В.Н. Адаменко  
инженер кафедры (подпись, дата)

**Министерство науки и высшего образования Российской Федерации**  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
**АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**  
(ФГБОУ ВО «АмГУ»)

Институт компьютерных и инженерных наук

Кафедра информационных и управляющих систем

УТВЕРЖДАЮ

Зав. кафедрой

\_\_\_\_\_ А.В. Бушманов

« \_\_\_\_\_ » \_\_\_\_\_ 2024 г.

**ЗАДАНИЕ**

К выпускной квалификационной работе студента \_\_\_\_\_ Маркова Н.Е.

1. Тема выпускной квалификационной работы: Разработка модулей хранения и интерфейса образовательного web-сервиса «ActiveTemplate» (утверждено приказом от 03.04.2024 № 890-уч)

2. Срок сдачи студентом законченной работы (проекта): 17.06.2024

3. Содержание выпускной квалификационной работы: анализ предметной области; проектирование образовательного сервиса, разработка алгоритма решения; реализация проектов его модулей хранения и интерфейса.

4. Перечень материалов приложения: свидетельство о регистрации программы для ЭВМ.

5. Дата выдачи задания: 02.10.2023

Руководитель выпускной квалификационной работы: \_\_\_\_\_

Галаган Т.А. доцент кафедры ИиУС, канд. техн. наук, доцент

(фамилия, имя, отчество, должность, уч.степень, уч.звание)

Задание принял к исполнению (02.10.2023) \_\_\_\_\_

(подпись студента)

## РЕФЕРАТ

Выпускная квалификационная работа содержит 97 с., 79 рисунков, 4 таблицы, 2 приложения, 23 источника.

### ПРОЕКТИРОВАНИЕ, РАЗРАБОТКА, ВЕБ-СЕРВИС, ВЕБ-ПРИЛОЖЕНИЕ, ASP.NET, ПАТТЕРН MVC

Объектом исследования данной выпускной квалификационной работы является высшее учебное заведение АмГУ.

Предметом исследования является образовательные процессы, проводимые высшим учебным заведением Амурским государственным университетом.

Цель работы: создание функционально-полноценной системы, способной хранить созданные преподавателями задания, а также предоставлять удобный и интуитивно понятный интерфейс для взаимодействия с ними преподавателям и обучающимся.

Задачи работы: анализ существующих образовательных веб-сервисов, изучение особенностей разработки интерфейсов веб-приложений, проектирование функционального дизайна интерфейса, разработка основных функций веб-сервиса (регистрация аккаунта с подтверждением электронной почты, личный кабинет, создание и сохранение заданий, сохранение результатов прохождения заданий, встраивание заданий во внешние сайты), тестирование работы веб-сервиса.

Результатом выполнения работы является веб-приложение, предоставляющее пользователям возможность создания и прохождения заданий на основе созданных шаблонов.

## СОДЕРЖАНИЕ

Введение	7
1 Анализ функционала и особенности использования образовательных веб-сервисов	8
1.1 Обоснование актуальности темы ВКР	8
1.2 Общая характеристика веб-сервисов	9
1.2.1 Описание предметной области	9
1.2.2 Веб-сервис как объект исследования	10
1.2.3 Важность графического интерфейса в веб-приложении	12
1.3 Особенности использования веб-сервисов	12
1.3.1 Составляющие веб-сервисов решения рассматриваемой задачи	12
1.3.2 Подробное рассмотрение веб-сервисов по решению задачи	13
1.4 Задача исследования и общая методика её решения	18
2 Проектирование образовательного веб-сервиса	20
2.1 Описание функциональности образовательного веб-сервиса	20
2.2 Обзор выбранного программного обеспечения	22
2.3 Проектирование структуры базы данных	25
2.3.1 Инфологическое проектирование базы данных	25
2.3.2 Логическое проектирование базы данных	28
2.3.3 Физическое проектирование базы данных	32
2.4 Проектирование модулей для взаимодействия с базой данных	36
2.5 Проектирование пользовательского графического интерфейса	37
3 Разработка модулей образовательного веб-сервиса	43
3.1 Разработка модулей хранения данных	43
3.1.1 Модуль хранения данных пользователя	43
3.1.2 Модуль хранения данных шаблонов	47
3.1.3 Модуль хранения данных заданий	50
3.1.4 Модуль хранения данных результатов	53

3.2 Разработка интерфейса взаимодействия модулей	55
3.3 Тестирование работоспособности веб-сервиса	60
4 Безопасность и экологичность	73
4.1 Безопасность	73
4.2 Экологичность	80
4.3 Чрезвычайные ситуации	82
Заключение	85
Библиографический список	86
Приложение А Техническое задание на разработку	88
Приложение Б	97

## ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

БД – база данных.

Бэкенд – это логика работы сайта, скрытая от пользователя.

ВКР – выпускная квалификационная работа.

ВО – высшее образование.

ПИ – пользовательский интерфейс.

ПО – программное обеспечение.

Проектирование – процесс создания прототипа предполагаемого или возможного объекта.

Расширяемость – это показатель способности расширять систему и уровня усилий, необходимых для реализации расширения.

Разработка – это создание разных продуктов, для которых нужен код.

СУБД – система управления базами данных.

ФГБОУ ВО – федеральное государственное бюджетное образовательное учреждение высшего образования.

Фронтенд – это презентационная часть web-приложений, информационной или программной системы, её пользовательский интерфейс и связанные с ним компоненты.

Функциональное тестирование – проверка правильности работы функций и возможностей веб-приложения.

CSS – Cascading Style Sheet – каскадные таблицы стилей.

HTML – HyperText Markup Language – язык гипертекстовой разметки.

JS – JavaScript.

MSSQL – Microsoft SQL Server.

SSMS – SQL Server Management Studio.

## ВВЕДЕНИЕ

С развитием цифровых технологий все больше учебных материалов и курсов представлены в онлайн-формате, и их доступность и удобство использования напрямую зависят от качества интерфейса. Оптимизированный и интуитивно понятный интерфейс способствует эффективному обучению, улучшает взаимодействие студентов с учебным материалом и повышает общую эффективность образовательного процесса.

Чтобы человеку было комфортно взаимодействовать с системой, необходимо определить правильный подход к построению пользовательского интерфейса. Если рабочая область будет слишком нагружена, то пользователь заскучает, что может привести к его отказу от системы. Поэтому очень важно, чтобы графическая часть проекта радовала и не нагружала глаз.

Дизайн пользовательского интерфейса играет решающую роль в программном обеспечении. Ключевой фактор обеспечения качества программного обеспечения иметь возможность полностью понять систему для использования. Дизайн пользовательского интерфейса – это единственное лицо программного обеспечения, которое видно пользователю, следовательно, если пользователь не сможет понять и полностью использовать его, то не сможет использовать разработанное программное обеспечение. В дополнение к этому, пользовательский интерфейс должен четко показывать основные функции и функциональные возможности системы.

Разработка удобного и интуитивно понятного интерфейса для образовательного web-сервиса позволит упростить процесс обучения и повысить эффективность усвоения знаний студентами. Хорошо спроектированный интерфейс поможет пользователям не только быстро находить необходимую информацию, но и взаимодействовать с различными функциями сервиса и повысить мотивацию к обучению. Качественный интерфейс привлекает новых пользователей и способствует росту популярности образовательного web-сервиса.

# 1 АНАЛИЗ ФУНКЦИОНАЛА И ОСОБЕННОСТИ ИСПОЛЬЗОВАНИЯ ОБРАЗОВАТЕЛЬНЫХ ВЕБ-СЕРВИСОВ

## 1.1 Обоснование актуальности темы ВКР

Веб-сервис – это приложение, расположенное в веб-пространстве, с помощью которого осуществляются взаимодействия между пользователем и организацией. Оно представляет собой сайт, который обычно расположен в общем доступе для максимального охвата пользователей. Функциональность веб-сервиса напрямую зависит от организации, которая создала этот сервис.

Так вышло, что человек современного общества ожидает от каждого предприятия доступ к его услугам не только в физическом мире, но и в мире электронном. Другими словами, пользователю зачастую удобнее просмотреть информацию или получить услугу через сеть Интернет, чем тратить силы и время на транспорт до определенного предприятия. Таким образом, ведущие сотрудники принимают решение создать свой веб-сервис с заданными функциями с целью получить большую выгоду.

Основное отличие веб-сервиса от веб-сайта – это интерактивность первого. Веб-сайт в основном создаётся для размещения информации об услугах, товарах и прочем, тогда как веб-сервис делает не только всё перечисленное, но и включает в себя выполнение различных услуг прямоком на сайте. Веб-сервисы уже давно стали обыденностью и привычной вещью для каждого, ведь ежедневно каждый человек, который имеет доступ к сети Интернет, пользуется тем, или иным веб-сервисом.

Одним из самых главных занятий человека можно выделить получение образование или какого-либо навыка. В современном мире человеку без образования придётся нелегко, поэтому обучение является очень важным процессом саморазвития.

В век информационных технологий существует большое количество веб-сервисов, которые помогают в получении тех, или иных знаний. Всё, что

нужно от человека это свободное время и концентрация, остальное может предоставить сама служба.

Грамотно построенные веб-сервисы позволят человеку получить знания, которые ему необходимы, в наиболее комфортной обстановке с наилучшими условиями. Но даже если пользователю не понравится используемый им веб-сервис, он всегда в праве отказаться от него и найти новый.

С каждым днём на просторах сети Интернет появляется всё большее число веб-сервисов, направленных исключительно на образование и саморазвитие. Некоторые из них предоставляют высокое качество оказания услуг и продуманы до мелочей, но остальные сделаны будто по шаблону, что резко снижает их качество. Вспоминая веб-сервисы от корпораций, можно отметить, что каждый из них выполнен в таком «корпоративном» стиле, что отнимает у данного продукта его индивидуальность и естественность.

Веб-сервисы играют не последнюю роль в поддержании контакта между пользователем и организацией. Одна и та же работа может быть начата в физическом мире и продолжена в мире электронном. Значимость веб-сервиса при этом как для организации, так и для пользователя увеличивается в особо крупных масштабах.

## **1.2 Общая характеристика веб-сервисов**

### **1.2.1 Описание предметной области**

Предметной областью данной работы являются образовательные процессы, реализованные в ФГБОУ ВО «Амурский государственный университет».

Процесс обучения студентов на каждом направлении схож с процессом обучения любого другого университета. Студенты должны овладевать знаниями, выполнять в установленные сроки все виды заданий, предусмотренных учебным планом и образовательными программами ВО, соблюдать устав вуза, правила внутреннего распорядка и правила общежития.

Процесс овладения знаниями может выполняться разными способами, но в основном студент посещает и физически присутствует на лекционных, практических и лабораторных занятиях, где он сам контролирует свою успеваемость. Практически всегда обучающемуся выдают домашнее задание, которое он обязан своевременно выполнить, однако зачастую форма представления задания, которое нужно выполнить дома, оставляет желать лучшего, в следствие чего студент не выполняет задание.

Преподаватель в таком случае имеет минимальный набор инструментов, с помощью которых он может вести учёт выполнения домашних заданий. Данные инструменты в большинстве случаев выполняют свою единственную задачу и дают простейшую сводку результатов. Как преподаватели, так и студенты нередко теряют интерес к использованию данного вида обучения, что является серьезным упущением.

Чтобы мотивировать студента выполнять задания дома, нужно иметь грамотно исполненный инструмент, обладающий дружественным интерфейсом, высоким быстродействием и интересным набором заданий, которые будут интересно выполнять. Инструмент с удобным и понятным интерфейсом также будет очень полезен преподавателю.

В качестве такого инструмента будет использоваться веб-сервис, который будет обладать возможностью создания заданий по шаблону и сохранения результатов пройденных заданий.

### 1.2.2 Веб-сервис как объект исследования

Веб-сервис – это идентифицируемая веб-адресом программная система со стандартизированными интерфейсами. По наполнению они бывают разными в зависимости от целей их создания.

Архитектура такого программного обеспечения делится на 2 части: фронтенд (то, что видит пользователь на экране) и бэкенд (запрограммированная реакция на действия пользователя). Вместе эти части образуют приложение, которое позволяет пользователю взаимодействовать с предоставляемыми

услугами. Четкое описание и реализация каждой части являются ключами к качественному сервису. Таким образом, если бэкенд выполнен некачественно, то никакое качество фронтенд не исправит ситуацию, и наоборот.

Фронтенд является презентационной частью программной системы, её пользовательский интерфейс и связанные с ним компоненты. В разработку данной части входит создание функционала сайта, который может выполняться на стороне клиента. Разработчик должен удостовериться в работе всех элементов взаимодействия, в их правильном расположении. Чтобы конечный продукт был востребованным среди пользователей необходимо иметь четкую картину интеракции пользователя с сайтом, поставить себя на место пользователя.

В бэкенд-разработку входит создание скриптов для сервера, на котором размещается веб-приложение, наполнение системы веб-ресурса, работа с серверными технологиями. От данной разработки зависит основной функционал сайта. Идеальный разработчик должен обладать такими качествами как структурный тип мышления, внимательность к мелочам, аналитический склад ума, хорошо развитое воображение и умение развиваться в плане своих технических способностей.

Наглядное представление различий между фронтенд и бэкенд разработками показано на рисунке 1.



Рисунок 1 – Различия между фронтенд и бэкенд разработками

Функциональность веб-приложения и его применение целиком ограничивается только рамками мышления разработчика, но от того, как реализован веб-сервис и что он может предложить, во многом зависит успех проекта.

### 1.2.3 Важность графического интерфейса в веб-приложении

По своей сути, веб-сервис – это реализация интерфейсов обмена данными между различными приложениями, которые написаны не только на разных языках, но и распределены на разных узлах сети.

Пользовательский интерфейс имеет ключевое значение для обеспечения удобства и эффективности работы пользователя с системой, ведь если интерфейс веб-сайта неудобен или сложен в использовании, то пользователи потеряют интерес и не продолжат им пользоваться.

Интерфейс любого программного обеспечения должен быть лаконичным, визуально приятным и интуитивно понятным. Разработчик при создании графического интерфейса должен всегда ставить себя на место пользователя и задавать себе наводящие вопросы, которые бы помогли определить вектор развития.

Каждый интерфейс должен обладать следующими особенностями:

- интуитивность (пользователь приложения должен в короткие сроки и самостоятельно понимать, как взаимодействовать с приложением);
- эффективность (интерфейс должен обеспечивать эффективное выполнение всей совокупности функций системы);
- согласованность (элементы графического интерфейса должны быть типовыми и содержать единый дизайн приложения);
- отзывчивость (должно обеспечиваться быстрое реагирование сайта на действия пользователя).

## 1.3 Особенности использования веб-сервисов

### 1.3.1 Составляющие веб-сервисов решения рассматриваемой задачи

В сети Интернет можно найти большое число веб-сервисов, которые позволяют выполнить ту, или иную задачу. Большинство данных приложений

бесплатны или предлагают пробный период для ознакомления с работой функций. То же самое и с сервисами, которые предлагают создание интерактивных заданий.

При ознакомлении с веб-сервисом пользователь в первую очередь смотрит на возможности, которые тот предоставляет, и, в случае если пользователя устраивает набор встроенных функций, рассматривает приложение глубже. Далее пользователя интересует стоимость предоставляемых услуг и если цена за услугу выше ожидаемой, то пользователь теряет интерес к приложению. Это является проблемой большинства приложений, расположенных в сети Интернет. Для решения данной проблемы был придуман пробный период, который предоставляет доступ к полному набору функций бесплатно, но на короткий срок.

Пользователю должно быть удобно использовать приложение, то есть чтобы графический дизайн не вредил глазам, расположение функций было интуитивным и сохранение выполненной работы постоянным. Если хотя бы один из данных критериев не соблюдается, то пользователь почувствует дискомфорт при работе с приложением.

Основным набором функций, которые должно быть в веб-сервисе для решения выполняемой задачи, должен быть возможность создания, распространения и прохождения заданий с последующим сохранением результата прохождения.

### 1.3.2 Подробное рассмотрение веб-сервисов по решению задачи

Существует несколько веб-приложений, которые в точности позволяют выполнить все вышеперечисленные функции.

«LearningApps» – это конструктор интерактивных заданий, который позволяет закрепить полученные знания в игровой форме, а также поддерживает процессы обучения и преподавания с помощью небольших интерактивных мультимедийных упражнений. Домашняя страница сервиса представлена на рисунке 2.

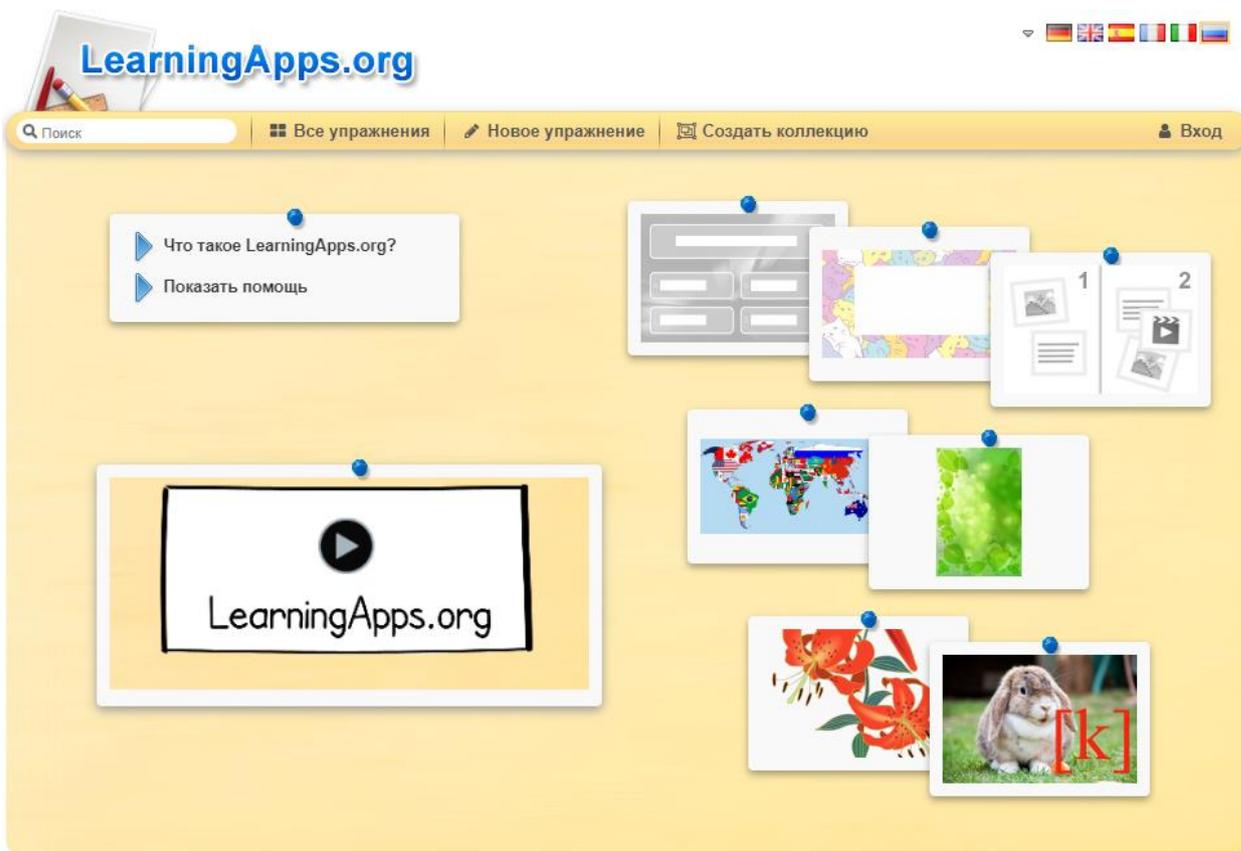


Рисунок 2 – Домашняя страница «LearningApps.org»

Данный сервис обладает возможностью создавать задания на основе заранее спроектированных шаблонов, которые заполняются пользователем, после чего распространяются посредством ссылки на сайт веб-сервиса. Сами упражнения не представляют собой самостоятельные учебные задания. Они предназначены только для отработки и углубления навыков.

Графический дизайн веб-сервиса выглядит устаревшим и простым, хоть и выполняет все, что нужно пользователю. Однако, главная проблема, которая возникает у пользователя с сайтом, это невозможность сохранения результатов пройденных тестов. Это значит, что отследить статус прохождения обучающимся теста не представляется возможным, что вызывает дополнительные проблемы у преподавателя.

Главным достоинством данного веб-сервиса является его стоимость. «LearningApps» доступен абсолютно бесплатно и содержится за счет вноса пожертвований пользователей.

Таким образом, из-за отсутствия сохранения прогресса внутри приложения сервис «LearningApps» не подходит для преподавателя.

«WordWall» – это многофункциональный инструмент для создания как интерактивных, так и печатных материалов. Большинство шаблонов доступны как в интерактивной, так и в печатной версии. При помощи этого сервиса можно создавать интерактивные упражнения и мини-игры, такие как викторины, словесные игры, сопоставления. Домашняя страница «WordWall» расположена на рисунке 3.

The screenshot displays the WordWall.net homepage. At the top, it features the headline "The easy way to create your own teaching resources." and a sub-headline "Make custom activities for your classroom." Below this, it lists "Quizzes, match ups, word games, and much more." and a "Sign Up To Start Creating" button. A small illustration shows a teacher at a desk with a screen, labeled "Teacher", "Printables", and "Interactives". A counter indicates "72.334.827 resources created".

The middle section is titled "Easy as 1-2-3" and describes the process in three steps: 1. Pick a template, 2. Enter your content, and 3. Print out your activities or play on a screen.

The bottom section is titled "Find out about our templates" and lists 18 different activity types with brief descriptions:

- Match up: Drag and drop each keyword next to its definition.
- Flash cards: Test yourself using cards with prompts on the front and answers on the back.
- Complete the sentence: A short activity where you drag and drop words into blank spaces within a text.
- Unjumble: Drag and drop words to rearrange each sentence into its correct order.
- Wordsearch: Words are hidden in a letter grid. Find them all. Fast as you can.
- Crossword: Use the clues to solve the crossword. Tap on a word and type in the answer.
- Quiz: A series of multiple choice questions. Tap the correct answer to proceed.
- Spin the wheel: Spin the wheel to see which item comes up next.
- Anagram: Drag the letters into their correct position to unscramble the word or phrase.
- Labelled diagram: Drag and drop the pins to their correct place on the image.
- Gameshow quiz: A multiple choice quiz with time pressure, lifelines and a bonus round.
- Speaking cards: Deal out cards at random from a shuffled deck.
- Group sort: Drag and drop each item into its correct group.
- Find the match: Tap the matching answer to eliminate it. Repeat until all answers are gone.
- Open the box: Tap each box in turn to open them up and reveal the same inside.
- Hangman: Try to complete the word by picking the correct letters.
- Airplane: Use touch or keyboard to fly into the correct answers and avoid the wrong ones.

A "Sign Up To Start Creating" button is located at the bottom of the page.

Рисунок 3 – Домашняя страница «WordWall.net»

Как и прошлый веб-сервис, «WordWall» предлагает создание заданий на основе шаблонов. Графический дизайн и звуковое оформление сайта выполнены в приятном и современном стиле.

В отличие от «LearningApps», за использование «WordWall» пользователю придется платить немалую сумму. Очевидно, что преподаватель не станет платить большую сумму денег за веб-сервис, даже не смотря на его высокое качество.

Еще одним отличием от «LearningApps» является то, что в данном веб-сервисе есть возможность точно отслеживать и просматривать результаты выполненных обучающимися заданий, однако, учитывая стоимость, приложение это не спасает.

В итоге, хотя «WordWall» и обладает всеми функциями, которые нужны преподавателю для создания и отслеживания заданий, из-за высокой стоимости актуальность данного веб-сервиса резко падает.

«Quizlet» – это сервис, который позволяет с помощью карточек легко запоминать любую информацию. Сервис помогает создавать флэш-карточки с картинкой или без для изучения или закрепления лексики. Такой формат заданий позволяет тренировать такие навыки, как чтение, говорение, аудирование и письмо. Домашняя страница сервиса расположена на рисунке 4.

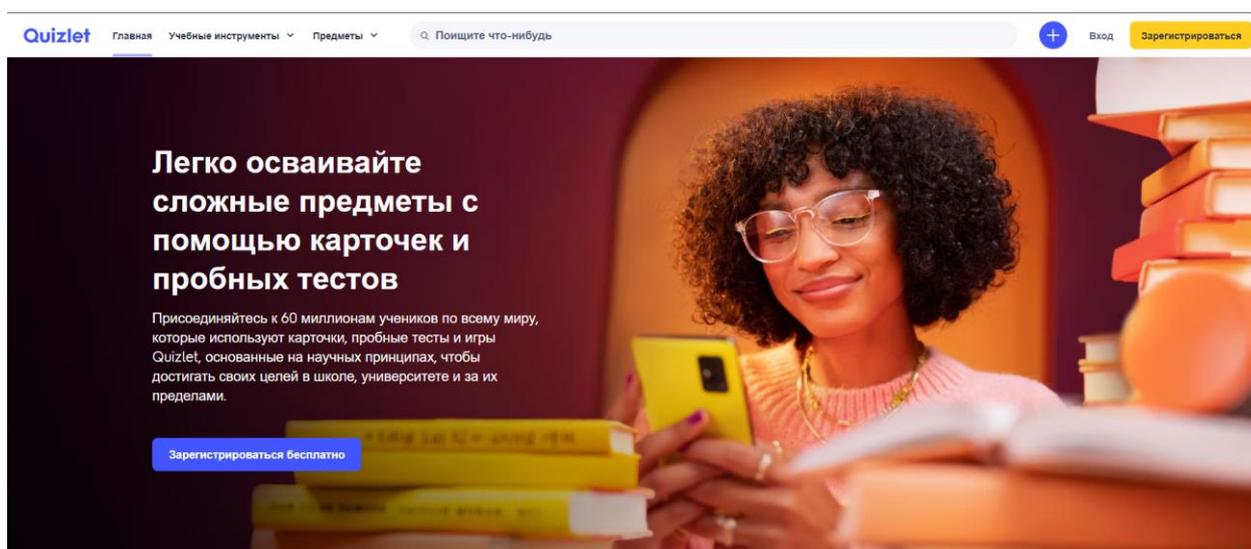


Рисунок 4 – Домашняя страница «Quizlet.com»

Данный веб-сервис считается «обучающей платформой №1» и предлагает быстрое освоение материала за счёт изучения карточек. Хотя и «Quizlet» предлагает учиться только на карточках, популярность к этому сервису с каждым днём только растёт.

Помимо веб-приложения «Quizlet» имеет мобильное приложение, которое позволяет заучивать материал не только дома, но и в любом другом месте, используя свой смартфон.

Данный веб-сервис предлагает все функции, которые быгодились преподавателю, однако отсутствие разнообразия шаблонов может повлечь за собой скуку и потерю интереса обучающегося. Ко всему этому также добавляется немалая стоимость приложения.

Таким образом, веб-сервис «Quizlet» не способен удовлетворить потребности преподавателя в создании интерактивных заданий.

Результатом анализа сильных и слабых сторон данных веб-сервисов является таблица 1.

Таблица 1 – Сравнение аналогов веб-сервисов по показателям

Веб-приложение	Графический дизайн	Сохранение и контроль прогресса заданий	Стоимость
LearningApps.org	Устаревший и простой	Отсутствует	Бесплатно
WordWall.net	Приятный и современный стиль	Присутствует	5£ / месяц за простой вариант подписки
Quizlet.com	Задания только в виде карточек, ограниченность создания заданий	Присутствует	530Р / месяц

В результате, можно сделать вывод, что каждое из этих приложений в чем-то лучше других, но универсального решения для поставленной задачи – нет.

#### **1.4 Задача исследования и общая методика её решения**

Разработку образовательного интерактивного веб-сервиса можно разбить на несколько крупных функциональных частей. В данной работе можно выделить следующие основные аспекты:

- разработка модулей хранения и интерфейса;
- разработка шаблонов интерактивных заданий;

Целью данного исследования является разработка модулей хранения и интерфейса образовательного веб-сервиса «ActiveTemplate».

Для достижения поставленной цели составлен план, по которому была проведена работа:

- анализ существующих образовательных веб-сервисов;
- изучение особенностей разработки интерфейсов веб-приложений;
- проектирование функционального дизайна интерфейса;
- разработка основных функций веб-сервиса (регистрация аккаунта с подтверждением электронной почты, личный кабинет, создание и сохранение заданий, сохранение результатов прохождения заданий, встраивание заданий во внешние сайты, поиск заданий по критериям);
- тестирование работы веб-сервиса.

Первый этап предполагает рассмотрение и оценочный анализ веб-сервисов аналогов, выполняющих ту же функцию, что и «ActiveTemplate».

На втором этапе рассматриваются все функциональные особенности веб-сервиса, которые необходимы для выполнения всей совокупности действий. Такими функциональными особенностями являются: окно авторизации, личный кабинет и профиль, отображение заданий, доступных для выпол-

нения, отображение результатов и их сохранение в профиле, отображение прочей важной информации. Между данными функциями определены взаимосвязи.

Третий этап включает в себя определение алгоритмов поведения. Определяется алгоритм действий веб-сайта при нажатии того, или иного элемента взаимодействия. На данном этапе программист должен поставить себя на место пользователя с целью выявления закономерностей работы функций.

Затем разрабатывается функциональный дизайн веб-сайта. Дизайн должен быть выполнен органичным, приятным глазу и доступным для любого пользователя. На этом этапе необходимо понимать, что расположение элементов взаимодействия играет очень важную роль при работе с веб-приложением. Если элемент расположен в не интуитивном месте, то это может вызвать негативные эмоции у пользователя.

Четвёртый этап состоит из проектирования и разработки отдельных модулей хранения данных. Данный этап также включает в себя проектирование базы данных, которое состоит из инфологического, логического и физического проектирования. Затем полученная база данных интегрируется с разработанными модулями хранения данных.

Последний этап включает в себя тестирование работоспособности веб-сервиса. Проверяется корректность работы модулей хранения данных и их взаимодействие между собой. Найденные ошибки проектирования и разработки исправляются.

Вся работа была проведена в соответствии с техническим заданием (приложение А).

## 2 ПРОЕКТИРОВАНИЕ ОБРАЗОВАТЕЛЬНОГО ВЕБ-СЕРВИСА

### 2.1 Описание функциональности образовательного веб-сервиса

В качестве основных функций веб-сервиса выделены следующие:

- аккаунт и личный кабинет;
- создание заданий по существующим шаблонам;
- хранение заданий в базе данных;
- сохранение результатов прохождения заданий;
- возможность быстрого распространения созданных заданий.

Функция аккаунта и личного кабинета позволяет разграничить каждого пользователя, при этом давая ему полный контроль над определенной частью веб-сервиса. Это предоставляет возможность не только дать пользователям место, в котором они вольны наблюдать результаты их деятельности, но и облегчить написание логики поведения веб-сервиса в различных ситуациях.

Функция гибкой настройки заданий предполагает возможность веб-сервисом персонализации каждого задания, путём изменения тех, или иных его элементов. Для разных шаблонов реализован различающийся набор атрибутов персонализации.

Возможность выполнения заданий без входа в аккаунт позволяет пользователю проходить уже созданные другими пользователями задания, пока не осуществлён вход в аккаунт. Данная функция предоставляет возможность любому человеку проверить функциональность сайта, перед тем как осуществлять на нём регистрацию.

Прямое назначение создаваемого программного продукта – создание заданий по существующим шаблонам. Используя созданные заранее программистами шаблоны, которые хранят в себе данные, отдельно прописанные специалистом, пользователь может создавать уникальные задания, вводя необходимый набор данных.



На данной диаграмме можно увидеть четырёх актёров и прецеденты, с которыми они связаны. Актёры «Преподаватель» и «Обучающийся» представляют собой разграничение пользователей на тех, кто создаёт задания и тех, кто их проходит. Однако, на данной диаграмме это представлено лишь для наглядности, в самом веб-сервисе любой пользователь может создавать и проходить задания. Действующее лицо «База данных» показывает взаимодействие базы данных веб-сервиса с людьми. Веб-сервис «ActiveTemplate» обладает свойством расширяемости, что позволяет актёру «Программист» добавлять новые шаблоны.

Любое задание должно иметь возможность быстрого и легкого распространения. Это необходимо для облегчения опыта взаимодействия пользователя с программой. Данная функция реализована в виде нескольких строк, которые представляют собой ссылку на задание и код для его встраивания во внешние ресурсы. Таким образом, доступ к создаваемым заданиям можно получить не только на официальном сайте веб-сервиса, но и на других ресурсах, владельцы которого воспользовались функцией встраивания задания.

Добавление новых шаблонов на сайт является одним из самых важных аспектов, правильность которого нужно обязательно обеспечить.

Веб-сервис обладает свойством расширяемости, это выражается благодаря шаблонам, которые можно легко и просто внести в базу данных. Единственное что для этого нужно – тело шаблона, написанное программистом отдельно.

## **2.2 Обзор выбранного программного обеспечения**

В качестве среды разработки использована Visual Studio 2022, которая представляет собой полностью интегрированную среду разработки. Она спроектирована таким образом, чтобы делать процесс написания кода, его отладки и компиляции в сборку для поставки конечным потребителям как можно более простым. На практике это означает, что Visual Studio является очень сложным

приложением с многодокументным интерфейсом, в котором можно делать практически все, что касается разработки кода.

Также, Visual Studio предоставляет множество других инструментов для работы: многофункциональный редактор форм для создания графического интерфейса приложения, мощный веб-редактор, инструмент дизайна классов и инструмент дизайна схем баз данных.

Для корректной работы данной среды программирования необходим компьютер с данными характеристиками:

- процессор с тактовой частотой 1.6 ГГц или больше;
- 1.5 Гб ОЗУ;
- 4 Гб доступного пространства на жестком диске;
- видеоадаптер с поддержкой DirectX 9.

Платформой разработки является ASP.NET, которая является отличным решением для программиста, стремящегося создать уникальное веб-приложение.

Архитектура классического ASP.NET является компонентно-ориентированной, что облегчает работу специалиста. Ближайшим аналогом данной платформы является язык программирования PHP, однако ASP.NET проще него в том смысле, что нужно меньше писать для решения одной и той же задачи. Большинство функций, которые реализованы во множестве веб-приложений, уже представлены и встроены в ядро ASP.NET.

Работа на ASP.NET позволит не только сократить время разработки веб-сервиса, но реализовать те функции, на реализацию которых не хватило бы знаний и навыков.

Когда речь заходит о разработке веб-приложения, всегда используются языки разметки и таблиц стилей, HTML и CSS, соответственно. Именно с помощью этих двух языков происходит основная «вёрстка» веб-сервиса, тем самым образуя интерфейс (фронтенд).

Язык разметки HTML является стандартизированным языком гипертекстовой разметки документов для просмотра веб-страниц в браузере, что означает его использование возможно на всех современных интернет-браузерах. В HTML есть теги, которые помогают браузеру правильно отобразить содержимое сайта. С помощью него определяется постановка текста и других элементов взаимодействия на странице. Именно язык разметки обращается ко внутренней (бэкенд) части путём посылания запроса.

Язык таблиц стилей CSS неразрывно связан с HTML и является его дополнением. Объекты, расположенные на странице, размещаются с помощью HTML. А вот CSS отвечает за то, как эти объекты выглядят. Их размер, цвет, фоновое изображение, степень прозрачности, расположение относительно других элементов, поведение при наведении курсора, визуальное изменение кнопок при нажатии и т.п. CSS предоставляет возможность, как угодно, стилизовать веб-страницу, тем самым придав ей уникальный вид.

Для реализации некоторых функций веб-сервиса используется язык программирования JavaScript, который изначально был придуман для браузера, чтобы придать страницам интерактивность и динамичность. С помощью него выполняются некоторые действия, которые было бы невозможно воплотить без использования данного языка.

В качестве системы управления реляционными базами данных используется Microsoft SQL Server, которая обладает высоким уровнем надежности и защиты данных. Данная СУБД зарекомендовала себя как удобное и надежное решение по структуризации данных. Кроме того, MS SQL Server поддерживает SQL Server Management Studio – инструмент, который позволяет в графическом интерфейсе управлять базой данных. С помощью данного программного обеспечения происходит основное взаимодействие с базой данных, которое выражается в создании таблиц, указании атрибутов, определении связей между таблицами и выполнении запросов.

Распределение программного обеспечения по фронтенд и бэкенд частям показано на рисунке 6, где ASP.NET представляет платформу, на которой происходят все взаимодействия остальных языков, HTML, CSS и JavaScript отвечают за фронтенд, а с помощью MSSQL происходит связь веб-сервиса с базой данных.

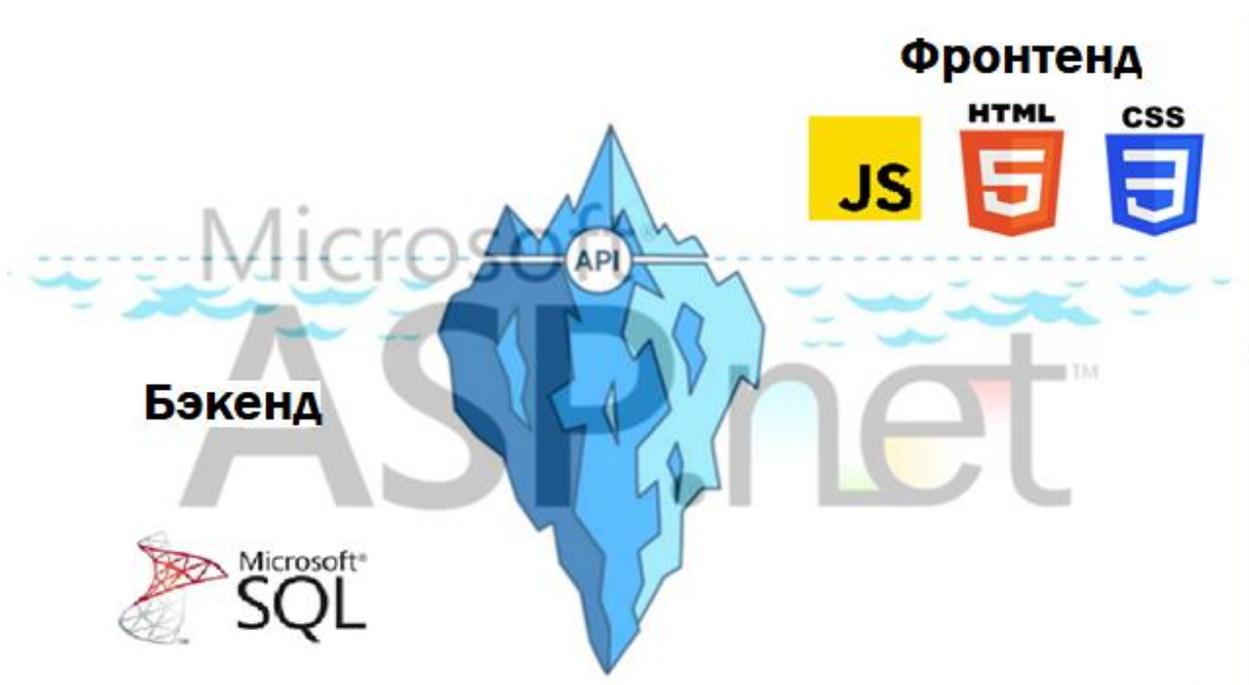


Рисунок 6 – Распределение программного обеспечения по частям

Таким образом, каждый язык, СУБД и платформа использованы по их эксплуатационному назначению. ПО, не выполняющее функций, в данном проекте отсутствует.

### **2.3 Проектирование структуры базы данных**

Проектирование структуры базы данных проходит через три стадии: инфологическое, логическое и физическое проектирования.

#### **2.3.1 Инфологическое проектирование базы данных**

Цель инфологического проектирования – обеспечение наиболее естественных для человека способов сбора и представления той информации, которую предполагается хранить в создаваемой базе данных.

В результате логического проектирования была разработана следующая модель (рис. 7), реализованная в нотации Чена:

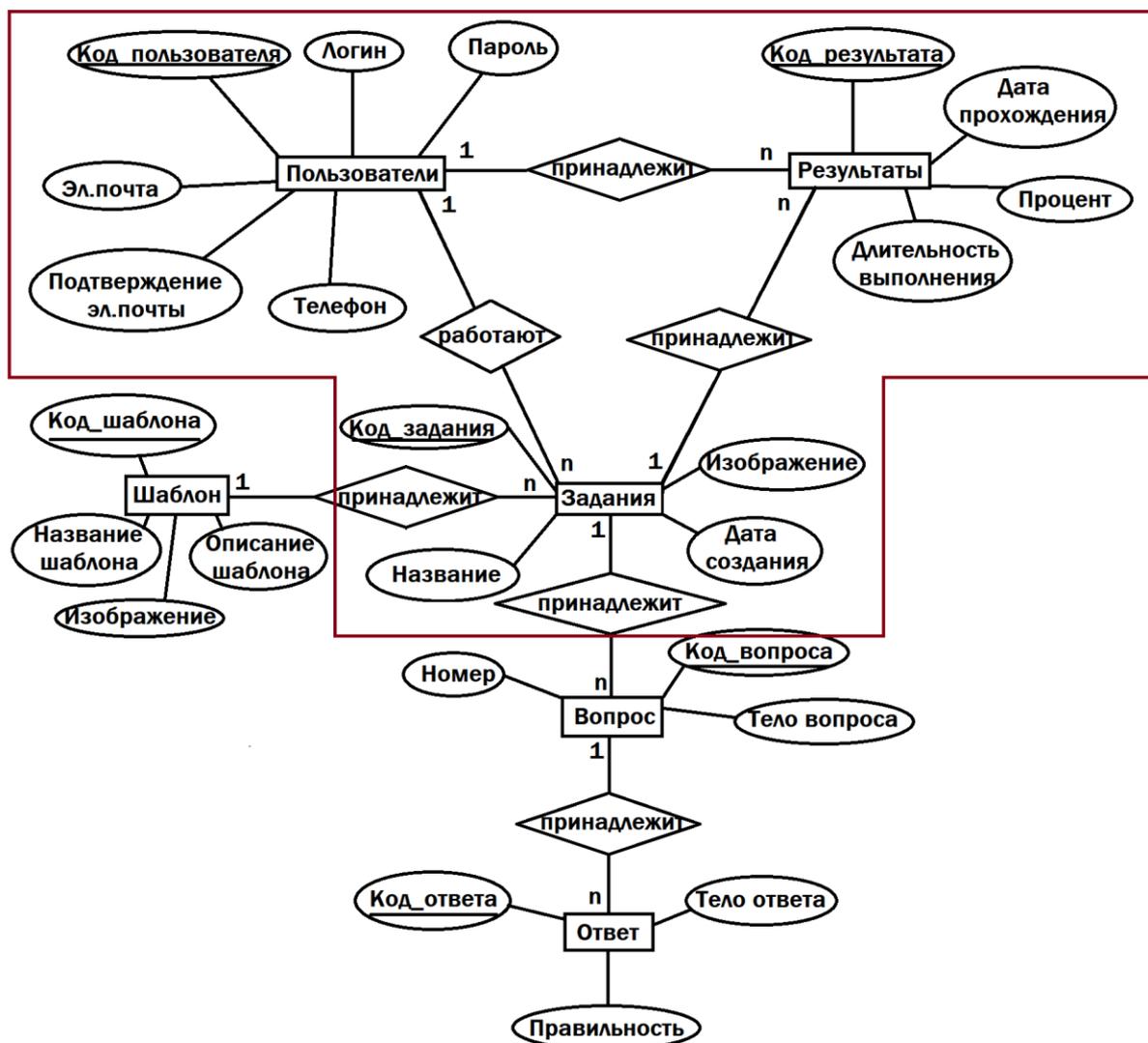


Рисунок 7 – Модель базы данных в нотации Чена

Жирной красной линией выделена часть базы данных, относящаяся к выполнению данного выпускной квалификационной работы.

Данная модель в полной мере описывает структуру базы данных создаваемого веб-сервиса. В данной нотации сущности показаны в виде прямоугольников, связи – ромбы, атрибуты – овалы.

Таким образом, веб-сервис содержит данные о четырёх сущностях:

- Пользователи содержит в себе данные об аккаунте пользователя;
- Результаты содержит в себе результаты прохождения заданий;

– Задания содержит в себе список созданных заданий.

Сущность «Пользователи» включает в себя следующие атрибуты:

– код\_пользователя: является первичным ключом, конкретно определяет каждую запись сущности «Пользователи»;

– логин: определяет имя аккаунта пользователя, может быть изменен в любой момент посредством перехода в личный кабинет;

– пароль: определяет кодовое слово для входа в аккаунт пользователя, может быть изменен в любой момент посредством перехода в личный кабинет;

– электронная почта: определяет электронный адрес, на который будут отправляться письма оповещений и подтверждения действий над аккаунтом. Может быть изменена в любой момент посредством перехода в личный кабинет;

– телефон: определяет номер телефона пользователя для дальнейших оповещений пользователя.

Сущность «Результаты» включает в себя следующие атрибуты:

– код\_результата: является первичным ключом, конкретно определяет каждую запись сущности «Результаты»;

– дата/время прохождения: определяет дату и время прохождения пользователем задания в форме, установленной на компьютере пользователя;

– процент: определяет степень прохождения задания. Для каждого задания шаблона устанавливается собственная логика определения степени прохождения;

– время прохождения: определяет время, затраченное на прохождение задания.

Сущность «Задания» включает в себя следующие атрибуты:

– код\_задания: является первичным ключом, конкретно определяет каждую запись сущности «Задания»;

– дата/время создания: определяет дату и время создания задания, в форме, установленной на компьютере пользователя;

– название: определяет название, которое видит пользователь перед открытием задания;

– изображение: определяет изображение, которое используется при отображении задания. Является необязательным атрибутом.

Организованы связи между сущностями:

– Пользователи-Результаты имеет связь один-ко-многим, то есть у одного пользователя может быть множество результатов, но у одного результата один пользователь, тот, который получил этот результат;

– Пользователи-Задания обладает связью один-ко-многим, то есть у одного пользователя может быть множество заданий, но у одного задания только один пользователь, тот, который его создал;

– Задания-Результаты имеет связь один-ко-многим, то есть у задания может быть множество результатов, но у одного результата только одно задание;

– Шаблоны-Задания обладает связью один-ко-многим, то есть от одного шаблона может быть создано множество заданий, но у каждого задания только один шаблон, тот, по которому оно было создано.

### 2.3.2 Логическое проектирование базы данных

Целью логического проектирования является создание логической модели данных для исследуемой части предприятия. Логическая модель данных учитывает особенности выбранной модели организации данных в целевой СУБД.

Этап логического проектирования включает в себя анализ каждой связи между сущностями с целью составления отношений, которые включают в себя внешние ключи. Все связи, обозначенные на предыдущем этапе, являются связями типа «один-ко-многим».

Связь Пользователи-Результаты (рис. 8): сущность «Результаты» является исходной, так как из неё исходит простая связь, в то время как сущность «Пользователи» является порождённой.



Рисунок 8 – Связь Пользователи-Результаты

В итоге, первичный ключ сущности «Пользователи» заносится в отношение «Результаты», что можно увидеть на рисунке 9.



Рисунок 9 – Результат анализа связи Пользователи-Результаты

Связь Пользователи-Задания (рис. 10): сущность «Задания» является исходной, так как из неё исходит простая связь, в то время как сущность «Пользователи» является порождённой.



Рисунок 10 – Связь Пользователи-Задания

В результате анализа, первичный ключ сущности «Пользователи» заносится в отношение «Задания», что можно увидеть на рисунке 11.

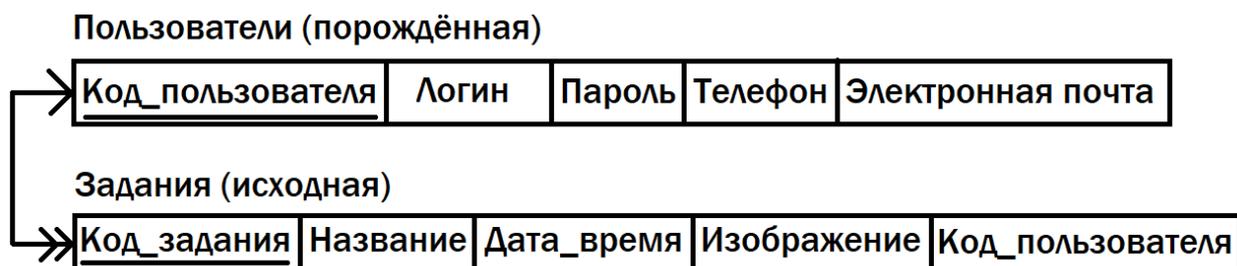


Рисунок 11 – Результат анализа связи Пользователи-Задания

Связь «Задания»-«Результаты» (рис. 12): сущность «Результаты» является исходной, так как из неё исходит простая связь, в то время как сущность «Задания» является порождённой.

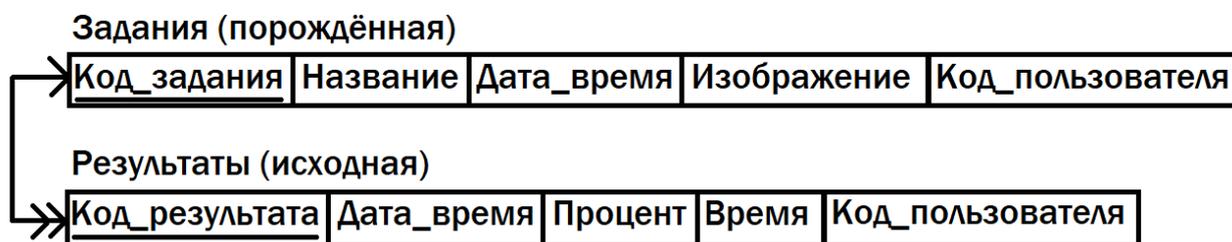


Рисунок 12 – Связь Задания-Результаты

В результате анализа, первичный ключ сущности «Задания» заносится в отношение «Результаты», что можно увидеть на рисунке 13.



Рисунок 13 – Результат анализа связи Задания-Результаты

Связь Шаблоны-Задания (рис. 14): сущность «Задания» является исходной, так как из неё исходит простая связь, в то время как сущность «Шаблоны» является порождённой.

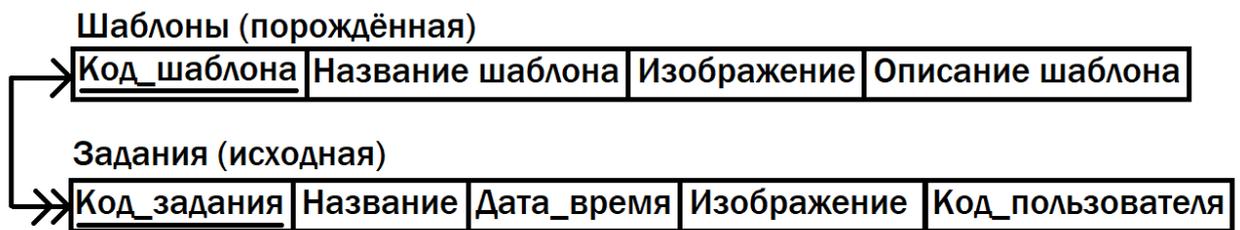


Рисунок 14 – Связь Шаблоны-Задания

Результат анализа связи Шаблоны-Задания можно увидеть на рисунке 15, где первичный ключ сущности «Шаблоны» был занесён в отношение «Задания».

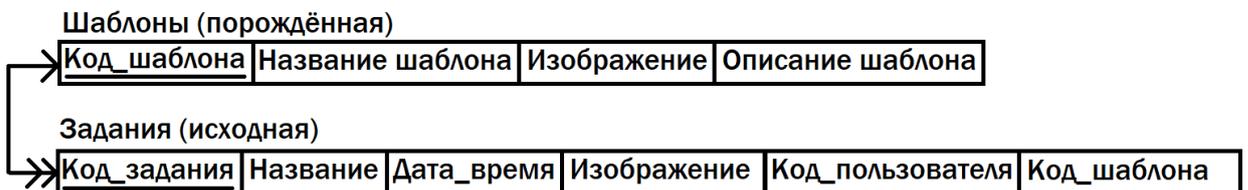


Рисунок 15 – Результат анализа связи Шаблоны-Задания

В результате анализа всех связей получается определенный набор результирующих отношений, которые можно увидеть на рисунке 16.

**Задания**

<u>Код_задания</u>	Название	Дата_время	Изображение	Код_пользователя	Код_шаблона
--------------------	----------	------------	-------------	------------------	-------------

**Результаты**

<u>Код_результата</u>	Дата_время	Процент	Время	Код_пользователя	Код_задания
-----------------------	------------	---------	-------	------------------	-------------

**Пользователи**

<u>Код_пользователя</u>	Логин	Пароль	Телефон	Электронная почта
-------------------------	-------	--------	---------	-------------------

**Шаблоны**

<u>Код_шаблона</u>	Название шаблона	Изображение	Описание шаблона
--------------------	------------------	-------------	------------------

Рисунок 16 – Результирующие отношения

Результатом логического проектирования является диаграмма базы данных, представленная на рисунке 17.

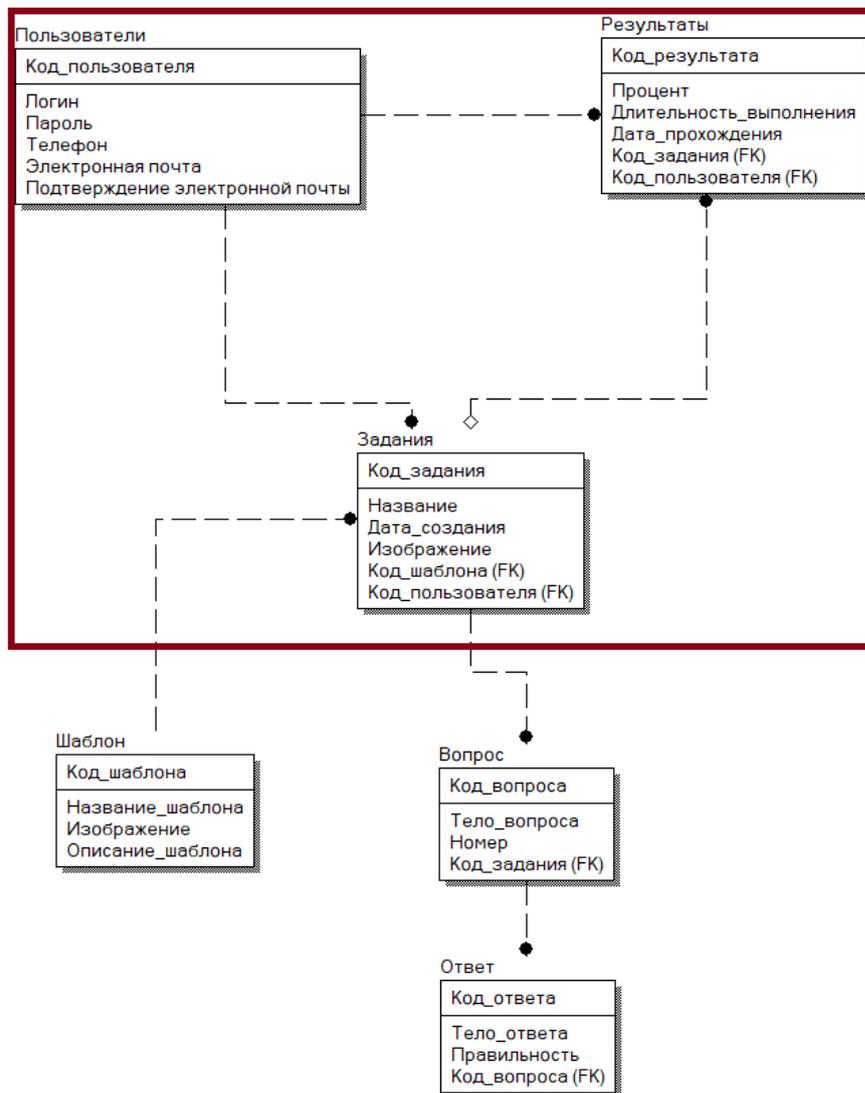


Рисунок 17 – Логическая модель базы данных

Красной жирной линией выделена часть базы данных, разработанная в рамках данной выпускной квалификационной работы.

Данная диаграмма была создана, используя инструмент Erwin Data Modeler. Были уточнены первичные и добавлены внешние ключи, образуя тем самым связи один-ко-многим.

### 2.3.3 Физическое проектирование базы данных

Цель этапа физического проектирования – описание конкретной реализации базы данных, размещаемой во внешней памяти компьютера. Это описание структуры хранения данных и эффективных методов доступа к данным базы.

Определены и нормализованы схемы каждого отношения:

– отношение «Пользователи» (таблица 2), как и все другие отношения, имеет первичный ключ, который выражается в виде идентификационного номера пользователя. Помимо этого, отношение не имеет внешних ключей, однако она всё равно неразрывно связана с другими отношениями;

Таблица 2 – Схема отношения «Пользователи»

Поле	Тип поля	Размер поля	Значение по умолчанию	Ограничения	Ключ или индекс
Код_пользователя	Int	4	IDENTITY(1,1)	Обязательное поле	Первичный ключ
Логин	varchar	50	–	Обязательное поле	–
Электронная_почта	varchar	Max	–	Обязательное поле	–
Подтверждение_эл.почты	boolean	4	–	Обязательное поле	–
Пароль	varchar	Max	–	Обязательное поле	–
Телефон	varchar	50	–	Необязательное поле	–

– отношение «Задания» (таблица 3) обладает двумя внешними ключами, «Код\_пользователя» и «Код\_шаблона», к таблицам «Пользователи» и «Шаблон» соответственно. Таким образом оформлена связь один-ко-многим, где задание связано с пользователем и шаблоном, по которому оно было создано;

Таблица 3 – Схема отношения «Задания»

Поле	Тип поля	Размер поля	Значение по умолчанию	Ограничения	Ключ или индекс
Код_задания	int	4	IDENTITY(1,1)	Обязательное поле	Первичный ключ

Название	varchar	50	–	Обязательное поле	–
Дата_время	datetime	Max	–	Обязательное поле	–
Изображение	varbinary	Max	–	–	–
Код_пользователя	int	4	–	Обязательное поле	Внешний ключ к Пользователи
Код_шаблона	int	4	–	Обязательное поле	Внешний ключ к Шаблоны

– отношение «Результаты» (таблица 4) обладает двумя внешними ключами к таблицам «Пользователи» и «Задания». Связь один-ко-многим определена следующей логикой: у одного результата может быть только один пользователь и одно задание, однако оных результатов может быть множество;

Таблица 4 – Схема отношения «Результаты»

Поле	Тип поля	Размер поля	Значение по умолчанию	Ограничения	Ключ или индекс
Код_результата	int	4	IDENTITY(1,1)	Обязательное поле	Первичный ключ
Процент	int	4	–	Обязательное поле	–
Дата_время	datetime	max	–	Обязательное поле	–
Время	int	4	–	Обязательное поле	–
Код_пользователя	int	4	–	Обязательное поле	Внешний ключ к Пользователи

Код_задания	int	4	—	Обязательное поле	Внешний ключ к Задания
-------------	-----	---	---	-------------------	------------------------

В результате физического проектирования базы данных разработана физическая модель базы данных, которая показана на рисунке 18.

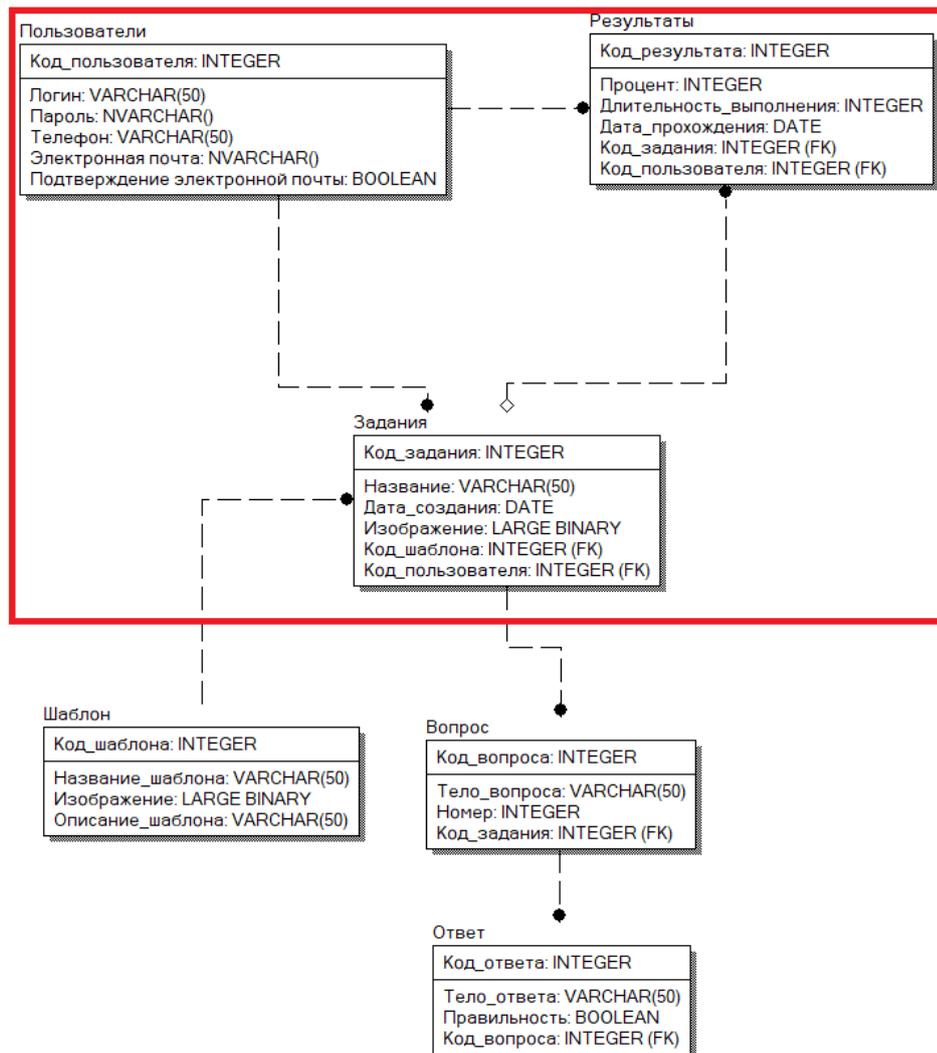


Рисунок 18 – Физическая модель базы данных

Красной жирной линией выделена часть модели БД, разработанная в данной выпускной квалификационной работе.

Таким образом, все этапы проектирования базы данных были последовательно пройдены. Цели каждой стадии достигнуты, задачи выполнены.

## 2.4 Проектирование модулей для взаимодействия с базой данных

Проектирование модулей хранения предполагает описание реализации средств взаимодействия пользователя с веб-сервисом, который, в свою очередь, взаимодействует с реализованной базой данных.

Отношение «Пользователи» создано исключительно для работы с данными пользователей, поэтому каждый атрибут сущности представляет собой ту или иную часть, отвечающую за данные аккаунта человека. Учитывая данные особенности отношения составлен перечень задач, которые необходимо выполнить для обеспечения полноценной функциональности модуля:

- создать отдельную страницу регистрации и входа в аккаунт;
- создать отдельную страницу личного кабинета, в котором можно изменить данные аккаунта;
- реализовать систему подтверждения изменения данных путём отправки письма на электронную почту пользователя;
- обеспечить пользователя доказательством входа в аккаунт в любом месте веб-сервиса.

При выполнении данных задач появляется возможность реализации особой логики на странице. Например, если пользователь не вошел в аккаунт или не активировал его, то ему заблокирована возможность создания, но доступна функция просмотра заданий.

Отношение «Задания» предполагает содержание в себе всех заданий, созданных пользователями на веб-сервисе. Данное отношение неотъемлемо связано с отношением «Пользователи», то есть задание не может существовать без пользователя, который его создал. Имея это в виду, введено ограничение на создание заданий неавторизованным пользователям. Также, выделены следующие задачи по реализации модуля хранения заданий:

- создать отдельную страницу создания заданий на основе шаблонов;
- разработать отдельную форму создания задания для каждого шаблона;
- разработать перенос данных из формы в тело шаблона;

- разработать систему по сохранению задания с введёнными данными в базу данных;

- создать отдельные страницы просмотра всех заданий и заданий, созданных пользователем.

Выполнение данных задач предполагает наличие уже разработанного модуля хранения данных о шаблонах, поэтому отношение «Шаблоны» реализовано таким образом. Для этого отношения необходимо выполнить следующие задачи:

- определить, какие данные должны быть в форме создания задания чтобы в полной мере передать суть шаблона;

- разработать систему динамического обновления доступных шаблонов (таким образом пользователь не сможет создать задание по шаблону, находящемуся в разработке или в доработке).

После создания задания по определенному шаблону пользователь неизбежно займёт желание посмотреть и пройти созданное задание. Отношение «Результаты» создано именно для отслеживания результатов прохождения заданий. Учитывая это, необходимо выполнить следующие задачи для достижения корректной работоспособности модуля:

- определить пределы прохождения реализованных шаблонов;
- создать систему по определению степени прохождения задания;
- создать отдельную страницу по просмотру результатов прохождения заданий, созданных и пройденных пользователем.

Каждый модуль хранения несёт в себе определенные функции, которые неразрывно связаны друг с другом, и в случае отказа одного из модулей пользователю может понадобится помощь.

## **2.5 Проектирование пользовательского графического интерфейса**

Пользовательский графический интерфейс определяет, каким образом пользователь будет взаимодействовать с приложением. В качестве инструмен-

тов взаимодействия могут выступать: текстовые поля, кнопки и галочки, выпадающие списки, всплывающие подсказки, переключатели, элементы меню сайта и многие другие.

Для создания качественного и эффективного интерфейса необходимо учитывать на протяжении всего цикла разработки основные принципы проектирования:

- сигнализировать о состоянии системы – после каждого совершенного действия пользователь должен знать, что происходит и получать обратную связь;

- использовать опыт пользователя и силу привычки – необходимо учитывать предыдущий опыт пользователя и привычные ему шаблоны поведения, расставляя элементы взаимодействия в положенном им месте;

- экономить усилия пользователя – программа должна запоминать информацию, которая в течение длительного времени не изменится, к примеру, логин и пароль для аутентификации;

- продумать пути выхода из сценария – неверные действия пользователя не должны приводить к критической ситуации, всегда должна быть кнопка отката действия;

- позволить пользователям настроить интерфейс для себя – качественная программа должна поддерживать возможности изменения интерфейса для улучшения качества взаимодействия и эстетического удовольствия пользователя;

- помнить о согласованности данных друг с другом – данные должны быть всегда согласованы между собой, стоит придерживаться одной концепции дизайна и не менять её в угоду красоты страницы;

- предотвращать ошибки – нужно помогать пользователю избежать ошибок во время прохождения сценария, к примеру, поставить дальше друг от друга кнопки с противоположным действием;

– дозировать информацию – слишком большой объем информации, как правило, утомляет глаз и понижает внимание пользователя, что приводит к уменьшению его заинтересованности проектом.

Соблюдение данных принципов способствует созданию качественного и легкого в обращении графического интерфейса пользователя.

Одной немало важной деталью интерфейса является его дизайн – то, от чего зависит внимание пользователя. Красиво оформленный простой веб-сервис всегда будет привлекательнее, чем скучный на дизайн сложный веб-сервис. С этой мыслью принято решение о реализации одностраничного дизайна интерфейса.

Одностраничный дизайн позволит сконцентрировать внимание пользователя на одной странице, не отвлекая его от взаимодействия с веб-сервисом. Это достигается путём приведения каждой страницы веб-приложения к похожему виду и незамысловатой анимации перехода от одной страницы к другой.

Необходимо помнить о единстве дизайна каждой страницы. Появление цвета из незнакомой палитры цветов покажется пользователю неестественным, что может изрядно попортить опыт пользования веб-сервисом. В данном веб-сервисе основным цветом выступает «dodgerblue» (#1E90FF), представляющий собой светло-синий. Цвет подобного оттенка не напрягает глаз и дополняет разнообразие веб-приложения.

Важно учитывать размер шрифта в каждом месте, где присутствует текст. Слишком большой или маленький текст мешает сканированию страницы глазом, что негативно сказывается на времени, которое пользователь тратит бездействуя. Грамотно подобранный размер и шрифт говорят об опытности специалиста, разработавшего веб-сайт.

Проектирование интерфейса заключается в проектировании каждой отдельной страницы. Необходимо учитывать и использовать опыт пользователя при составлении концепта каждой страницы. К примеру, домашняя страница является той, которая встречает пользователя, только перешедшего по ссылке

на сайт. В данном случае стоит подсказать пользователю о предлагаемых возможностях веб-сервиса и указать на то, что ему следует делать дальше для продвижения.

После перехода на домашнюю страницу пользователю рекомендуется выполнить регистрацию или войти в аккаунт. Страницы подобных действий оформлены в похожем виде, чтобы не сбивать пользователя с толку.

При оформлении страницы просмотра шаблонов для создания заданий необходимо учитывать время пользователя. Для улучшения опыта взаимодействия пользователя с веб-сервисом все шаблоны оформлены в виде карточек, которые на лицевой стороне показывают картинку шаблона, описывающую его суть, и само его название. Описание к шаблону может быть броским и занимать слишком много времени, поэтому реализована особая анимация, которая проигрывается при наведении курсора на карточку.

Тело шаблона оформляется отдельным программистом и является уникальным в своем роде, однако общая особенность каждого – указание ссылки на задание и строки кода для введения задания во внешний ресурс. Обе эти строки оформлены таким образом, чтобы они были видны в любом случае, независимо от вида заднего фона. Отдельной полезной функцией является кнопка быстрого копирования строки, значительно упрощающая взаимодействие пользователя с заданием.

Страницы просмотра заданий обладают практически идентичным видом, но отличия достаточно заметны, для различения двух страниц. Каждое задание содержит в себе внешние данные, которые включают в себя название, картинку, описание и автора, поэтому в данном веб-сервисе просмотр информации о заданиях реализован в виде строк, которую можно быстро прочитать слева-направо.

Для быстрой навигации по веб-сервису предусмотрено горизонтальное меню, находящееся в верхней части экрана. Данное меню всегда находится на

своем месте, что позволяет пользователю в любой момент переместиться на одну из предлагаемых страниц.

Для того, чтобы разделить логические части приложения и создавать их отдельно друг от друга используется паттерн MVC (Model-View-Controller).

При использовании данного паттерна создаются три отдельных компонента:

- модель (Model): описывает используемые в приложении данные, а также логику, которая связана непосредственно с данными, например, логику валидации данных. Объекты моделей хранятся в базе данных. Модель может содержать данные, хранить логику управления этими данными. Кроме того, модель не должна содержать логику отображения данных в представлении;

- представление (View): отвечают за визуальную часть или пользовательский интерфейс, часто html-страница, через которую пользователь взаимодействует с приложением. Также представление может содержать логику, связанную с отображением данных. В то же время представление не должно содержать логику обработки запроса пользователя или управления данными;

- контроллер (Controller): представляет центральный компонент MVC, который обеспечивает связь между пользователем и приложением, представлением и хранилищем данных. Он содержит логику обработки запроса пользователя. Контроллер получает вводимые пользователем данные и обрабатывает их.

Отношения между компонентами можно описать следующим образом (рис. 19):

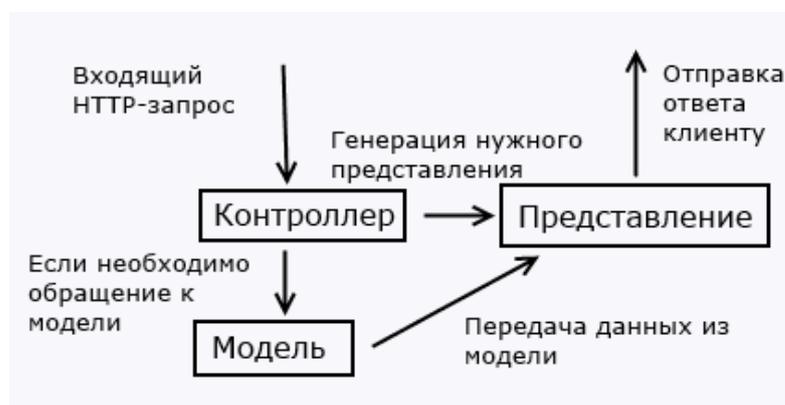


Рисунок 19 – Связи между компонентами на основе паттерна MVC

В этой схеме модель является независимым компонентом – любые изменения контроллера или представления никак не влияют на модель. Контроллер и представление являются относительно независимыми компонентами. Так, из представления можно обращаться к определенному контроллеру, а из контроллера генерировать представления, но при этом нередко их можно изменять независимо друг от друга.

Такое разграничение компонентов приложения позволяет реализовать концепцию разделение ответственности, при которой каждый компонент отвечает за свою строго очерченную сферу. В связи с чем легче построить работу над отдельными компонентами. И благодаря этому приложение легче разрабатывать, поддерживать и тестировать отдельные компоненты.

Проектирование веб-сервиса осуществляется посредством создания данных компонентов. Для каждой сущности базы данных создаётся своя модель, которая, взаимодействуя через представление с контроллером, обращается к базе данных, тем самым сохраняя или выгружая из нее данные.

Результатом проектирования образовательного веб-сервиса «ActiveTemplate» является следующее:

- определен набор программных средств, способствующих разработке программного продукта;
- определены основные функции веб-сервиса;
- установлены вектора развития веб-сервиса после окончания его разработки;
- разработана инфологическая, логическая и физическая модель базы данных в программе SQL Server Management Studio;
- определены задачи по разработке каждого модуля хранения;
- разработаны концепции каждой страницы веб-сервиса.

## 3 РАЗРАБОТКА МОДУЛЕЙ ОБРАЗОВАТЕЛЬНОГО ВЕБ-СЕРВИСА

### 3.1 Разработка модулей хранения данных

Данный этап предполагает выполнение вышеописанных задач по разработке модулей хранения данных. Каждый модуль разрабатывается последовательно в соответствии с установленными задачами.

#### 3.1.1 Модуль хранения данных пользователя

Модуль хранения данных пользователя представляет собой формы регистрации и входа, которыми пользуется человек единожды при заходе на веб-сервис.

Каждая форма выполнена с помощью тэга «form» и его вспомогательных тегов «input». Таким образом, при подтверждении формы происходит отправка данных в виде модели пользователя, которая проверяется на присутствие или отсутствие в базе данных.

Учтена одинаковая направленность обеих страниц, что выражается в практически идентичном оформлении (рис. 20):

## Регистрация

### Создайте новый аккаунт

Электронная почта
Логин
Пароль
Подтверждение пароля
<b>Зарегистрироваться</b>

### Вход

### Используйте существующий аккаунт чтобы войти

Логин
Пароль
<input type="checkbox"/> Запомнить вход?
<b>Войти</b>
<a href="#">Зарегистрировать нового пользователя</a>

Рисунок 20 – Формы регистрации и входа

С помощью данных форм пользователь создаёт, либо входит в уже созданный аккаунт. Для каждой формы описан свой набор элементов взаимодействия, который выражается в наличии электронной почты, логина, пароля и подтверждения пароля для регистрации, и в наличии логина и пароля для входа.

Для формы регистрации установлены меры, обеспечивающие целостность в базе данных. Таким образом, если пользователь попытается зарегистрировать аккаунт, логин которого уже занят, при нажатии кнопки регистрации будет выдана ошибка. То же самое происходит и при вводе уже зарегистрированной электронной почты (рис. 21):

## Регистрация

### Создайте новый аккаунт

• Никнейм "example" уже занят

Электронная почта  
example@mail.ru

Никнейм  
example

Пароль  
.....

Подтверждение пароля  
.....

**Зарегистрироваться**

## Регистрация

### Создайте новый аккаунт

• Почта "example@mail.ru" уже занята

Электронная почта  
example@mail.ru

Никнейм  
example

Пароль  
.....

Подтверждение пароля  
.....

**Зарегистрироваться**

Рисунок 21 – Сообщения об ошибках при вводе существующих данных

Также, присутствует система подсказок, которая описывает ошибки заполнения полей. Эти ошибки могут варьироваться для каждого поля.

Таким образом, если пользователь введёт электронную почту, не включающую в себя символ «@» и дополнительные символы после него, то ошибка

выдаст ошибку заполнения, так как такого адреса точно не существует. В случае с созданием пароля для аккаунта существует несколько правил, которые должны быть учтены для создания записи в базе данных. Одним из таких правил является количество символов в пароле не меньше 6 и не больше 100. Также, обязательно должны совпадать данные, введённые в поле «Пароль» и «Подтверждение пароля».

Форму ошибок можно увидеть на рисунке 22.

## Регистрация

### Создайте новый аккаунт

The image shows a registration form with four input fields and a submit button. Each field has a red error message below it. The fields and their errors are: 1. 'Электронная почта' with 'example mail.ru' and error 'Некорректный электронный адрес'. 2. 'Никнейм' with 'example'. 3. 'Пароль' with four dots and error 'Длина должна составлять не менее 6 и не более 100 символов соответственно.'. 4. 'Подтверждение пароля' with one dot and error 'Пароль и пароль для подтверждения не совпадают.'. At the bottom is a blue button labeled 'Зарегистрироваться'.

Электронная почта  
example mail.ru  
Некорректный электронный адрес

Никнейм  
example

Пароль  
••••  
Длина должна составлять не менее 6 и не более 100 символов соответственно.

Подтверждение пароля  
•  
Пароль и пароль для подтверждения не совпадают.

Зарегистрироваться

Рисунок 22 – Сообщения об ошибках заполнения полей регистрации

Нажатие кнопки регистрации создаёт запрос в базу данных на добавление новой учетной записи в таблице «Пользователи». Однако, аккаунт считается не активированным, пока пользователь не подтвердит его через письмо, отправленное на электронный адрес. После подтверждения аккаунта пользователю необходимо войти в него, используя данные, введённые при регистрации.

Форма входа не обладает особыми проверками заполнения полей, однако если пользователь введёт неверные данные, будет выведена ошибка (рис. 23):

**Вход**  
**Используйте существующий аккаунт чтобы войти**

• Неверная попытка входа

Никнейм  
example

Пароль

Запомнить вход?

**Войти**

[Зарегистрировать нового пользователя](#)

Рисунок 23 – Сообщение об ошибке неверного ввода данных

Дополнительно на данной форме предусмотрена галочка «Запомнить вход?», которая запоминает введённые данные пользователя при следующем открытии веб-приложения.

Все данные пользователя можно изменить путём перехода в личный кабинет, который представлен на рисунке 24.

### Управляйте своим аккаунтом

Поменяйте настройки своего аккаунта

**Профиль**

Электронная почта

Пароль

Персональные данные

**Профиль**

Логин  
example

Телефон

**Сохранить**

Рисунок 24 – Форма личного кабинета пользователя

На данной странице можно поменять свой логин, электронную почту, телефон и пароль. Каждое действие должно быть подтверждено путём отправки письма с подтверждением на электронную почту.

Таким образом пресекаются возможности несанкционированного изменения данных аккаунта.

### 3.1.2 Модуль хранения данных шаблонов

После входа в аккаунт пользователь может перейти к следующему модулю хранения данных, а именно к шаблонам. Данный модуль использует в качестве модели шаблоны, которые с каждым обновлением страницы загружаются из базы данных. На данной странице предоставлена возможность выбрать один из доступных шаблонов, как это показано на рисунке 25.

#### Выберите один из представленных шаблонов:



Рисунок 25 – Окно выбора шаблонов для создания задания

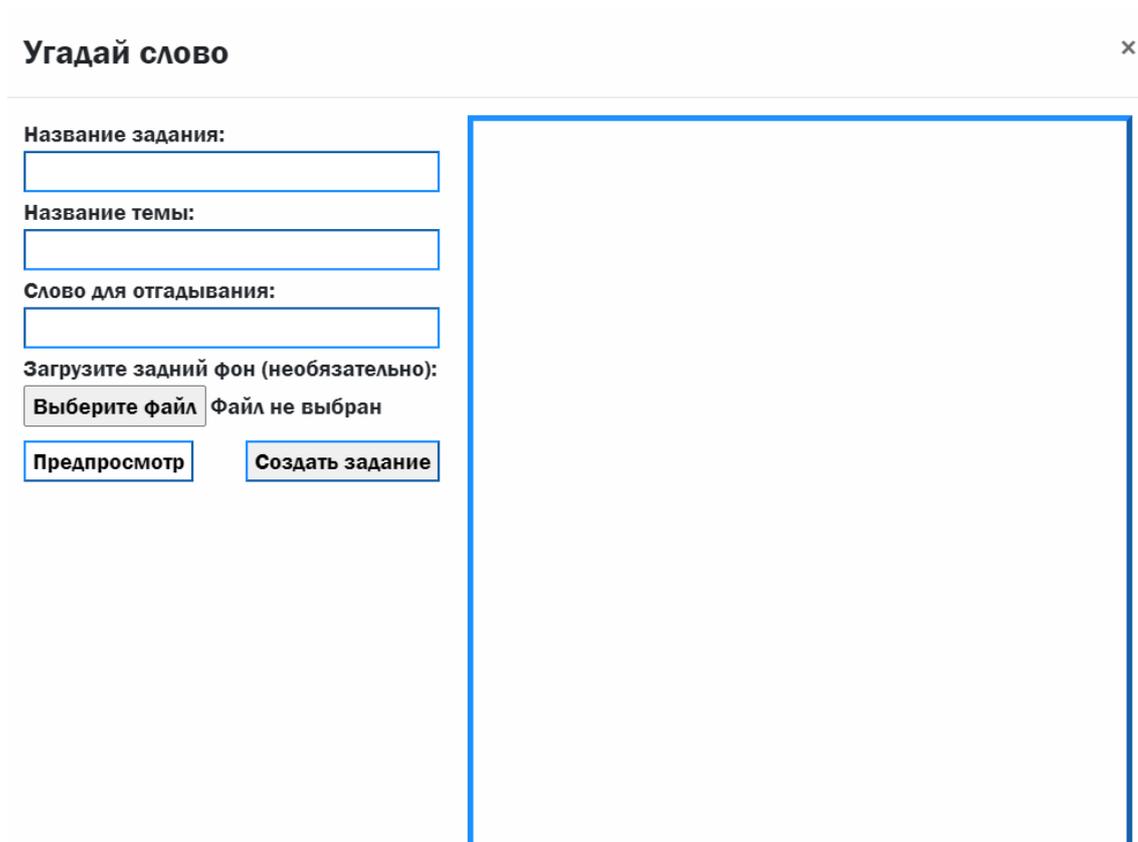
Здесь все шаблоны представлены в виде карточек, на лицевой стороне которых расположены название и изображение, описывающее суть шаблона. Для большей подробности реализована анимация переворачивания карточки с изменением её наполнения. Таким образом, при наведении на шаблон задний фон становится прозрачнее и показывается его описание (рис. 26):

#### Выберите один из представленных шаблонов:



Рисунок 26 – Описание шаблона «Угадай слово»

После того, как пользователь определился с выбранным шаблоном, происходит открытие формы заполнения тела данного шаблона. Данная форма заполнения представляется пользователю модальным окном, в котором, путём заполнения полей происходит наполнение задания. Форма к шаблону является уникальной, то есть для каждого шаблона определён свой набор атрибутов, как это показано на рисунке 27.



The image shows a modal window titled "Угадай слово" (Guess the word) with a close button (x) in the top right corner. The form contains the following elements:

- Label: "Название задания:" followed by an empty text input field.
- Label: "Название темы:" followed by an empty text input field.
- Label: "Слово для отгадывания:" followed by an empty text input field.
- Label: "Загрузите задний фон (необязательно):" followed by a "Выберите файл" button and the text "Файл не выбран".
- Two buttons at the bottom: "Предпросмотр" (Preview) and "Создать задание" (Create task).

Рисунок 27 – Форма к шаблону «Угадай слово»

Задание не может быть создано без заполнения всех основных полей. Опционально для данного шаблона можно загрузить свой задний фон. Существуют ограничения на количество символов, которые можно ввести в каждое поле. Также, для поля «Слово для отгадывания» предусмотрен ввод только строчных русских букв для работы самого шаблона.

Дополнительно предусмотрен запрет создания задания с одноименным названием. То есть, на данном веб-сервисе не может существовать два задания с одинаковыми названиями.

Создание задания предполагает аутентификацию человека, который создает его. Поэтому кнопка «Создать задание» становится неактивной и меняет свой текст на «Войдите в аккаунт», когда пользователь не аутентифицирован.

Реализована функция предпросмотра создаваемого задания. Заполнив все обязательные поля, пользователь может заранее увидеть представление задания, которое он создает.

Пример работы функции предпросмотра показан на рисунке 28.



Рисунок 28 – Демонстрация работы функции предпросмотра

После заполнения всех обязательных полей пользователь создаёт задание нажатием кнопки «Создать задание», что приводит к отправке запроса на добавление задания.

В случае, если пользователь попытается создать задание, название которого уже присутствует в базе данных, будет выдана ошибка, показанная на рисунке 29.

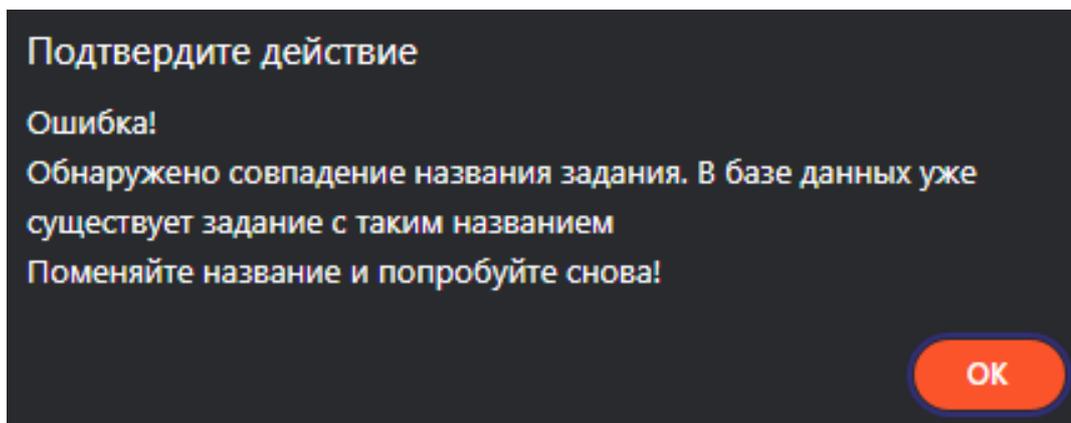


Рисунок 29 – Сообщение об ошибке при совпадении названия

Об успешном создании задания пользователя информирует следующее окно (рис. 30):

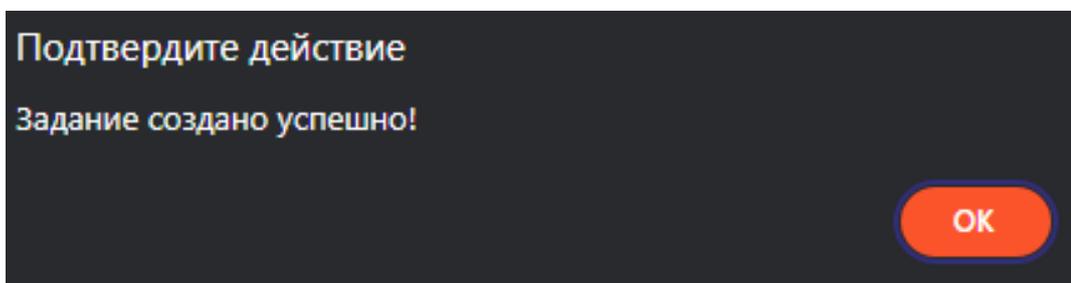


Рисунок 30 – Вид оповещения об успешном создании

Созданное задание сохраняется в таблице отведенного ему шаблона и в таблице всех заданий, доступ к которым можно получить через модуль хранения данных заданий.

### 3.1.3 Модуль хранения данных заданий

Добавление задания в таблицу «Задания» приводит к его появлению в двух отдельных страницах, «Просмотр» и «Мои задания». На данных страницах используется модель задания, включающая в себя внешние данные задания, которые видит пользователь при обращении к заданию.

На странице «Просмотр» представлен поиск задания по определенным критериям. Также, предусмотрена быстрая выборка десяти случайных заданий, когда-либо созданных на веб-сервисе. На рисунке 31 представлен вид страницы «Просмотра».

Показать 10 случайных заданий:

[Случайный поиск](#)

Найти задание:

[Поиск](#)

- Название
- Автор

Показать задания по одному из шаблонов:



Рисунок 31 – Страница просмотра всех заданий

Заполнив текстовое поле и выбрав критерий поиска, пользователь найдёт задание, удовлетворяющее запросу.

Так, при выборе критерия «Название» и вводе текста «Задание №1» будет выведена следующая строка результата (рис. 32):

Результаты поиска: [Вернуться к поиску](#)

ID	Изображение	Название	Автор	Шаблон	Задание
13	 Active Template	Задание №1 5/3/2024 3:48:30 PM	кекман	Угадай слово	<a href="#">Просмотреть</a>

Рисунок 32 – Результат поиска задания по названию

Данный результат поиска предлагает пользователю просмотреть все основные параметры задания:

- ID: идентификационный номер задания;
- Изображение: изображение, которое выбрал пользователь для своего задания (в случае, если пользователь не добавил конкретное изображение для задания, в качестве изображения встает логотип веб-сервиса);
- Название: наименование задания, установленное пользователем при создании;

- Автор: имя автора, создавшего данное задание;
- Шаблон: наименование шаблона, по которому было создано задание;
- Задание: ссылка на задание.

При нажатии на кнопку «Случайный поиск» произойдет выборка десяти случайных заданий, как это представлено на рисунке 33.

Десять случайных заданий: [Вернуться к поиску](#)

ID	Изображение	Название	Автор	Шаблон	Задание
1056	Active Template	Биология, задание 6 5/18/2024 5:14:25 PM	example	Угадай слово	<a href="#">Просмотреть</a>
1059	Active Template	Задание №6 5/18/2024 5:17:23 PM	example	Угадай слово	<a href="#">Просмотреть</a>
1057	Active Template	Периферийные устройства 5/18/2024 5:15:22 PM	example	Угадай слово	<a href="#">Просмотреть</a>
1062	Active Template	Задание №10024 5/18/2024 5:21:10 PM	example	Угадай слово	<a href="#">Просмотреть</a>
1058	Active Template	Задание №10 5/18/2024 5:16:58 PM	example	Угадай слово	<a href="#">Просмотреть</a>
1054	Active Template	Пример 5/18/2024 5:12:47 PM	example	Угадай слово	<a href="#">Просмотреть</a>
1055	Active Template	Задание №5 5/18/2024 5:13:38 PM	example	Угадай слово	<a href="#">Просмотреть</a>
1061	Active Template	Задание №11, работа над ошибками 5/18/2024 5:20:19 PM	example	Угадай слово	<a href="#">Просмотреть</a>
1053	Active Template	Задание №1 5/18/2024 5:12:35 PM	example	Угадай слово	<a href="#">Просмотреть</a>
1060	Active Template	Задание №9 5/18/2024 5:18:12 PM	example	Угадай слово	<a href="#">Просмотреть</a>

Рисунок 33 – Десять случайных заданий

Страница «Мои задания» предоставляет пользователю выборку из всех заданий, которые когда-либо были им созданы. Она имеет следующий вид (рис. 34):

Ваши созданные задания:   Название  Шаблон [Поиск](#)

ID	Изображение	Название	Шаблон	Задание
1053	Active Template	Задание №1 5/18/2024 5:12:35 PM	Угадай слово	<a href="#">Просмотреть</a>
1054	Active Template	Пример 5/18/2024 5:12:47 PM	Угадай слово	<a href="#">Просмотреть</a>
1055	Active Template	Задание №5 5/18/2024 5:13:38 PM	Угадай слово	<a href="#">Просмотреть</a>
1056	Active Template	Биология, задание 6 5/18/2024 5:14:25 PM	Угадай слово	<a href="#">Просмотреть</a>
1057	Active Template	Периферийные устройства 5/18/2024 5:15:22 PM	Угадай слово	<a href="#">Просмотреть</a>
1058	Active Template	Задание №10 5/18/2024 5:16:58 PM	Угадай слово	<a href="#">Просмотреть</a>
1059	Active Template	Задание №6 5/18/2024 5:17:23 PM	Угадай слово	<a href="#">Просмотреть</a>
1060	Active Template	Задание №9 5/18/2024 5:18:12 PM	Угадай слово	<a href="#">Просмотреть</a>
1061	Active Template	Задание №11, работа над ошибками 5/18/2024 5:20:19 PM	Угадай слово	<a href="#">Просмотреть</a>
1062	Active Template	Задание №10024 5/18/2024 5:21:10 PM	Угадай слово	<a href="#">Просмотреть</a>

Рисунок 34 – Страница «Мои задания»

Как и на странице «Просмотр», здесь присутствует поиск по определенным критериям, который позволяет быстро и эффективно найти то, или иное задание.

В случае, если пользователь попытается осуществить переход к несуществующему заданию, будет показана страница ошибки 404, которую можно увидеть на рисунке 35.

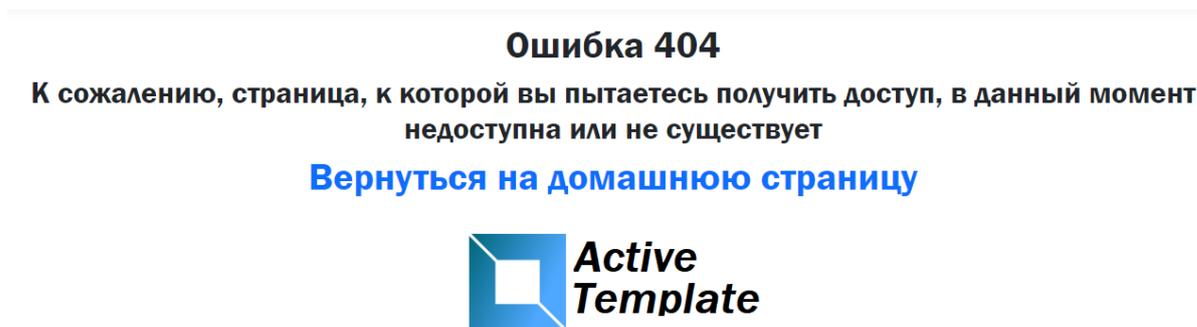


Рисунок 35 – Сообщение об ошибке 404

Это помогает пользователю быстро вернуться в темп взаимодействия с приложением.

К каждому представленному заданию можно перейти и выполнить, после чего результат сохранится в модуле хранения данных результатов.

#### 3.1.4 Модуль хранения данных результатов

После прохождения пользователем одного из заданий происходит сохранение результата его работы на отдельной странице веб-сервиса, «Результаты». В качестве модели данный модуль использует результат, представляющий пользователю сводку о прохождении задания в определенное время, на определенный процент, затрачивая при этом конкретное значение времени.

Страница «Результаты» показывает только те результаты, которые удовлетворяют следующим условиям:

- результат получен с задания, которое было пройдено владельцем аккаунта;

– результат получен от другого пользователя с задания, которое было создано владельцем аккаунта.

Только в этих двух случаях результаты показываются пользователю.

Вид страницы показан на рисунке 36.

**Общие результаты по вашим заданиям:**  [Поиск](#)

ID	ID задания	Процент	Время	Проходящий	Дата
1032	1057 <a href="#">Посмотреть</a>	100%	10 сек.	example	5/18/2024 5:27:04 PM
1033	1057 <a href="#">Посмотреть</a>	100%	18 сек.	example	5/18/2024 5:27:23 PM
1034	1056 <a href="#">Посмотреть</a>	100%	7 сек.	example	5/18/2024 5:28:09 PM
1035	1061 <a href="#">Посмотреть</a>	100%	13 сек.	example	5/18/2024 5:28:28 PM
1036	1054 <a href="#">Посмотреть</a>	100%	12 сек.	example	5/18/2024 5:28:50 PM

Рисунок 36 – Страница «Результаты»

Как и на страницах хранения заданий, страница результатов обладает быстрым поиском, по которому можно выбрать все результаты, относящиеся к конкретному идентификационному номеру задания. Демонстрация работы поиска представлена на рисунке 37.

**Общие результаты по вашим заданиям:**  [Поиск](#)

ID	ID задания	Процент	Время	Проходящий	Дата
1036	1054 <a href="#">Посмотреть</a>	100%	12 сек.	example	5/18/2024 5:28:50 PM

Рисунок 37 – Поиск результатов по заданию

Сущность «Результат» обладает некоторыми параметрами:

- ID: идентификационный номер результата;
- ID задания: идентификационный номер задания;
- Процент: определяет то, на сколько процентов было пройдено задание;
- Время: определяет время прохождения задания;
- Проходящий: логин человека, проходящего задание;
- Дата: дата и время прохождения задания.

Каждое задание обладает своей логикой высчитывания процента его прохождения. Данная логика прописывается программистом при создании шаблона.

### 3.2 Разработка интерфейса взаимодействия модулей

Далее необходимо соединить созданные модули между собой. С помощью платформы создания веб-приложений ASP.NET сначала создается модель веб-сайта с заранее расположенным горизонтальным меню, выступающим в роли «хедера», и «подвалом» для дополнительной информации о веб-сервисе. Во всём остальном веб-приложение является абсолютно пустым по наполнению.

При входе на страницу веб-сервиса не аутентифицированного пользователя встречает домашняя страница, оформленная следующим образом (рис. 38):

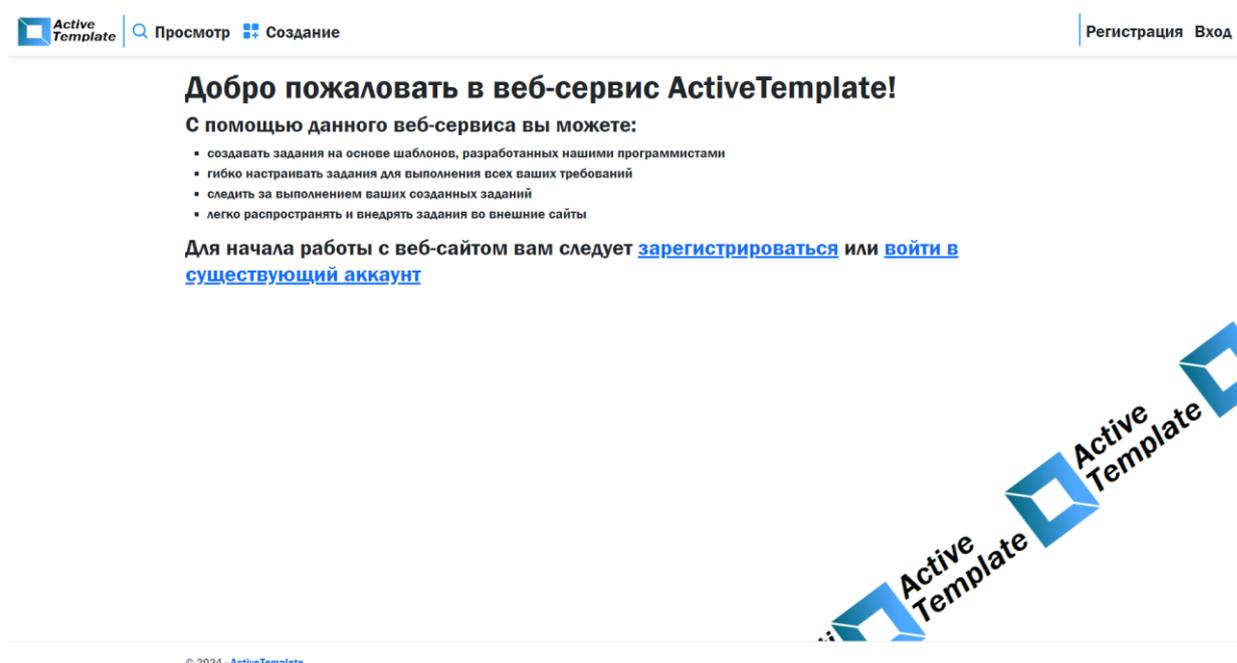


Рисунок 38 – Домашняя страница без аутентификации

Данная страница представляет пользователю основной набор функций, которыми обладает веб-сервис. Предусмотрены отдельные анимации, призванные скрасить статичность страницы.

После аутентификации домашняя страница меняет свой вид на следующий (рис. 39):



Рисунок 39 – Домашняя страница с аутентификацией

Аутентификация пользователя открывает ему все возможности работы с веб-сервисом, становятся доступными страницы «Мои задания» и «Результаты».

Путём добавления специальных элементов взаимодействия происходит оформление веб-приложения. Таким образом реализованы переходы к каждому созданному модулю:

– модуль хранения данных пользователя (кнопки «Регистрация», «Вход», «Логин» верхнего горизонтального меню) представлен на рисунке 40;

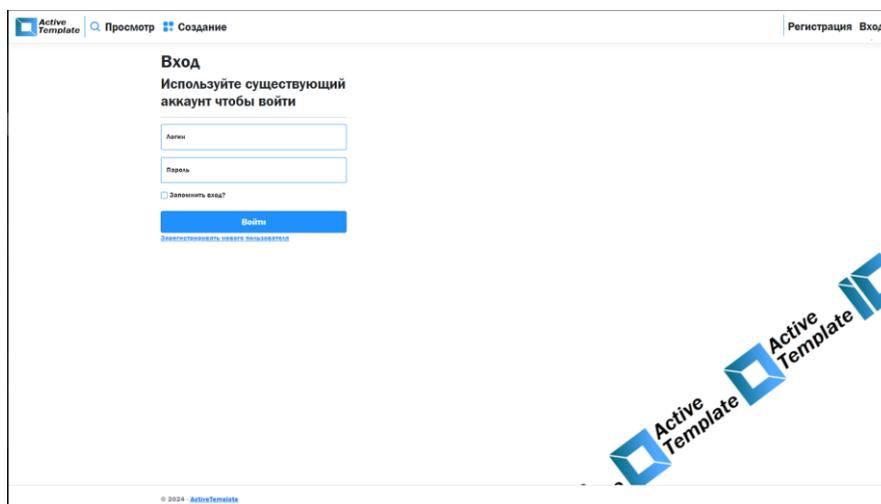


Рисунок 40 – Страница входа в аккаунт

– модуль хранения шаблонов (кнопка «Создание» верхнего горизонтального меню) представлен на рисунке 41;

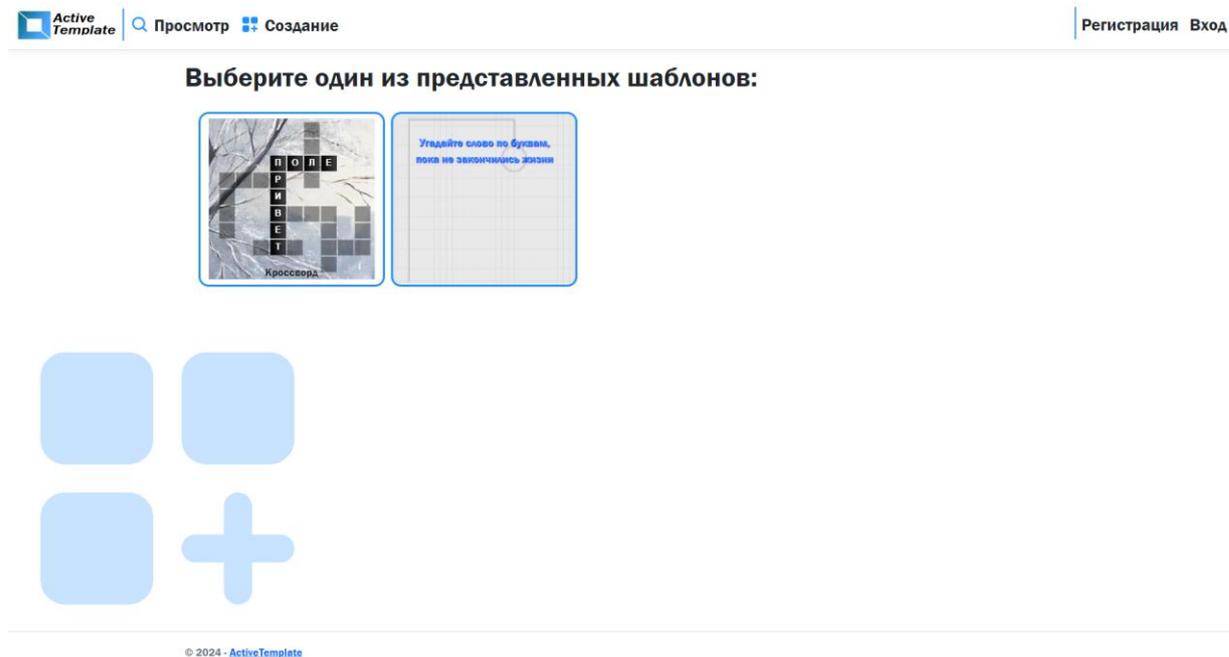


Рисунок 41 – Страница создания заданий

– модуль хранения заданий (кнопки «Просмотр» и «Мои задания» верхнего горизонтального меню) представлен на рисунках 42 и 43;

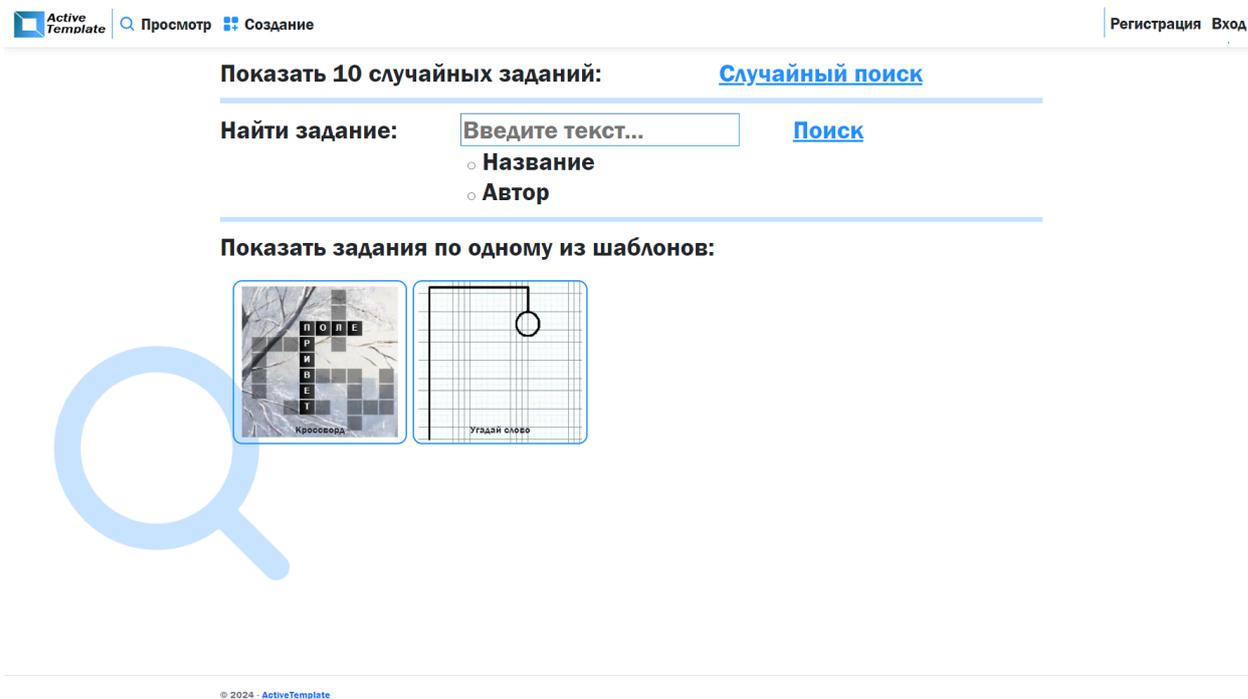


Рисунок 42 – Страница просмотра заданий (выбор критерия поиска)

Десять случайных заданий: [Вернуться к поиску](#)

ID	Изображение	Название	Автор	Шаблон	Задание
1056	 Active Template	Биология, задание 6 5/18/2024 5:14:25 PM	example	Угадай слово	<a href="#">Просмотреть</a>
1060	 Active Template	Задание №9 5/18/2024 5:18:12 PM	example	Угадай слово	<a href="#">Просмотреть</a>
1054	 Active Template	Пример 5/18/2024 5:12:47 PM	example	Угадай слово	<a href="#">Просмотреть</a>
1058	 Active Template	Задание №10 5/18/2024 5:16:58 PM	example	Угадай слово	<a href="#">Просмотреть</a>
1062	 Active Template	Задание №10024 5/18/2024 5:21:10 PM	example	Угадай слово	<a href="#">Просмотреть</a>
1057	 Active Template	Периферийные устройства 5/18/2024 5:15:22 PM	example	Угадай слово	<a href="#">Просмотреть</a>
1053	 Active Template	Задание №1 5/18/2024 5:12:35 PM	example	Угадай слово	<a href="#">Просмотреть</a>
1059	 Active Template	Задание №6 5/18/2024 5:17:23 PM	example	Угадай слово	<a href="#">Просмотреть</a>
1061	 Active Template	Задание №11, работа над ошибками 5/18/2024 5:20:19 PM	example	Угадай слово	<a href="#">Просмотреть</a>
	 Active Template	Задание №5			

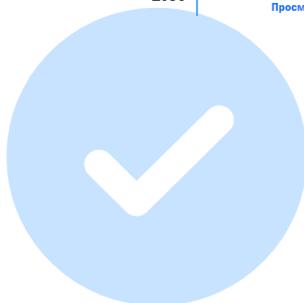
© 2024 - ActiveTemplate

Рисунок 43 – Страница просмотра заданий (результаты поиска)

– модуль хранения результатов (кнопка «Результаты» верхнего горизонтального меню) представлен на рисунке 44.

Общие результаты по вашим заданиям:  [Поиск](#)

ID	ID задания	Процент	Время	Проходящий	Дата
1032	1057 <a href="#">Просмотреть</a>	100%	10 сек.	example	5/18/2024 5:27:04 PM
1033	1057 <a href="#">Просмотреть</a>	100%	18 сек.	example	5/18/2024 5:27:23 PM
1034	1056 <a href="#">Просмотреть</a>	100%	7 сек.	example	5/18/2024 5:28:09 PM
1035	1061 <a href="#">Просмотреть</a>	100%	13 сек.	example	5/18/2024 5:28:28 PM
1036	1054 <a href="#">Просмотреть</a>	100%	12 сек.	example	5/18/2024 5:28:50 PM



© 2024 - ActiveTemplate

Рисунок 44 – Страница результатов

Каждая страница оформлена таким образом, чтобы пользователь всегда мог интуитивно понять, где он находится. Это реализовано путём оформления заднего фона под иконку, соответствующую кнопке страницы.

Результатом этапа разработки является набор файлов созданного проекта (рис. 45):

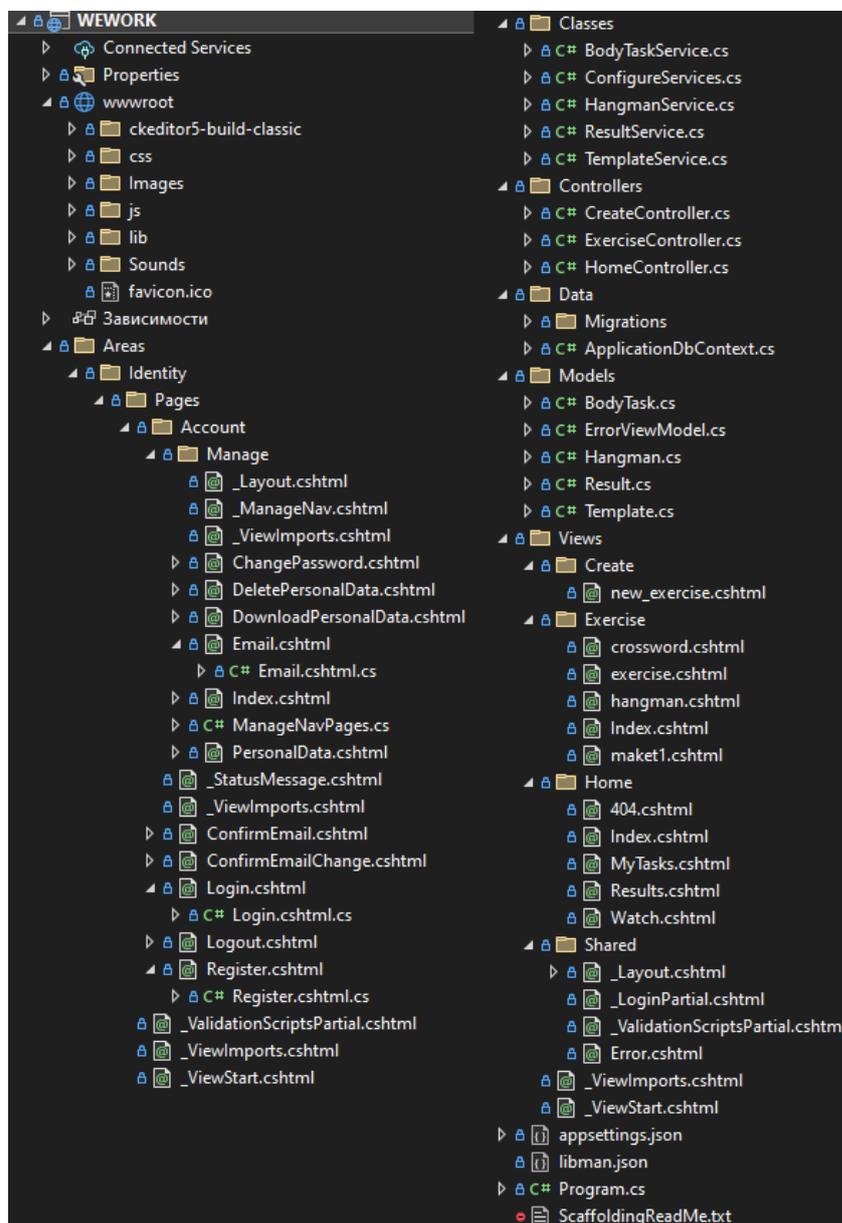


Рисунок 45 – Каталог файлов проекта

Данные файлы разделяются на файлы представлений (Views), моделей (Models) и контроллеров (Controllers). Файлы представлений отвечают за ви-

зуальное оформление страниц. Файлы моделей содержат в себе описание сущностей базы данных. Файлы контроллеров отвечают за соединение фронтенда и бэкенда, то есть контролируют взаимодействие представлений и моделей.

Таким образом, задача разработки интерфейса взаимодействия модулей хранения считается выполненной.

### 3.3 Тестирование работоспособности веб-сервиса

Основной целью функционального тестирования является проверка работоспособности всех функций программного обеспечения. Для этого были выделены следующие основные функции разработанного веб-сервиса:

- регистрация пользователя и создание личного кабинета;
- создание заданий на основе шаблонов;
- сохранение результата прохождения задания;
- возможность рассылки и доступа к заданию без аутентификации;
- поиск заданий по критериям.

#### Регистрация пользователя и создание личного кабинета:

При входе на сайт по ссылке, пользователя встречает домашняя страница, которая рекомендует выполнить вход в аккаунт путём регистрации или аутентификации (рис. 46):

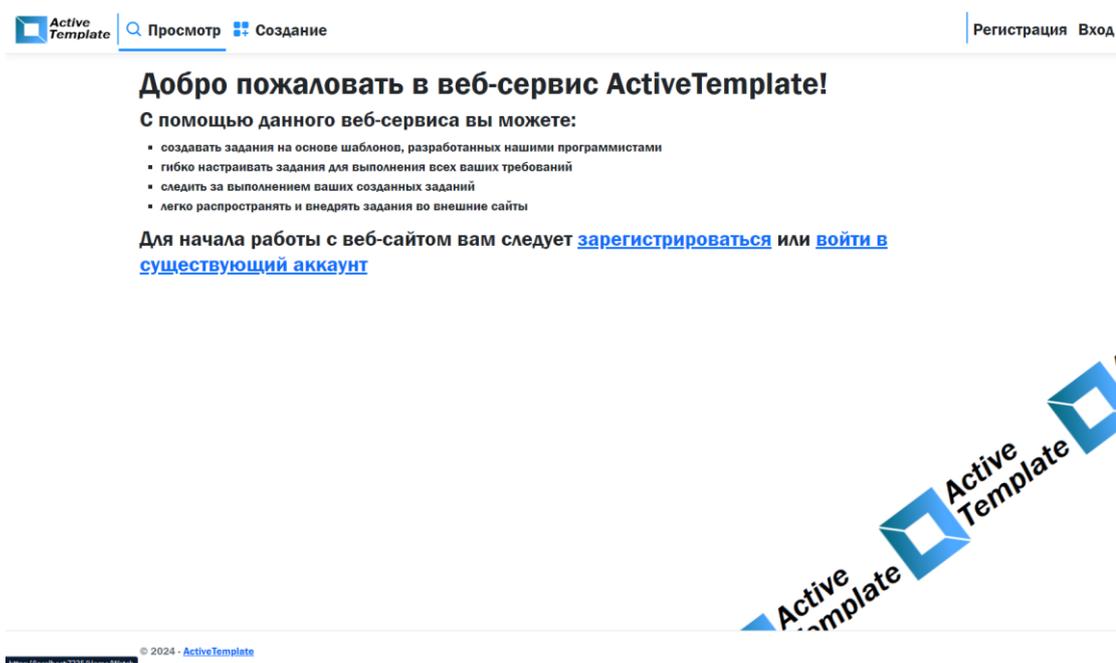


Рисунок 46 – Домашняя страница веб-сервиса

Пользователю, еще не зарегистрированному на сайте, необходимо пройти процесс регистрации по форме, показанной на рисунке 47.

Active Template | Поиск | Просмотр | Создание | Регистрация | Вход

### Регистрация

Создайте новый аккаунт

Электронная почта

Логин

Пароль

Подтверждение пароля

Зарегистрироваться

© 2024 - ActiveTemplate

Рисунок 47 – Страница регистрации на веб-сервисе

При этом каждое поле должно быть заполнено по определенным правилам. Таким образом, электронная почта обязательно должна быть подлинной и иметь символ «@» с указанием домена.

Логин является уникальным атрибутом аккаунта, поэтому существование двух одинаковых значений – исключено. То же самое правило имеет место быть и для указанной электронной почты.

У пароля также есть определенные правила, по которым должен происходить его подбор: длина должна составлять не менее 6 и не более 100 символов соответственно, должна быть хотя бы одна цифра, строчная и заглавная буква.

Регистрация на сервере включает в себя подтверждение подлинности через отсылку письма на электронный адрес. На рисунке 48 представлено

письмо, содержащее в себе ссылку, по которой необходимо перейти для подтверждения регистрации на веб-сервисе.

Подтверждение аккаунта | ActiveTemplate

[Добавить категорию письма](#)

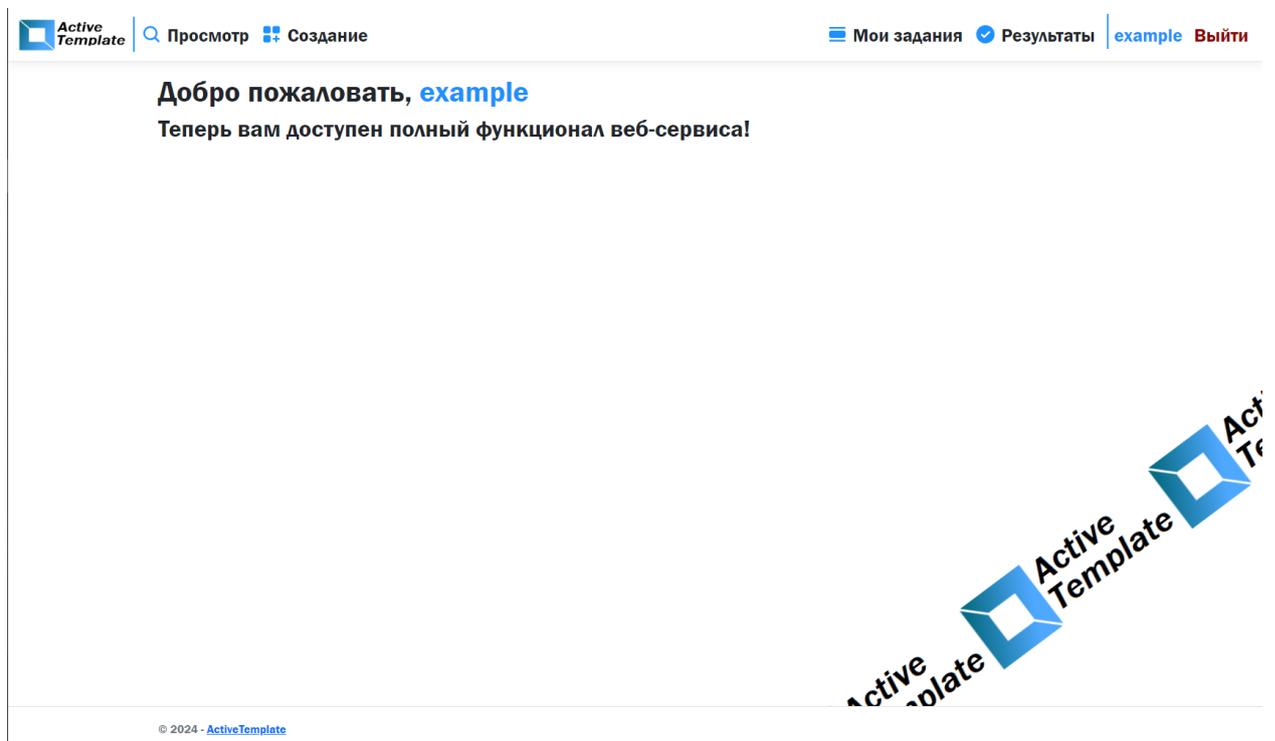
WEWORK 18 мая, 17:09  
Кому: вам

[📧](#) [🔗](#) [📅](#) [🖨️](#) [⋮](#)

<https://localhost:7235/identity/Account/ConfirmEmail?userId=191a42a8-629b-420e-ace6-9a8424ce72dd&code=Q2ZESjhEWk5ZUIErMXU5TnZLSStCtEJNMW9wc0lVvdGNkxyY3pxTXpKUxGldUxGZTI4d2t1SkFwWW10E13UEVvT1dBRldZcDkvR3B1TzJWWW9MRWpGQy9jVWpTMzJXYiNiWkYvaFRPRFhPbXFDVy8wOXRoOVFPem1aR3NZZzJKQzR0QTE3ZUIxQlpUTTBTjREUkxDUm9OeXlXUHFM3RscFAyTXJMQis2b2RWZ2d>

## Рисунок 48 – Письмо подтверждения регистрации на веб-сервисе

После подтверждения аккаунта происходит аутентификация, изменяющая вид домашней страница на следующий (рис. 49):

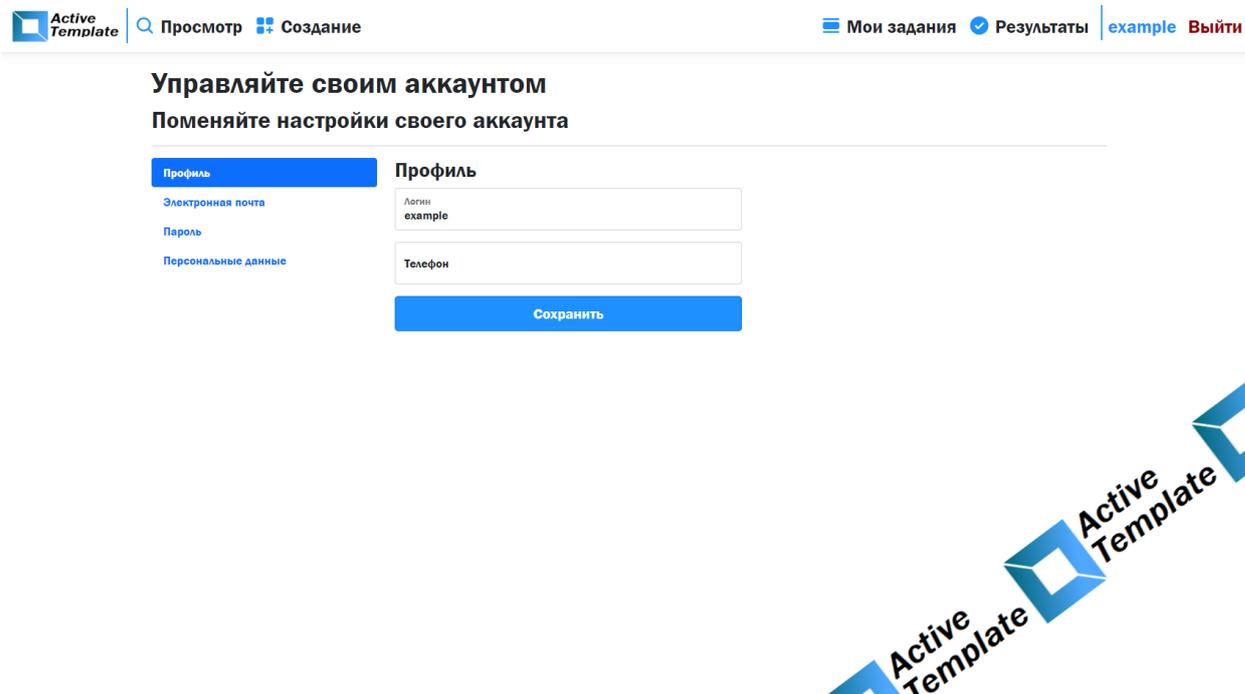


## Рисунок 49 – Домашняя страница после аутентификации

Пользователь волен изменять атрибуты своего аккаунта как ему угодно. Чтобы перейти к личному кабинету необходимо нажать на кнопку верхнего горизонтального меню, текст которого копирует логин аккаунта.

Страница личного кабинета представляет собой набор из вкладок, которые отвечают за конкретный атрибут аккаунта. В каждой такой вкладке можно

поменять значение каждого атрибута аккаунта. На рисунке 50 расположена страница личного кабинета аккаунта.



© 2024 - ActiveTemplate

Рисунок 50 – Страница личного кабинета

Изменение каждого атрибута требует подтверждения действия путём отсылки письма на электронную почту. Например, письмо с подтверждением изменения электронной почты выглядит следующим образом (рис. 51):

#### Смена электронной почты | ActiveTemplate

WEWORK 18 мая, 18:19  
Кому: вам



<https://localhost:7235/Identity/Account/ConfirmEmailChange?userId=191a42a8-629b-420e-ace6-9a8424ce72dd&email=nekitmark@mail.ru&code=Q2ZESjhEWk5ZUIErMXU5TnZLSSiCTEJNMW9wYTNUZmZuK3BWNDNDYi9aVIVzUfKxbno5d2U3ZVNRdXN2UHRZd2VrQTlxVnl4NG9lYysvTTdVNmhtQ3B2eHRPbFVlRjdKOTBkRldVVG5KaEZZdS9lZ0ZxcllxU0gxRDF6Q1FWM2VhRFhoZGN0L0g0czB5YmJPU1R2anZZbUVJukk0SjgyZE5OR2>

Рисунок 51 – Письмо подтверждения смены электронной почты

### Создание заданий на основе шаблонов:

Кнопка «Создание» верхнего горизонтального меню отвечает за страницу создания заданий на основе шаблонов. Данный элемент доступен всем пользователям в независимости от того аутентифицирован ли пользователь или нет.

На рисунке 52 расположена страница создания заданий, предоставляющая выбор из доступных шаблонов.

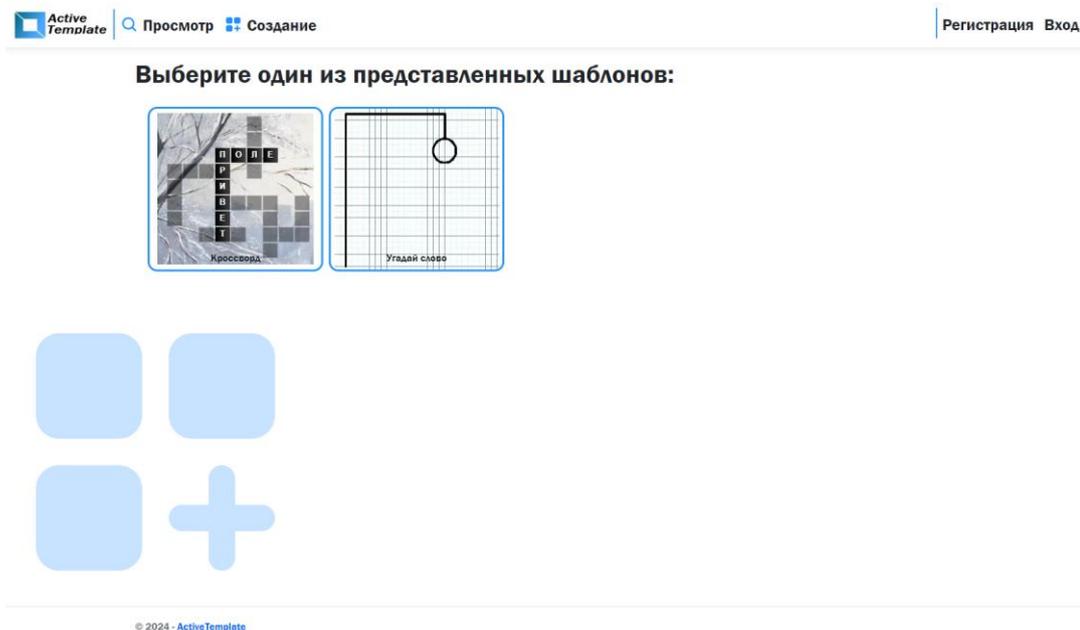


Рисунок 52 – Страница создания заданий

Вид формы шаблонов будет отличаться в зависимости от аутентификации пользователя. Не аутентифицированный пользователь не сможет создать задание, так как кнопка «Создать задание» будет заменена на другую, «Войдите в аккаунт» (рис. 53):

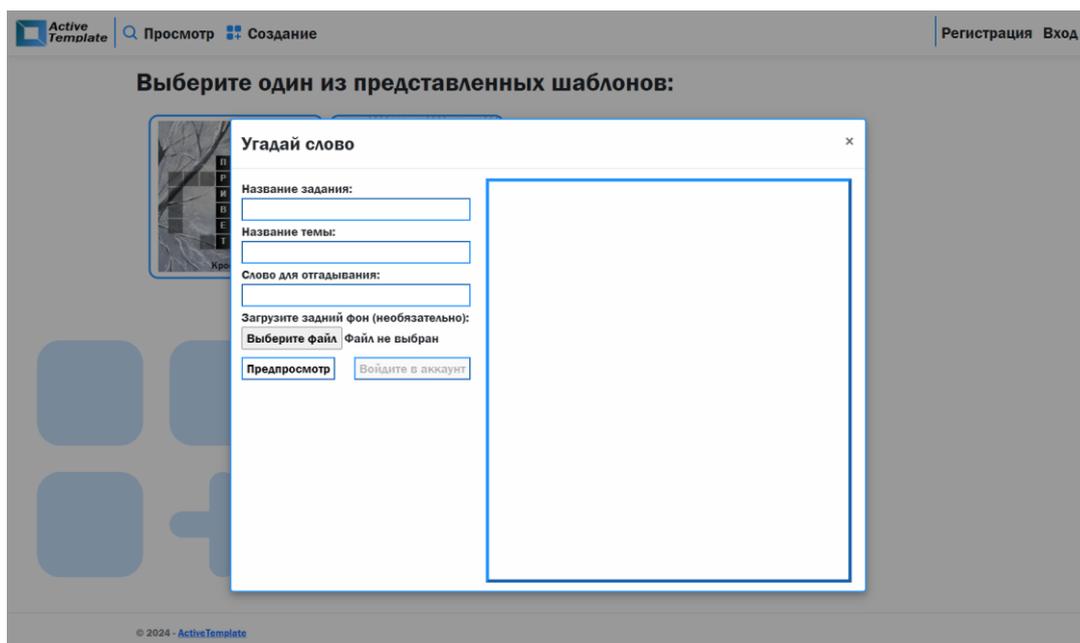


Рисунок 53 – Форма шаблона «Угадай слово» без аутентификации

Однако это не означает, что такому пользователю заблокированы все функции создания заданий. К примеру, любой пользователь может заполнить форму и нажать на кнопку «Предпросмотр», которая заполнит пустое поле предпросмотром задания (рис. 54):

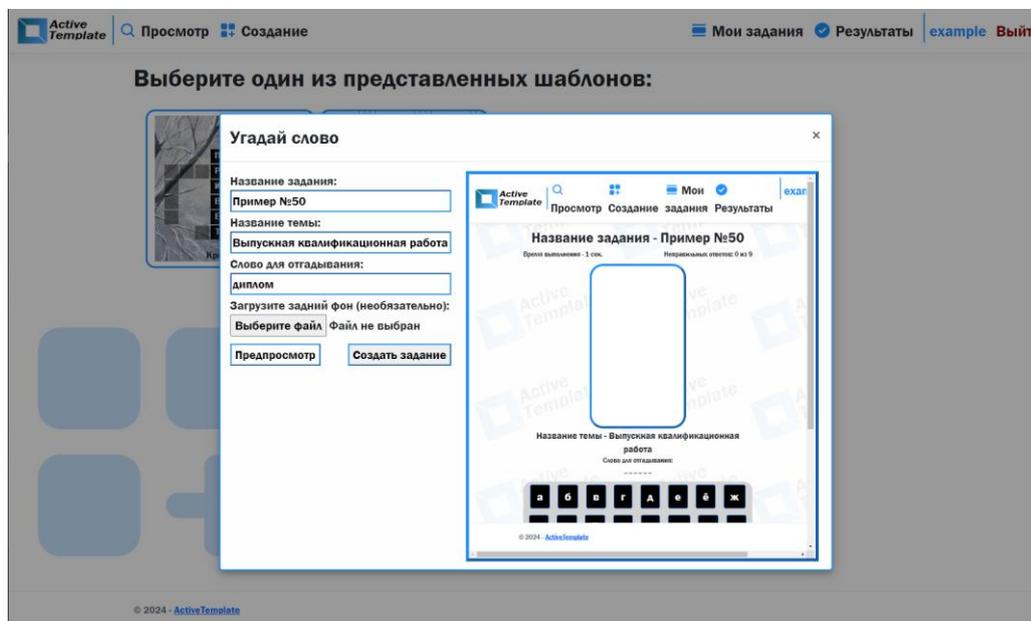


Рисунок 54 – Работа функции предпросмотра задания

Если пользователь аутентифицирован, то он сможет создать подобное задание, нажатием кнопки «Создать задание». На рисунке 55 представлено окно об успешном создании задания.

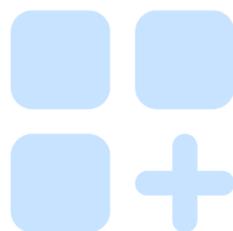


Рисунок 55 – Подтверждение успешного создания задания

Атрибут формы «Название задания» является уникальным, что говорит о невозможности его совпадения с другими заданиями. В случае, если пользователь пытается создать задание с существующим названием, будет выведено поясняющее окно об ошибке (рис. 56):

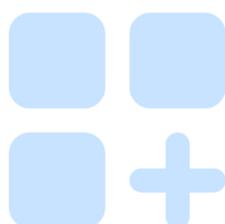
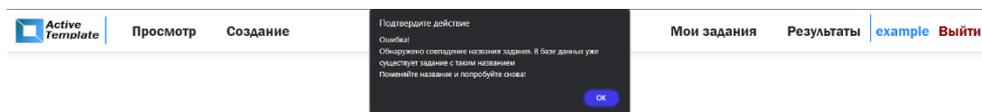


Рисунок 56 – Окно об ошибке создания задания

### Сохранение результата прохождения задания:

Созданное задание можно найти на странице кнопок «Мои задания» и «Просмотр». На рисунке 57 представлена страница «Мои задания» с заданием, созданным на рисунке 54.

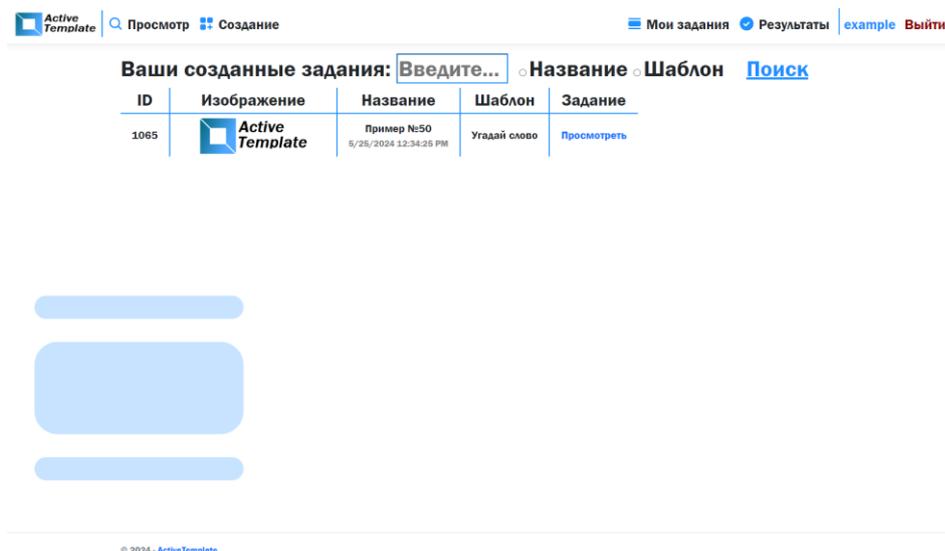


Рисунок 57 – Страница «Мои задания»

После нажатия на ссылку задания происходит его открытие (рис. 58):

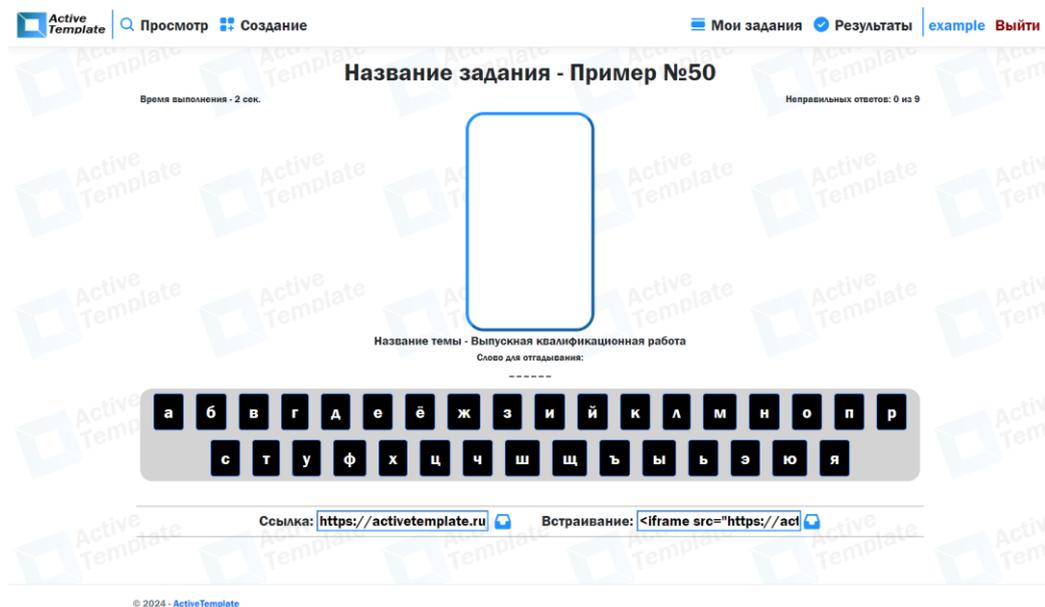


Рисунок 58 – Задание, созданное на рисунке 54

На данном рисунке дополнительно можно увидеть реализацию следующей функции, возможность рассылки и доступа к заданию без аутентификации, оформленная в виде двух полей ввода в нижней части задания. Иконки, расположенные рядом с данными полями, позволяют быстро скопировать содержимое элемента.

После выполнения задания страница примет следующий вид (рис. 59):

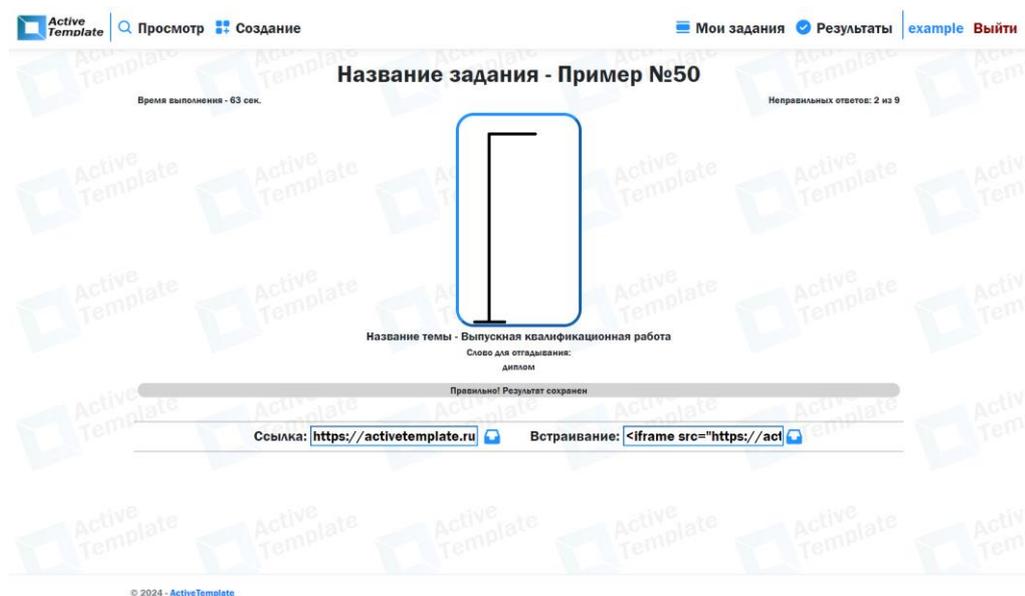


Рисунок 59 – Выполненное задание с аутентификацией

В случае, если пользователь не аутентифицирован, сохранение результата не произойдет и страница примет вид, представленный на рисунке 60.

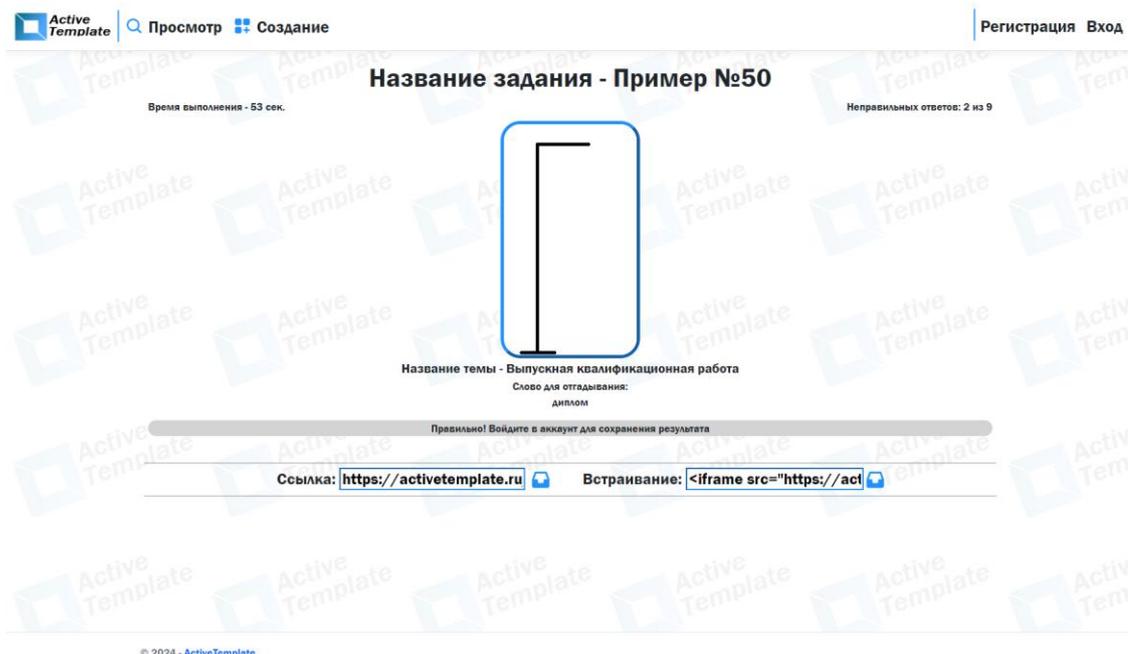


Рисунок 60 – Выполненное задание без аутентификации

Результаты прохождения заданий, созданных пользователем, и результаты прохождения пользователя расположены на странице «Результаты» (рис. 61):

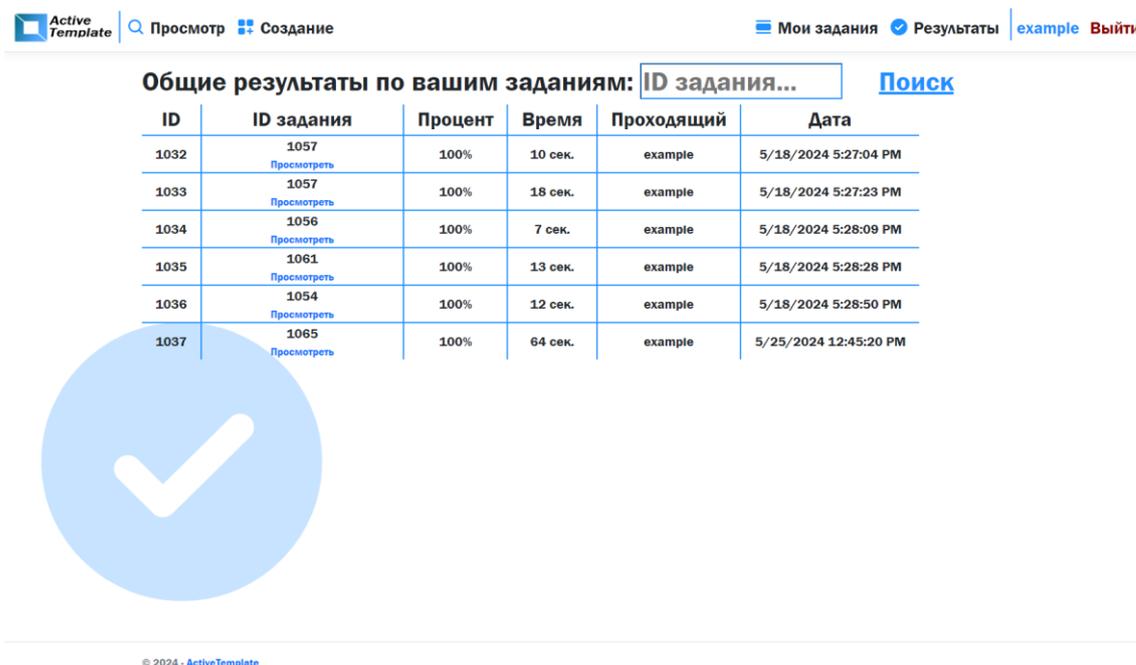


Рисунок 61 – Страница «Результаты»

На данной странице расположен результат прохождения задания, расположенного на рисунке 59. Идентификационный номер данного результата – 1037.

### Поиск заданий по критериям:

С целью доступа к любому созданному заданию реализована страница «Просмотр», на которой можно осуществить поиск заданий по критериям. Вид страницы «Просмотр» расположен на рисунке 62.

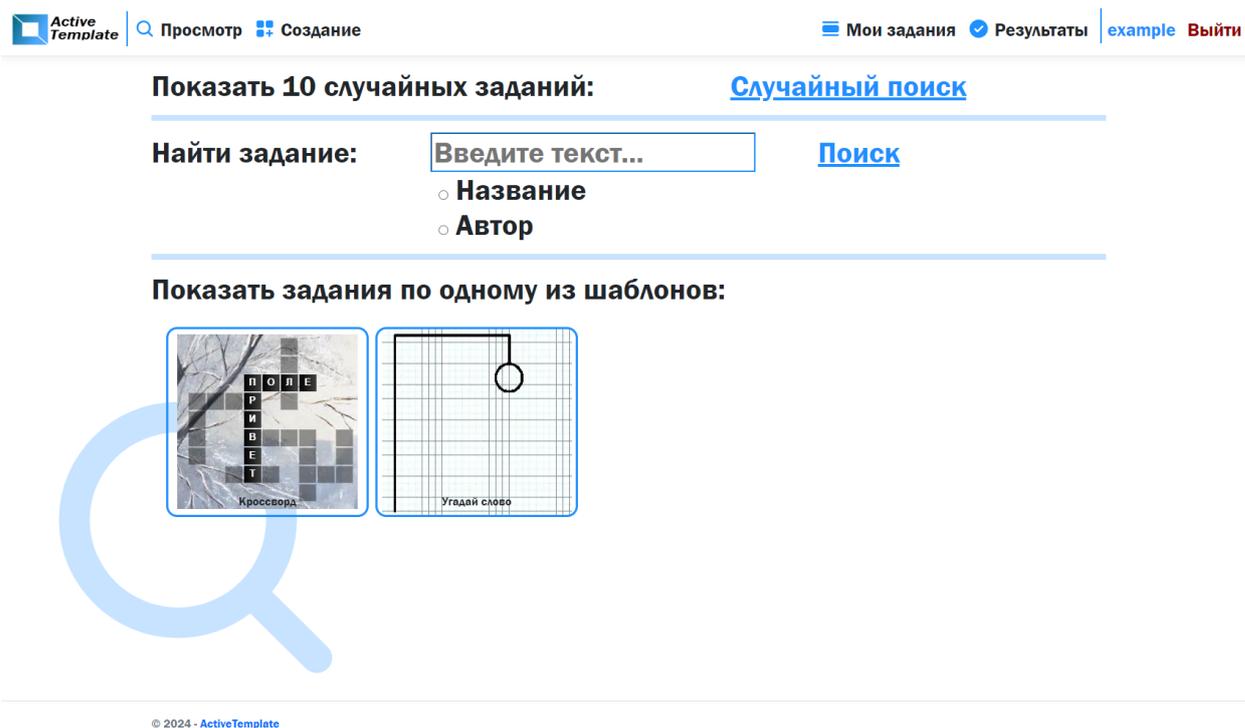


Рисунок 62 – Страница «Просмотр»

Пользователю даётся выбор в поиске заданий:

- показ десяти случайных заданий;
- поиск заданий по названию или автору;
- показ заданий по доступным шаблонам.

Показ десяти случайных заданий реализует выборку из десяти идентификационных номеров заданий, которые затем формируются в таблицу. Работа случайного показа заданий представлена на рисунке 63.

Десять случайных заданий: [Вернуться к поиску](#)

ID	Изображение	Название	Автор	Шаблон	Задание
1065	 Active Template	Пример №50 5/25/2024 12:34:25 PM	example	Угадай слово	<a href="#">Просмотреть</a>
1062	 Active Template	Задание №10024 5/18/2024 5:21:10 PM	example	Угадай слово	<a href="#">Просмотреть</a>
1057	 Active Template	Периферийные устройства 5/18/2024 5:15:22 PM	example	Угадай слово	<a href="#">Просмотреть</a>
1059	 Active Template	Задание №6 5/18/2024 5:17:23 PM	example	Угадай слово	<a href="#">Просмотреть</a>
1055	 Active Template	Задание №5 5/18/2024 5:13:38 PM	example	Угадай слово	<a href="#">Просмотреть</a>
1053	 Active Template	Задание №1 5/18/2024 5:12:35 PM	example	Угадай слово	<a href="#">Просмотреть</a>
1064	 Active Template	Задание №1001 5/20/2024 6:00:08 PM	example	Угадай слово	<a href="#">Просмотреть</a>
1054	 Active Template	Пример 5/18/2024 5:12:47 PM	example	Угадай слово	<a href="#">Просмотреть</a>
1063	 Active Template	Проверочное задание 5/18/2024 6:41:53 PM	example	Угадай слово	<a href="#">Просмотреть</a>
1061	 Active Template	Задание №11, работа над ошибками 5/18/2024 5:00:00 PM	example	Угадай слово	<a href="#">Просмотреть</a>

© 2024 - ActiveTemplate

Рисунок 63 – Десять случайных заданий

Случай, когда на веб-сервисе нет десяти заданий, предполагает показ определенной страницы, показанной на рисунке 64.

**Ошибка!**

К сожалению, на веб-сервисе в данный момент находится меньше десяти созданных заданий  
Создайте больше заданий для корректной работоспособности данной функции

[Вернуться на страницу просмотра](#)

© 2024 - ActiveTemplate

Рисунок 64 – Ошибка показа десяти случайных заданий

Поиск заданий по названию или автору реализует выборку заданий, атрибуты которых удовлетворяют запросу поиска. Таким образом, если пользователь попытается найти задание под названием «Пример №50», то будет выведена следующая страница (рис. 65):

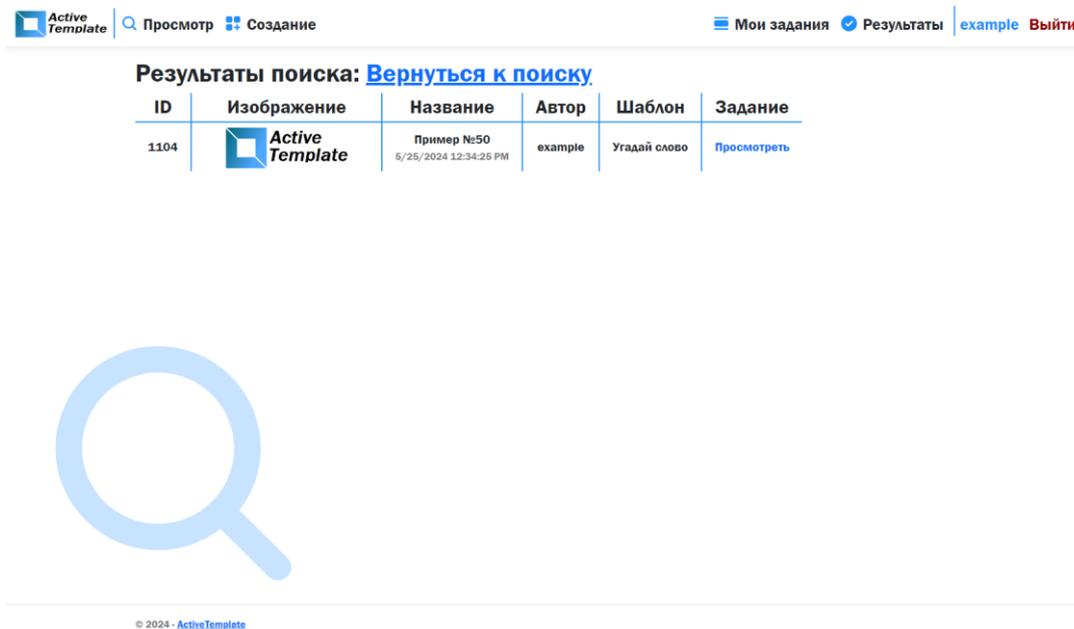


Рисунок 65 – Результат поиска задания с названием «Пример №50»

В случае, если не было найдено результатов по запросу поиска, будет выведено следующее (рис. 66):

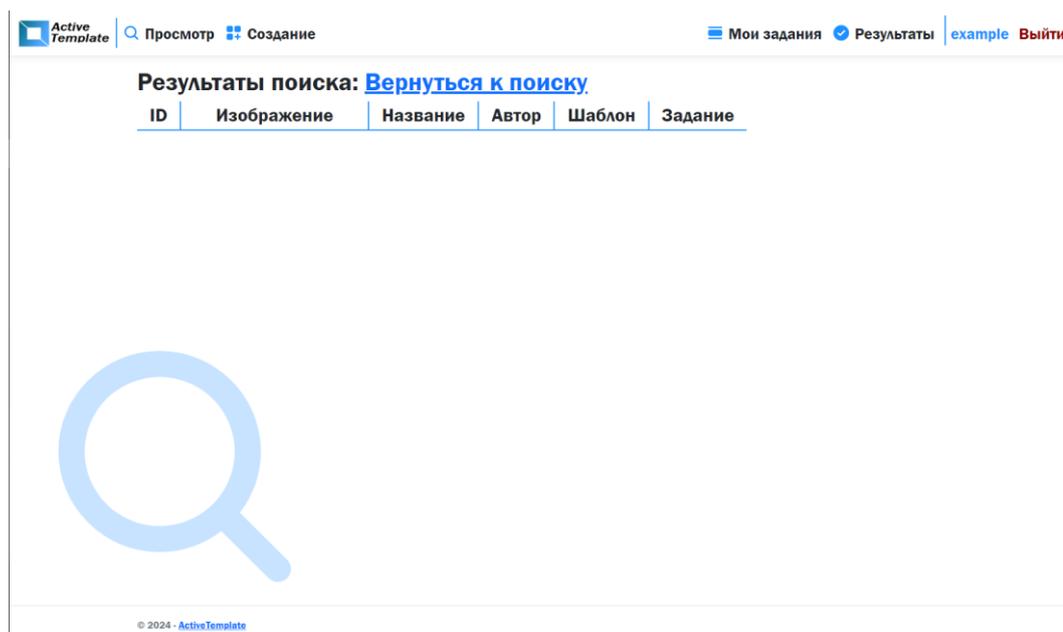


Рисунок 66 – Нулевой результат поиска

Показ заданий по доступным шаблонам реализует выборку всех заданий, в которых атрибутом названия шаблона является конкретный шаблон. Таким образом, если пользователь выберет шаблон «Угадай слово», то будет выведена следующая страница (рис. 67):

Active Template | Поиск | Просмотр | Создание | Мои задания | Результаты | example | Выйти

Результаты поиска: [Вернуться к поиску](#)

ID	Изображение	Название	Автор	Шаблон	Задание
1092		Задание №1 5/18/2024 5:12:35 PM	example	Угадай слово	<a href="#">Просмотреть</a>
1093		Пример 5/18/2024 5:12:47 PM	example	Угадай слово	<a href="#">Просмотреть</a>
1094		Задание №5 5/18/2024 5:13:38 PM	example	Угадай слово	<a href="#">Просмотреть</a>
1095		Биология, задание 6 5/18/2024 5:14:25 PM	example	Угадай слово	<a href="#">Просмотреть</a>
1096		Периферийные устройства 5/18/2024 5:15:22 PM	example	Угадай слово	<a href="#">Просмотреть</a>
1097		Задание №10 5/18/2024 5:16:58 PM	example	Угадай слово	<a href="#">Просмотреть</a>
1098		Задание №6 5/18/2024 5:17:23 PM	example	Угадай слово	<a href="#">Просмотреть</a>
1099		Задание №9 5/18/2024 5:18:12 PM	example	Угадай слово	<a href="#">Просмотреть</a>
1100		Задание №11, работа над ошибками 5/18/2024 5:20:19 PM	example	Угадай слово	<a href="#">Просмотреть</a>
1101		Задание №10024 5/18/2024 5:21:10 PM	example	Угадай слово	<a href="#">Просмотреть</a>

© 2024 - ActiveTemplate

Рисунок 67 – Результат поиска заданий по шаблону «Угадай слово»

В результате функционального тестирования не были выявлены какие-либо ошибки или неисправности. Каждая реализованная функция работает корректно.

## 4 БЕЗОПАСНОСТЬ И ЭКОЛОГИЧНОСТЬ

### 4.1 Безопасность

Безопасность рабочего места предполагает отсутствие рисков для вреда здоровью человека при длительной работе. Безопасное рабочее место должно быть удобным и эргономичным.

Технологический процесс включает в себя длительную работу за компьютером сидя. Данная рабочая позиция может нанести вред здоровью трудящегося при несоблюдении определенных норм и правил, установленных федеральным органом исполнительной власти.

При организации рабочего места необходимо постоянно ссылаться на эргономику, которая собой представляет научную дисциплину, изучающую взаимодействие человека и других элементов системы. Полное соблюдение норм эргономики является залогом безопасного рабочего места.

При рассмотрении эргономики рабочего места важно уделять особое внимание основным предметам, которые, при неправильном использовании, могут нанести вред. Этими предметами риска являются кресло, стол, пользовательский интерфейс и экран. Эргономика предполагает, что мебель, используемая для работы сидя перед монитором, должна быть пригодной для использования и доступной.

Проектирование (или перепроектирование) пригодности использования включает рассмотрение вопросов пригодности использования на всех этапах жизненного цикла, включая концепцию, детальное проектное решение, оценку, внедрение, долговременное применение, обслуживание, утилизацию и переработку.

Доступность – это свойство продукции, систем, услуг, сред или сооружений, при наличии которого они могут быть использованы людьми из совокупности с самым широким диапазоном возможностей для достижения установленных целей в определенных условиях использования.

Правильное использование кресла предотвращает пользователя от возможных последствий, связанных со здоровьем. Сиденье кресла должно быть расположено таким образом, чтобы сидя можно было коснуться стопами пола. Высота сиденья стула от пола зависит от высоты стола: при высоте стола (720-780) мм удобен стул (420-480) мм. Спинка должна быть откалибрована таким образом, чтобы как можно больше уменьшить давление на поясницу. Подлокотники должны быть настроены так, чтобы при сгибании рук пользователь лишь слегка касался их локтями. Они не должны давить на них и уж тем более поднимать их выше естественного положения. В случае, если пользователь всё ещё чувствует дискомфорт при правильной настройке мебели, это говорит о нарушении эргономики рабочего места. Необходимо уделить особое внимание расположению рабочей поверхности. Важно, чтобы при положении сидя пользователь не чувствовал дискомфорт и не отвлекался от работы на проблемы рабочего места.

Самым важным для стола является достаточное рабочее пространство. Его должно быть достаточно для размещения на нём периферийных устройств и остальных канцелярских и офисных предметов. Высота письменного стола должна составлять 750 мм. Ширина и глубина стола должны быть адаптированы под количество оборудования, которое будет стоять на столе. Клавиатура должна находиться на столе. Если использование стола приносит дискомфорт, необходимо срочно принять меры, иначе возможен риск ухудшения здоровья, выражающееся в виде проблем со здоровьем позвоночника.

Согласно санитарным нормам, предельно допустимый уровень шума на работе не должен превышать 70 децибел. Длительное воздействие шума, превышающего данный уровень, может привести к проблемам со слухом и дезориентации. Для устранения данной проблемы при проектировании комнат рекомендуется использовать звукоизоляционные краски, акустические панели и шумоподавляющие напольные покрытия. Если ремонт невозможен, то рекомендовано использовать стулья с высокими спинками и высокие шкафы.

Также, можно поставить перегородки для огораживания мест от посторонних звуков.

Монитор должен быть расположен так, чтобы верхний край матрицы был на уровне глаз. Рекомендуемое расстояние от глаз до монитора (50-70) см. Расположить монитор лучше так, чтобы можно было время от времени сфокусировать глаза на отдалённых объектах, и, если есть возможность, то у окна.

Чтобы пользователь мог точно считывать информацию с дисплея незащищенным глазом и не испытывать при этом дискомфорт необходимо соблюдать определенное сочетание значений яркости и контраста изображения, внешней освещенности экрана, углового размера знака и угла наблюдения экрана. Таким образом, яркость знака должна быть не менее 20 кд/м<sup>2</sup> для плоских дискретных экранов. Данные экраны в настоящее время составляют абсолютное большинство всех мониторов в мире. При этом яркостный контраст изображения должен быть не менее 3:1.

Государственным стандартом Р 50949-2001 установлены методы измерения и оценки следующих эргономических параметров и параметров безопасности, регламентированных ГОСТ Р 50948-2001:

- яркость изображения;
- неравномерность яркости рабочего поля экрана;
- неравномерность яркости элементов знаков на плоских дискретных экранах;
- контрастность деталей изображения;
- контрастность изображения на плоских экранах в зависимости от угла наблюдения;
- контрастность соседних уровней кодирования яркостью;
- постоянство размера знака;
- ширина контура знака;
- пространственные искажения изображения по рабочему полю;
- пространственная нестабильность изображения (дрожания);

- несведение цветов;
- временная нестабильность изображения (мелькания);
- электростатический потенциал экрана;
- напряженность переменного электрического поля;
- плотность магнитного потока.

Соблюдение корректных значений каждого представленного параметра способствует уменьшению влияния плоских дискретных мониторов на здоровье пользователя.

Графический интерфейс пользователя должен быть построен согласно государственному стандарту во избежание возможных проблем с эффективностью взаимодействия пользователя с системой. Все элементы должны быть интуитивно понятными пользователю и не вызывать вопросов при их использовании. В случае, если не удастся четко пояснить функцию элемента, то стоит указать пример правильного использования данного элемента.

На данный момент интерфейс веб-сервиса оснащен следующими графическими элементами:

- чек-бокс (рис. 68): необходим при отметке положительного или отрицательного значения параметра;

**Запомнить вход?**

Рисунок 68 – Чек-бокс сохранения входа в аккаунт

- диалоговое окно (рис. 69): необходимо для оповещения пользователя о результате действия;

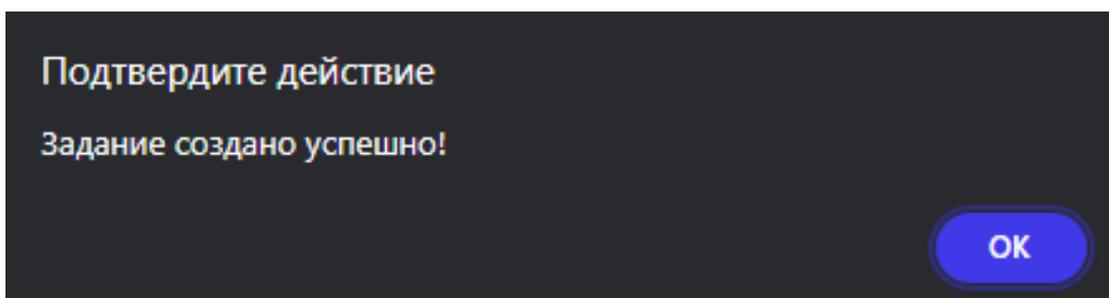


Рисунок 69 – Диалоговое окно о создании задания

– поле ввода (рис. 70): необходимо для ввода данных при том или ином действии;

**Название задания:**

Проверочное

Рисунок 70 – Поле ввода названия задания

– поясняющая информация (рис. 71): необходимо для пояснения чего-либо в программе;

## Добро пожаловать в веб-сервис ActiveTemplate!

С помощью данного веб-сервиса вы можете:

- создавать задания на основе шаблонов, разработанных нашими программистами
- гибко настраивать задания для выполнения всех ваших требований
- следить за выполнением ваших созданных заданий
- легко распространять и внедрять задания во внешние сайты

Для начала работы с веб-сайтом вам следует [зарегистрироваться](#) или [войти в существующий аккаунт](#)

Рисунок 71 – Поясняющая информация на домашней странице веб-сервиса

– надпись (рис. 72): необходима для описания части программы;

**Выберите один из представленных шаблонов:**



Рисунок 72 – Надпись создания задания по одному шаблону

– ссылка / гиперссылка (рис. 73): используется для перехода на другую страницу или выполнения запроса;

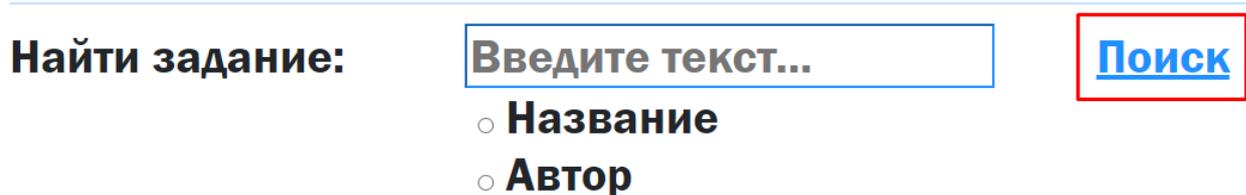


Рисунок 73 – Ссылка на выполнение поиска по параметрам

– кнопка списка/кнопка меню (рис. 74): используется для отображения элементов меню;

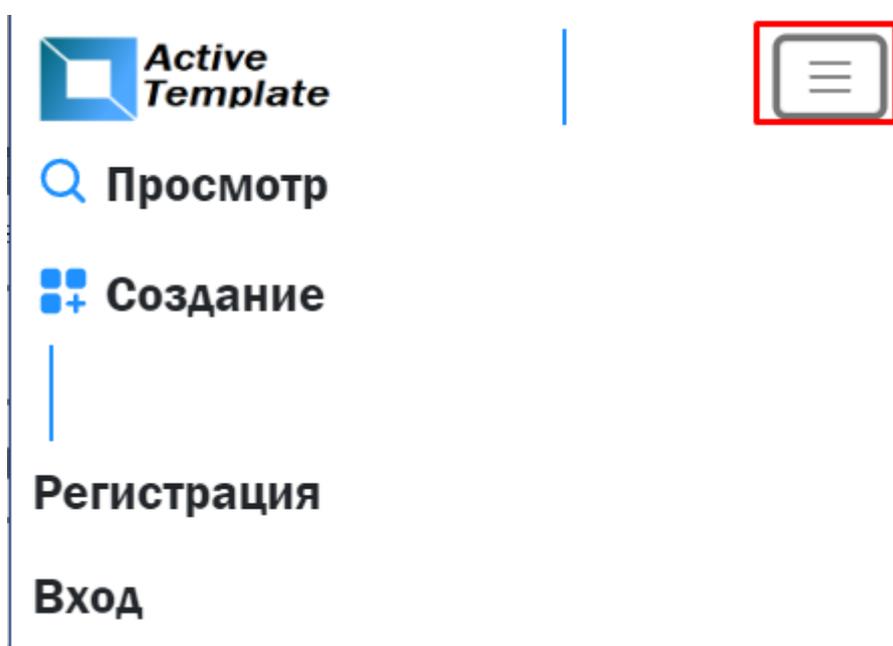


Рисунок 74 – Кнопка основного меню веб-сервиса

– нажимаемая кнопка (рис. 75): необходима для немедленного выполнения запроса;



Рисунок 75 – Нажимаемая кнопка входа в аккаунт

– радиокнопка (рис. 76): используется в группе других радиокнопок, среди которых активной может быть только одна, выполняет функцию чек-бокса;

**Найти задание:**  [Поиск](#)

**Название**

**Автор**

Рисунок 76 – Группа радиокнопок выбора параметра поиска

– поле только для чтения (рис. 77): необходимо для вывода информации, используемой только для чтения, в поле ввода;

**Ссылка:**  

Рисунок 77 – Поле только для чтения ссылки на задание

– таблица (рис. 78): используется для вывода большого числа информации на одну страницу;

ID	ID задания	Процент	Время	Проходящий	Дата
1032	1057 <a href="#">Просмотреть</a>	100%	10 сек.	example	5/18/2024 5:27:04 PM
1033	1057 <a href="#">Просмотреть</a>	100%	18 сек.	example	5/18/2024 5:27:23 PM
1034	1056 <a href="#">Просмотреть</a>	100%	7 сек.	example	5/18/2024 5:28:09 PM
1035	1061 <a href="#">Просмотреть</a>	100%	13 сек.	example	5/18/2024 5:28:28 PM
1036	1054 <a href="#">Просмотреть</a>	100%	12 сек.	example	5/18/2024 5:28:50 PM

Рисунок 78 – Таблица результатов прохождения заданий

– подсказка (рис. 79): необходимо для оповещения пользователя о необходимости выполнения того или иного действия.

**Название задания:**

**Название**  Заполните это поле.

Рисунок 79 – Подсказка о необходимости заполнения поля

Каждый элемент выполнен согласно ГОСТ Р ИСО 9241-161-2016, учитывая описание их действия, состояния и условия использования.

Графический интерфейс выполнен в светлых тонах с акцентированным светло-синим цветом («dodgerblue», #1E90FF). Данная цветовая палитра позволяет пользователю видеть и сканировать глазом информацию с экрана при любой освещенности комнаты. Дополнительно, у пользователей подобный цвет ассоциируется со спокойствием, безопасностью и открытостью, что положительно влияет на взаимодействие пользователя с системой. Также, во всем веб-сервисе используется шрифт с кеглем в 20 точек, что облегчает читаемость информации.

Помимо эргономики, веб-сервис оснащён системой информационной безопасности. Доступ к базе данных может быть осуществлен только напрямую через физический терминал, а данные внутри зашифрованы от несанкционированного доступа к ним. Таким образом, данные от аккаунта пользователя при регистрации преобразуются в хэш и в данной форме хранятся в базе данных.

## **4.2 Экологичность**

Для работы с веб-сервисом необходим хороший доступ к сети Интернет. При каждой загрузке веб-сервиса происходит полная выгрузка информации с сервера на клиент пользователя, которая, в свою очередь, потребляет немалое число Интернет-трафика. Чем больше весят страницы веб-сайта, тем больше трафика придется загружать, выпуская тем самым в атмосферу тонны CO<sub>2</sub>.

Чтобы предотвратить увеличение выбросов CO<sub>2</sub> в атмосферу рекомендуется использовать: масштабируемую архитектуру, экологичных провайдеров, правильные языки программирования и оптимизированные веб-страницы.

Масштабируемая архитектура предполагает такую архитектуру, которая бы в отсутствии высоких нагрузок отключала работающие вхолостую сервера.

Это поможет снизить затраты на электричество и увеличить отказоустойчивость веб-сайта.

Интернет-трафик зачастую проходит через самые дальние уголки мира, что может вызывать задержку. Данная проблема может не только увеличить задержку трафика, но и увеличить энергопотребление на различных серверах. Во избежание этого следует пользоваться экологичными провайдерами или поставщиками интернета, которые позволят самому выбрать сервер, работающий на возобновляемых источниках энергии.

Выбор правильных языков программирования поможет не только сократить потребляемый интернет-трафик, но и увеличить производительность веб-сайта.

Размер средней веб-страницы стал превышать больше 3 мегабайт, что является колоссальным объемом информации по меркам начала нулевых годов данного тысячелетия, и с каждым годом размер веб-страницы лишь растёт. Чтобы предотвратить это рекомендуется использовать как можно меньше неиспользуемого кода. Оптимизация отдельных методов в языках программирования приветствуется.

Изображения на сайтах являются основным источником углеродного следа и в среднем составляют 60 % всего объема ресурсов, их оптимизация может сократить нагрузку на сайт. Кроме того, многие изображения в вебе сохранены в неправильном формате, имеют реальные размеры, несоответствующие отображаемому, или плохо оптимизированы. Фаза разработки, когда определяется пользовательское взаимодействие, является идеальным временем для устранения требовательных к ресурсам элементов сайта. Стоит отказаться от слишком сложных и изощренных элементов интерфейса, таких как «Карусель». Данный элемент состоит из пары картинок, которые и так уже несут в себе немалый объем данных, но и при этом язык JavaScript, необходимый для её корректной работы, добавляет несколько сотен килобайт к общему весу элемента.

Также следует пересмотреть необходимость размещения на сайте кнопок публикации в социальные сети, встраиваемых карт, видео с автоматическим воспроизведением, Flash-контента, рекламных баннеров и встраиваемого контента от третьих лиц, так как все они требуют загрузки большого объема дополнительных данных.

Чтобы сделать Интернет более доступным и снизить его воздействие на окружающую среду рекомендуется:

- выбирать хостинг, пользующийся серверами, работающими на возобновляемых источниках энергии;
- разрабатывать веб-сайт с нуля, а не с шаблона;
- отдельно сохранять и архивировать неактуальный контент;
- оптимизировать большие файлы (картинки и аудио).

### **4.3 Чрезвычайные ситуации**

Чрезвычайные ситуации, которые могут возникнуть в процессе взаимодействия с веб-сервисом, могут иметь следующий характер:

- проблемы с безопасностью;
- проблемы с производительностью.

Никакие чрезвычайные ситуации не могут возникнуть сами по себе. Такие ситуации являются последствием плохого контроля за состоянием базы данных. Необходимо ввести плановое выявление неисправностей, чтобы избежать возникновения проблем с базами данных.

Самыми распространёнными уязвимостями веб-сайта считаются:

- некорректная настройка параметров безопасности;
- межсайтовое выполнение сценариев;
- недостатки аутентификации;
- недостатки контроля доступа;
- внедрение.

Помимо возможных неисправностей, связанных с базой данных, злоумышленники в своих корыстных целях могут попытаться совершить атаку на

активный веб-сервис. Такая атака может не только ограничить действие веб-сайта, но и привести к сливу конфиденциальных данных пользователей. Во избежание такой неприятности необходимо тщательно продумывать работу системы безопасности проекта. Никто не застрахован от подобных атак, поэтому в случае её наступления необходимо иметь конкретный план поведения, с помощью которого можно незамедлительно пресечь или минимизировать ущерб от атаки.

Для отражения каждого типа такой атаки необходим индивидуальный подход. Таким образом, чтобы защититься от межсайтового выполнения скриптов (XSS) необходимо экранировать любой пользовательский ввод. То есть, заменять специальные символы безопасными.

Достижение и последующее поддержание высокого уровня защищенности веб-приложения является сложной задачей, однако наибольшую эффективность принесет придерживание следующих правил:

- исправлять выявленные уязвимости как можно раньше;
- автоматизировать процессы, где это возможно.

Проблемы с производительностью могут выражаться в виде:

- некорректной загрузки веб-страниц;
- низкой скорости работы элементов интерфейса;
- долгой обработки запросов.

Для пресечения возникновения подобных проблем рекомендуется:

- при любой возможности откладывать загрузку контента: иногда пользователю не нужна вся информация на странице, а лишь некоторая. В подобной ситуации рекомендуется загружать ту часть страницы, которая нужна пользователю;

- использовать отдельные файлы языков программирования: отдельные файлы позволяют загружать их содержимое лишь единожды и потом по необходимости использовать. Это лучше, чем с каждым новым файлом на странице

загружать те же самые методы, что могли бы присутствовать в отдельном глобальном файле;

- использовать системы кэширования: при наличии системы кэширования, сайт должен будет только создать контент один раз вместо того, чтобы создавать его каждый раз, когда страницу посещают пользователи;

- не использовать изображения для показа текста;

- избегать изменения размеров изображений в HTML: рекомендуется использовать исходные размеры изображений. Таким образом веб-сайту не понадобится загружать стиль, который меняет изображение, и само изображение не потеряет в качестве;

- оптимизировать метод написания кода: вместо использования разнообразных тегов, представленных языком HTML, можно использовать стандартный тег «<div>» и менять в нём стили с помощью языка CSS. Так веб-сайту не понадобится нагружать страницу различными тегами, а лишь указать им стили.

## ЗАКЛЮЧЕНИЕ

Во время выполнения выпускной квалификационной работы были развиты навыки проектирования и создания веб-приложений, графических интерфейсов пользователя, были улучшены навыки работы с Visual Studio 2022, её настройки и владения её функционалом, с платформой ASP.NET, с языками программирования HTML, CSS и JavaScript.

В работе выполнены:

- анализ особенностей разработки и использования образовательных веб-сервисов;

- проектирование и разработка базы данных, включающая в себя следующие сущности: Пользователи, Задания, Результаты, Шаблон, Вопрос, Ответ;

- проектирование и разработка модулей хранения данных, а именно: модуль хранения данных пользователя, модуль хранения данных заданий, модуль хранения данных результатов и модуль хранения данных шаблонов;

- проектирование и разработка графического интерфейса пользователя с использованием языков программирования HTML, CSS и JavaScript, используя платформу создания веб-приложений ASP.NET в программно-технической среде Visual Studio 2022;

- функциональное тестирование проекта.

Разработанный программный продукт предназначен для создания и прохождения заданий на основе созданных шаблонов. Определен вектор дальнейшего развития данного веб-приложения.

Образовательный веб-сервис «ActiveTemplate» успешно прошёл государственную регистрацию в Реестре программ для ЭВМ (приложение Б).

В дальнейшем планируется развитие функциональности программного продукта в виде обновлений интерфейса, добавления новых шаблонов для создания заданий.

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- 1 Введение в ASP.NET MVC [Электронный ресурс]. – Режим доступа: <https://metanit.com/sharp/mvc/1.1.php>. – 18.02.2024.
- 2 Веб - сервис: что это? [Электронный ресурс]. – Режим доступа: <https://dzen.ru/a/X0kRz8c--yimIjbH>. – 18.02.2024.
- 3 Веб-сервисы – Характеристики [Электронный ресурс]. – Режим доступа: <https://coderlessons.com/tutorials/veb-razrabotka/izuchite-veb-servisy/veb-servisy-kharakteristiki>. – 18.02.2024.
- 4 Верстка сайта [Электронный ресурс]. – Режим доступа: <https://sendpulse.com/ru/support/glossary/website-layout>. – 20.02.2024.
- 5 Восемь сервисов, которые помогут создать интерактивный и вовлекающий онлайн-курс [Электронный ресурс]. – Режим доступа: <https://skillbox.ru/media/education/9-servisov-kotorye-pomogut-sozdat-interaktivnyy-i-vovleayushchiy-onlaynkurs/>. – 18.02.2024.
- 6 ГОСТ Р 50949-2001. Средства отображения информации индивидуального пользования. Методы измерений и оценки эргономических параметров и параметров безопасности. М., 2002. 27 с.
- 7 ГОСТ Р 50948-2001. Средства отображения информации индивидуального пользования. Общие эргономические требования и требования безопасности. М., 2002. 13 с.
- 8 ГОСТ Р ИСО 9241-161-2016. Эргономика взаимодействия человек-система. Часть 161. Элементы графического пользовательского интерфейса. М., 2016. 58 с.
- 9 ГОСТ Р 56274-2014. Общие показатели и требования в эргономике. М., 2015. 31 с.
- 10 Круг, С. Не заставляйте меня думать / С. Круг. – М.: Эксмо, 2021. – 256 с.

11 Никсон, Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5 / Р. Никсон. – М.: Питер, 2023. – 832 с.

12 Новиков, Б. Основы технологий баз данных: учебное пособие / Б. Новиков. – М.: ДМК Пресс, 2020. – 582 с.

13 Разработка онлайн-курсов: 7 сервисов для создания интерактивных заданий [Электронный ресурс]. – Режим доступа: <https://netology.ru/blog/06-2022-interactive-exercises-services>. – 18.02.2024.

14 Робин, Н. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5 / Н. Робин – М.: Питер, 2022. – 832 с.

15 Сайт на ASP.NET для начинающих — первые шаги [Электронный ресурс]. – Режим доступа: <https://tproger.ru/articles/web-asp-net-for-beginners>. – 20.02.2024.

16 Сидоренко, И. Дизайнер интерфейсов / И. Сидоренко. – М.: Олимп-Бизнес, 2019. – 255 с.

17 Фрэйн, Б. Отзывчивый дизайн на HTML5 и CSS3 для любых устройств / Б. Фрэйн. – М.: Питер, 2021. – 336 с.

18 Что такое веб-дизайн [Электронный ресурс]. – Режим доступа: <https://brenda.ru/page6073910.html>. – 25.02.2024.

19 Что такое веб-сервисы и как они используются [Электронный ресурс]. – Режим доступа: <https://www.purrweb.com/ru/blog/chto-takoe-veb-servisy-i-kak-oni-ispolzuyutsya/>. – 20.02.2024.

20 Шилдс, У. SQL: быстрое погружение / У. Шилдс. – М.: Питер, 2022. – 224 с.

21 Шпикерманн, Э. О шрифте / Э. Шпикерманн. – М.: Манн, Иванов и Фербер, 2020. – 208 с.

22 Экологичный веб-дизайн [Электронный ресурс]. – Режим доступа: <https://www.frontender.info/sustainable-web-design/>. – 20.04.2024.

23 ASP.NET [Электронный ресурс]. – Режим доступа: <https://blog.skillfactory.ru/glossary/asp-net/>. – 15.04.2024.

## ПРИЛОЖЕНИЕ А

### Техническое задание на разработку

#### 1 ВВЕДЕНИЕ

##### **1.1 Полное наименование программы**

Полное наименование разрабатываемой программы – Образовательный веб-сервис «ActiveTemplate».

##### **1.2 Краткая характеристика области применения программы**

Программа предназначена для проверки знаний пользователя посредством получения результатов прохождения заданий. Задания составляются на основе существующих шаблонов, реализованных внутри веб-сервиса, результаты заданий сохраняются в личном кабинете пользователя. Каждое задание уникально и создаётся преподавателем для последующей его рассылки обучающимся. Обучающиеся, в свою очередь, проходят данное задание и результаты о его прохождении записываются в базу данных.

##### **1.3 Краткая характеристика объекта, в котором используют программу**

Объектом разработки программы является высшее учебное заведение Амурский государственный университет, предоставляющее образовательные услуги.

#### 2 ОСНОВАНИЯ ДЛЯ РАЗРАБОТКИ

Основанием для разработки является Договор №890-уч от 03.04.2024 Договор утвержден институтом компьютерных и инженерных наук Амурского государственного Университета, именуемым в дальнейшем Заказчиком, и Марковым Никитой Евгеньевичем (студент), именуемым в дальнейшем исполнителем.

Наименование темы: разработка модулей хранения и интерфейса образовательного web-сервиса ActiveTemplate.

## Продолжение ПРИЛОЖЕНИЯ А

Результатом работы является образовательный веб-сервис и пояснительная записка, оформленная по стандарту АмГУ в рамках дипломного проекта.

### 3 НАЗНАЧЕНИЕ РАЗРАБОТКИ

Программное изделие будет использоваться в дистанционном формате без подразделения пользователей на группы.

#### 3.1 Функциональное назначение

С помощью программного изделия пользователь сможет создавать на основе шаблонов и проходить уже созданные задания в целях обучения и саморазвития. Результаты пройденных заданий будут записываться в личный кабинет пользователя.

#### 3.2 Эксплуатационное назначение

Программное изделие должно эксплуатироваться на персональном компьютере или мобильном устройстве пользователя.

### 4 ТРЕБОВАНИЯ К ПРОГРАММЕ ИЛИ ПРОГРАММНОМУ ИЗДЕЛИЮ

#### 4.1 Требования к функциональным характеристикам

##### 4.1.1 Требования к составу выполняемых функций

Программное изделие при запуске, посредством обращения к сайту, ожидает ввода логина и пароля пользователя. Доступна регистрация новых пользователей с подтверждением аккаунта через электронную почту.

После успешной аутентификации пользователь может перейти по вкладке к созданию заданий или прохождению существующих.

При переходе на вкладку создания заданий, пользователя встречает форма создания, которую необходимо заполнить для успешного конструирования задания. Здесь представляется возможность выбора шаблона задания, после чего, в соответствии с выбранным шаблоном, необходимо заполнить соответствующие поля формы, после чего подтвердить свой выбор. Задание становится доступно для каждого аутентифицированного пользователя. Также

## Продолжение ПРИЛОЖЕНИЯ А

доступен текст, который можно вложить в код страницы с целью интегрирования данного задания в верстку внешнего сайта.

При переходе на вкладку прохождения заданий, страница показывает популярные созданные задания, которые можно выбрать для прохождения.

В случае успешного прохождения задания, результаты записываются в личный кабинет пользователя.

Программный продукт должен обеспечивать возможность выполнения следующих функций:

- создание аккаунта и личный кабинет пользователя;
- генерация упражнения по заданным параметрам;
- функция обеспечивающая возможность менять фон у макета задания;
- функция автоматической генерации полей упражнения;
- генерация ссылки на упражнение после его создания;
- генерация ссылки на возможность встраивания упражнения на другие веб-ресурсы;
- функция поиска заданий по критериям;
- функция отслеживания результатов.

### **4.1.2 Требования к организации входных и выходных данных**

Данные о персональных данных пользователей хранятся в базе данных. Ввод данных в базу выполняет администратор, выполняющий поставку программного обеспечения, с помощью запросов к СУБД через встроенный интерфейс.

### **4.1.3 Требования к временным характеристикам**

После изменения администратором данных, находящийся в базе данных, новая информация на клиентах отображается не позднее, чем через 5 минут.

### **4.2 Требования к надежности**

## Продолжение ПРИЛОЖЕНИЯ А

Вероятность безотказной работы системы должна составлять не менее 99.99% при условии исправности сети Интернет.

### **4.2.1 Требования к обеспечению надежного (устойчивого) функционирования программы**

Надежное функционирование программы должно быть обеспечено выполнением заказчиком совокупности организационно-технических мероприятий, перечень которых приведен ниже:

- организацией бесперебойного питания технических средств;
- использованием лицензионного программного обеспечения;
- регулярным выполнением рекомендаций Министерства труда и социального развития РФ, изложенных в Постановлении от 23 июля 1998 г. «Об утверждении межотраслевых типовых норм времени на работы по сервисному обслуживанию ПЭВМ и оргтехники и сопровождению программных средств»;
- регулярным выполнением требований ГОСТ 51188-98. Защита информации. Испытания программных средств на наличие компьютерных вирусов.

### **4.2.2 Время восстановления после отказа**

Время восстановления после отказа, вызванного сбоем электропитания технических средств (иными внешними факторами), не фатальным сбоем (не крахом) операционной системы, не должно превышать 10 минут при условии соблюдения условий эксплуатации технических и программных средств.

### **4.2.3 Отказы из-за некорректных действий оператора**

Отказы программы возможны вследствие некорректных действий оператора (пользователя) при взаимодействии с операционной системой. Во избежание возникновения отказов программы по указанной выше причине следует обеспечить работу пользователя без предоставления ему административных привилегий.

## Продолжение ПРИЛОЖЕНИЯ А

### **4.3 Условия эксплуатации**

Программное изделие работает в сети Интернет и запускается посредством открытия ссылки на одноименный сайт. Должна существовать устойчивая связь по сети между пользователем и сайтом.

Веб-сервис должен обладать простым в понимании дизайном, иметь высокую производительность и быстродействие.

#### **4.3.1 Климатические условия эксплуатации**

Специальные условия не требуются.

#### **4.3.2 Требования к видам обслуживания**

Программа не требует проведения каких-либо видов обслуживания.

#### **4.3.3 Требования к численности и квалификации персонала**

При установке и настройке системы необходим системный администратор. В процессе эксплуатации с программой работают пользователи.

Системный администратор должен иметь высшее профильное образование и сертификаты компании-производителя операционной системы. В перечень задач, выполняемых системным администратором, должны входить:

- установка клиентских приложений;
- настройка СУБД;
- настройка сети между клиентами и СУБД.
- Пользователь программы (оператор) должен обладать практическими навыками работы с графическим пользовательским интерфейсом операционной системы.

К квалификации сотрудника специальные требования не предъявляются.

### **4.4 Требования к составу и параметрам технических средств**

Состав технических средств:

Компьютер пользователя, включающий в себя:

## Продолжение ПРИЛОЖЕНИЯ А

- процессор x86 с тактовой частотой, не менее 1 ГГц;
- оперативную память объемом, не менее 1 Гб;
- подключение к интернету со скоростью, не менее 5 Мбит/с;
- видеокарту, монитор, мышь, клавиатура.

### **4.5 Требования к информационной и программной совместимости**

Приложение пользователя обмениваются с СУБД сообщениями по Интернет-сети. Должна быть исключена высокая нагрузка сети.

Исходные коды программы должны быть написаны на языках программирования и разметки html, css, javascript, c# с использованием технологии asp.net core. В качестве интегрированной среды разработки должна быть использована среда Microsoft Visual Studio 2022 и новее.

### **4.6 Требование к маркировке и упаковке**

Специальных требований к маркировке не предъявляется.

### **4.7 Требования к транспортированию и хранению**

Специальных требований не предъявляется.

### **4.8 Специальные требования**

Программа должна обеспечивать взаимодействие с пользователем посредством графического пользовательского интерфейса, разработанного согласно рекомендациям компании-производителя операционной системы.

## **5 ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ**

Предварительный состав программной документации:

- техническое задание (включает описание применения);
- руководство системного программиста;
- руководство оператора;
- ведомость эксплуатационных документов;
- формуляр.

## Продолжение ПРИЛОЖЕНИЯ А

### 6 ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ

Данное программное изделие обладает функциональностью, которая частично совпадает с функциональностью аналогов. Однако, за счёт дополнительных функций, таких как, возможность внедрения в верстку сайта, разнообразие представленных шаблонов и учёт результатов заданий в личном кабинете пользователя программное изделие является единственным в своем роде веб-сервисом, предоставляющим подобные услуги. Работа данного веб-сервиса может быть распространена не только на объект, для которого он делался, но и на подобные ему учреждения. Экономический эффект при этом может быть обеспечен за счет платной установки системы.

### 7 СТАДИИ И ЭТАПЫ РАЗРАБОТКИ

Разработка должна быть проведена в три стадии:

- техническое задание;
- технический (и рабочий) проекты;
- защита проекта комиссии.

На стадии «Техническое задание» должен быть выполнен этап разработки, согласования и утверждения настоящего технического задания.

На стадии «Технический (и рабочий) проект» должны быть выполнены перечисленные ниже этапы работ:

- разработка программы;
- разработка программной документации;
- испытания программы.

На стадии «Защита проекта комиссии» необходимо подготовить выступление с речью для защиты перед комиссией.

На этапе разработки технического задания должны быть выполнены перечисленные ниже работы:

## Продолжение ПРИЛОЖЕНИЯ А

- постановка задачи;
- определение и уточнение требований к техническим средствам;
- определение требований к программе;
- определение стадий, этапов и сроков разработки программы и документации на нее;
- согласование и утверждение технического задания.

На этапе разработки программы должна быть выполнена работа по программированию (кодированию) и отладке программы.

На этапе разработки программной документации должна быть выполнена разработка программных документов в соответствии с требованиями ГОСТ 19.101-77.

На этапе испытаний программы должны быть выполнены перечисленные ниже виды работ:

- разработка, согласование и утверждение порядка и методики испытаний;
- проведение приемо-сдаточных испытаний;
- корректировка программы и программной документации по результатам испытаний.

На этапе подготовки к выступлению перед защитой должны быть подготовлены речь и презентация для выступления.

### 8 ПОРЯДОК КОНТРОЛЯ И ПРИЕМКИ

При приемке программного изделия заказчик должен ознакомиться с документацией и руководством оператора. Примерка промежуточных и окончательных работ должна проводиться с участием тех лиц, которые будут работать с данной системой.

## Продолжение ПРИЛОЖЕНИЯ А

Заказчик должен проверить систему на соответствие предъявляемым требованиям. На основании протокола испытаний исполнитель с заказчиком подписывают акт приемки-сдачи программы в эксплуатацию.

# ПРИЛОЖЕНИЕ Б

**РОССИЙСКАЯ ФЕДЕРАЦИЯ**



**СВИДЕТЕЛЬСТВО**  
о государственной регистрации программы для ЭВМ  
**№ 2024661102**

**Образовательный веб-сервис «ActiveTemplate»**

Правообладатель: *Федеральное государственное бюджетное образовательное учреждение высшего образования "Амурский государственный университет" (RU)*

Авторы: *Галаган Татьяна Алексеевна (RU), Марков Никита Евгеньевич (RU), Яцун Евгений Олегович (RU)*

Заявка № **2024619923**  
Дата поступления **07 мая 2024 г.**  
Дата государственной регистрации  
в Реестре программ для ЭВМ **15 мая 2024 г.**

*Руководитель Федеральной службы  
по интеллектуальной собственности*



ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ  
Сертификат: 429b660fe3853164ba196183673b4aa7  
Владелец: **Зубов Юрий Сергеевич**  
Действителен с 18.08.2023 по 02.08.2024

*Ю.С. Зубов*

Рисунок А.1 – Свидетельство о регистрации программы для ЭВМ