МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Забайкальский государственный университет»

(ФГБОУ ВО «ЗабГУ»)

Энергетический факультет

Кафедра информатики, вычислительной техники и прикладной математики

КУРСОВОЙ ПРОЕКТ

по дисциплине: Технология разработки программного обеспечения

на тему «Онлайн-редактор мультипликации на основе HTML5»

Выполнил ст. гр. ИВТ-20

Банковский А.C.

Проверил доцент кафедры ИВТ и ПМ,

к.т.н., доцент Валова О.В.

Чита

2024

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение

высшего образования

«Забайкальский государственный университет»

(ФГБОУ ВО «ЗабГУ»)

Энергетический факультет

Кафедра информатики, вычислительной техники и прикладной математики

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проекту

по направлению подготовки 09.03.01 Информатика и вычислительная техника

на тему: Онлайн-редактор мультипликации на основе HTML5

Выполнил студент группы ИВТ-20 Банковский Александр Сергеевич

Руководитель работы: доцент кафедры ИВТ и ПМ, к.т.н., доцент Валова Ольга Валерьевна

Чита

2024

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Забайкальский государственный университет»

(ФГБОУ ВО «ЗабГУ»)

Энергетический факультет

Кафедра информатики, вычислительной техники и прикладной математики

ЗАДАНИЕ

на курсовой проект

По дисциплине: Технология разработки программного обеспечения

Студенту: Банковскому Александру Сергеевичу

по направлению подготовки: 09.03.01 Информатика и вычислительная техника

1. Тема курсового проекта: «Онлайн-редактор мультипликации на основе HTML5»

2. Срок подачи студентами законченной работы: 4.04.2024

3. Исходные данные к работе: согласно техническому заданию;

4. Перечень подлежащих разработке в курсовом проекте вопросов:

1. Постановка и анализ задачи;
2. Анализ задачи;
3. Программная реализация;
4. Техническое задание;
5. Руководство пользователя.

5. Перечень графического материала: –

Дата выдачи задания: 06.02.2023

Руководитель курсового проекта\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Валова О.В./

(подпись, расшифровка подписи)

Задание приняли к исполнению

«6» февраля 2024 г.

Подпись студента\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /Банковский А.С.

РЕФЕРАТ

Пояснительная записка – 31 с, 21 рис., 1 табл., 4 источника.

АНИМАЦИЯ, ВЕБ-АНИМАЦИЯ, РЕДАКТОР АНИМАЦИИ, ВЕБ-ПРИЛОЖЕНИЕ, ЯЗЫК C SHARP, BLAZOR, HTML, CSS.

Редактор мультипликации на основе HTML5 – это инструмент, предоставляющий пользователям все базовые возможности для создания анимации:

* добавление и удаление анимируемых объектов, объектов, участвующих в анимации;
* редактирование свойств объектов;
* создание анимации с помощью изменения свойств объектов на протяжении времени;

Данный инструмент нацелен на облегчение процесса создания анимации на основе HTML5.

В данной работе подробно описан процесс разработки онлайн-редактора анимации на основе HMTL5 с использованием технологий C#, Blazor, HTML5, CSS3, JavaScript, WebAssembly и PWA.

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 6](#_Toc164501025)

[1 Постановка и анализ задачи. 7](#_Toc164501026)

[1.1 Описание предметной области 7](#_Toc164501027)

[1.2 Диаграмма Вариантов использования 8](#_Toc164501028)

[1.3 Обоснование выбора средств реализации 10](#_Toc164501029)

[2 Описание данных 13](#_Toc164501030)

[3 Программная реализация 14](#_Toc164501031)

[3.1 Архитектура приложения 14](#_Toc164501032)

[3.2 Реализация клиентской части 15](#_Toc164501033)

[3.3 Реализация серверной части 16](#_Toc164501034)

[3.4 Описание алгоритмов 18](#_Toc164501038)

[4 Техническое задание 21](#_Toc164501039)

[4.1 Введение 21](#_Toc164501040)

[4.2 Назначение разработки 21](#_Toc164501041)

[4.3 Требования к функциональным характеристикам 21](#_Toc164501042)

[4.4 Требования к составу и параметрам технических средств 22](#_Toc164501043)

[4.5 Требования к информационной, программной совместимости 23](#_Toc164501044)

[4.6 Требования к программной документации 23](#_Toc164501045)

[5 Руководство пользователя 24](#_Toc164501046)

[ЗАКЛЮЧЕНИЕ 28](#_Toc164501047)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 29](#_Toc164501048)

ВВЕДЕНИЕ

Мультипликация или анимация - технология иллюзии движущейся картинки, путём смены статичных изображений [1].

Анимация стала неотъемлемой частью каждого пользователя не только интернета, но и компьютера, смартфона и телевидения. Анимация окружает каждого из нас: в рекламных баннерах, элементах графического пользовательского интерфейса, графических элементов видеоматериалов и т.п.

Веб-приложение является наиболее распространённым и популярным типом приложений. В веб-приложениях используется HTML-анимация, которую можно реализовать посредствами CSS технологий [2] или с использованием JavaScript [3]. В связи с широким применением веб-приложений, появился большой спрос на редакторы HTML-анимации.

В реализуемом программном средстве используются современный технологии веб-приложений: Blazor, WebAssembly и PWA.

Целью данной работы является разработка инструмента для создания и редактирования HTML-анимации в виде веб-приложения, способного упростить и оптимизировать работу с HTML-анимацией.

# Постановка и анализ задачи.

## Описание предметной области

Реализованное приложение представляет из себя редактор анимации на основе HTML5, CSS3 анимации и JavaScript анимации. Приложение предоставляет следующие возможности:

1. добавление и удаление анимируемых объектов (объектов, участвующих в анимации);
2. изменение свойств анимируемых объектов, таких как позиция на холсте, размер, цвет, прозрачность и другие;
3. создание анимации, путём изменения свойств анимируемых объектов с течением времени;
4. предпросмотр редактируемой анимации;

Особенностями реализуемого программного средства являются нацеленность на создание HTML5 мультипликации и возможность создания интерактивной анимации, вида анимации, которая позволяет зрителю участвовать в процессе анимации, зачастую путём использования наиболее распространённых устройств ввода, таких как клавиатура и мышь.

Разрабатываемое приложение предназначено для использования художниками и любителями, работающими с анимацией, разработчиками, желающих разнообразить своё веб-приложение с помощью анимации, а также остальным желающим.

Среди аналогов реализуемого программного средства можно выделить 3 типа аналогов: конструкторы сайтов, предоставляющих функционал создания веб-сайтов, включая инструменты для работы с анимацией.

Первый тип аналогов представляет из себя профессиональные инструменты для работы с анимацией любой сложности. К таким приложениям относятся Adobe Animate, Tumult Hype, Hippani, Radi и др.

Второй тип аналогов представляет из себя программные средства для создания рекламных баннеров, блоков, видео. К данному типу аналогов относятся следующие приложения: Onlymega, Google Web Designer и др.

Третий тип аналогов представляет из себя конструкторы веб-сайтов, предоставляющих обширный функционал создания веб-сайтов. К таким конструкторам относятся: Wix, Tobiz, Tilda, Рег.Ру и др.

Большинство представителей всех трёх типов аналогов обладают следующими особенностями:

1. возможность создавать интерактивную анимацию;
2. наличие инструментов для работы с 3D анимацией;
3. наличие магазина шаблонов, для быстрого старта проекта;

Приложения первых двух типов аналогов также предоставляют возможность экспорта проекта в популярные видео-форматы: mp4, gif и другие.

## 1.2 Диаграмма Вариантов использования

Ниже представлены рисунки, представляющие собой UML-диаграммы вариантов использования разрабатываемого приложения.

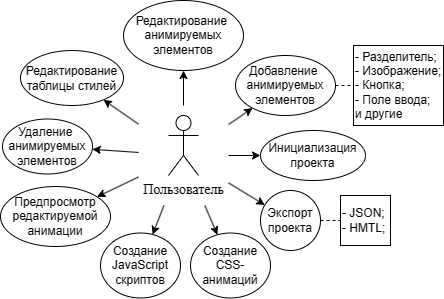


Рисунок 1- Главная UML-диаграмма вариантов использования

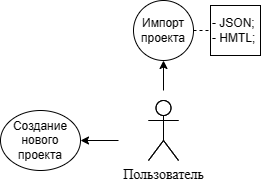


Рисунок 2 - UML-диаграмма вариантов использования инициализации проекта

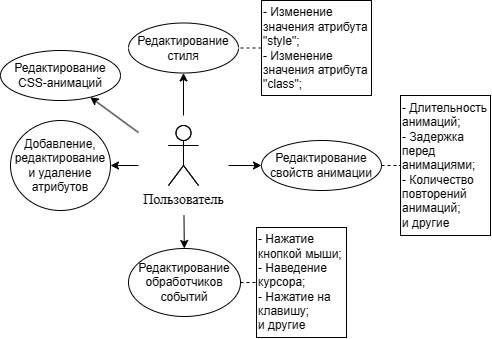


Рисунок 3 - UML-диаграмма вариантов использования редактирования анимируемых элементов

Реализуемое приложение предоставляет пользователю следующие возможности:

1. экспорт проекта в форматах JSON или HTML;
2. инициализация проекта;
   * импорт проекта из форматов JSON или HTML;
   * создание нового проекта;
3. добавление анимируемых элементов. Таких как: разделитель; изображение; кнопка; поле ввода и др;
4. удаление анимируемых элементов;
5. редактирование анимируемых объектов;
   * добавление, редактирование и удаление атрибутов;
   * редактирование CSS-анимаций;
   * редактирование стиля с помощью изменения значений атрибутов «style» и «class»;
   * редактирование свойств анимации. Таких как: длительность анимаций, задержка перед анимациями, количество повторений анимаций и др;
   * редактирование обработчиков событий. Таких как: нажатие кнопкой мыши на элемент, наведение курсора на элемент, нажатие на клавишу и др;
6. редактирование таблицы стилей;
7. создание и редактирование CSS-анимаций;
8. создание JavaScript скриптов;
9. предпросмотр редактируемой анимации;

## 1.3 Обоснование выбора средств реализации

В качестве языка программирования был выбран язык C# (C Sharp) [4]. C# - объектно-ориентированный язык программирования общего назначения. C Sharp является языком разработки приложения на платформе «Microsoft .NET framework» и «.NET Core». Язык имеет статическую типизацию.

Платформа .NET является кроссплатформенной, что означает, что приложение, разработанное на C#, может быть запущено на различных операционных системах без изменений исходного кода.

Также C Sharp имеет огромное сообщество разработчиков, что обеспечивает доступ к множеству ресурсов, библиотек и фреймворков, что может значительно упростить разработку проекта.

Язык C# позволяет обеспечить хорошую производительность реализуемого приложения, что делает приложение более комфортным в использовании.

Язык С# поддерживает полиморфизм, перегрузку операторов, делегаты, атрибуты, события, свойства, обобщённые типы и методы, итераторы, анонимные функции с поддержкой замыкания и другие механизмы, которые выделяют данный язык среди других языков, как мощный инструмент для создания разного рода приложений.

В качестве фреймворка для создания приложения была выбрана бесплатная веб-платформа с открытым исходным кодом - Blazor [5]. Blazor основан на мощной и гибкой компонентной модели для создания многофункционального интерактивного веб-интерфейса. Он позволяет разрабатывать веб-приложения с использованием C# и HTML, давая возможность разрабатывать как пользовательский интерфейс (далее «UI»), так и бизнес-логике приложения, не применяя других фреймворком или библиотек. Приложение Blazor позволяет реализовывать гибкий интерфейс, благодаря razor-компонентам - компонентам UI с собственной внутренней логикой. Такие компоненты позволяют реализовать UI-компонент единожды и использовать его по многу раз, обращаясь к нему, как к HMTL-элементу.

Для большей производительности разрабатываемое веб-приложение использует технологию WebAssembly (дальше «wasm») [6]. Wasm - язык программирования низкого уровня, исполняемый на стековой виртуальной машине, которая может быть запущена в среде браузера. Код на wasm - переносимое абстрактное синтаксическое дерево, что обеспечивает как более быстрый анализ, так и более эффективное выполнение в сравнении с JavaScript.

В связи с тем, что реализуемое приложение является одностраничным приложением и для повышения производительности, приложение является прогрессивным веб-приложением (Progressive Web Application - PWA) [7]. PWA - это технология в веб-разработке, которая визуально и функционально трансформирует сайт в приложение. PWA позволяет открыть приложение с помощью мобильного браузера и сохраняет функциональность нативного приложения [8]:

* отправка push-уведомлений;
* работа в режиме офлайн;
* доступ к аппаратному обеспечению устройства (с ограничениями);

возможность установить веб-приложение, как нативное;

В качестве среды разработки был выбран бесплатный кроссплатформенный мощный и легковесный текстовый редактор Visual Studio Code (далее «VS Code») [9]. Он включает в себя отладчик, инструменты для работы с системой контроля версий, подсветку синтаксиса, технологию автодополнения и автозавершения кода, инструменты для отладки и средства для рефакторинга.

VS Code имеет широкие возможности для кастомизации, такие как: пользовательские темы, настройка сочетания клавиш и файлы конфигурации. Благодаря технологии расширений, VS Code можно настроить под любые нужды, адаптировав разработку под конкретный проект, установив необходимые расширения, которые дополнят данную среду разработки новыми средствами отладки и редактирования кода на определённом языке программирования и не только.

# Описание данных

Реализуемый редактор анимации использует различные входные, промежуточные выходные данные.

К входным данным относятся все данные, поступающие со стороны пользователя, а именно вводимые значения свойств анимируемых элементов или проект в качестве импортируемого файла.

К промежуточным данным относятся все данные, которые реализуемое программное средство использует в ходе выполнения своей работы. Такие данные как внутреннее состояние анимируемых объектов, которые конвертируются в JSON или HTML файл при экспорте проекта.

К выходным данным относятся все данные, отображаемые данные анимируемых объектов на элементах графического интерфейса пользователя, а также проект в виде экспортируемого файла.

# 3 Программная реализация

## 3.1 Архитектура приложения

Для разработки приложения использовалась микросервисная архитектура. Приложение состоит из следующих сервисов (рис. 7), которые расположены на сервере и компьютере пользователя соответственно (рис. 8):

* клиентское приложение;
* ядро – сервис для хранения данных приложения и управления ими, а также сервис для автоматизированной системы.

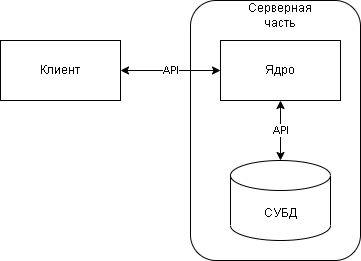


Рисунок 7 - Архитектура веб-приложения

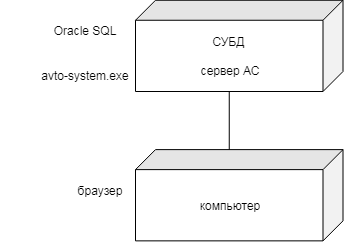


Рисунок 8 - Диаграмма размещения

Клиентское приложение – это программа, запускаемая в браузере пользователя, которая позволяет текстовым сочинителям получать доступ к веб-сервису, использовать его функции и взаимодействие с другими функциями.

Ядро – серверное приложение, которое управляет данными и функциональными возможностями сервиса. Ядро будет получать запросы от клиентского приложения, обрабатывать запрос и возвращать ответы клиенту.

СУБД (система управления базами данных) - это программная система, которая позволяет управляют структурой баз данных и контролировать организованные данные, хранящиеся в них. База данных [2] – это набор организованных данных, хранимых в соответствии со схемой данных, манипулирование которыми выполняют в соответствии с правилами средств моделирования.

## 3.2 Реализация клиентской части

Клиентская сторона является важнейшим компонентом веб-разработки, который играет ключевую роль в обеспечении позитивного пользовательского опыта. Это часть веб-приложения, которая запускается на устройстве пользователя и взаимодействует с сервером для обеспечения необходимой функциональности.

HTML-шаблоны используются для создания базовой структуры веб-страницы. Для работы используется Jinja2 [3] в качестве движка шаблонов, который позволяет создавать динамические HTML-файлы. Они изменяются на основе входных данных пользователя или внутренних переменных веб-сервиса. Как только шаблоны сгенерированы, они передаются браузеру клиента в качестве ответа на его запросы.

Таблицы стилей CSS используются для повышения визуальной привлекательности HTML-шаблонов. Они позволяют настраивать шрифты, цвета и макет веб-страницы, делая ее более привлекательной для пользователей.

JavaScript используется для выполнения асинхронных запросов к серверу, что позволяет веб-приложению обновлять данные, не прерывая работу пользователя. С помощью JavaScript создаются динамические формы, которые улучшают пользовательский интерфейс и делают веб-приложение более привлекательным.

Таким образом, реализация на стороне клиента основана на HTML, CSS и JavaScript. HTML генерируется динамически с помощью движка шаблонов Flask, а CSS используется для разработки HTML-шаблонов. JavaScript используется для выполнения асинхронных запросов к серверу и динамического обновления пользовательского интерфейса.

## 3.3 Реализация серверной части

Под серверной частью приложения понимается набор сервисов, с помощью которых реализована логика работы приложения. Серверная часть этого проекта построена с использованием Python и веб-фреймворка Flask. Flask – это легкий веб-фреймворк, который предоставляет мощный API для обработки HTTP-запросов и ответов. Диаграмма компонентов представлена на рисунке 9.

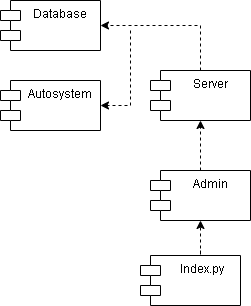


Рисунок 9 - Диаграмма компонентов

Ядро выполняет следующие задачи:

* общение с клиентской частью через API;
* аутентификация и авторизация пользователей;
* ограничение доступа к панели администратора на основе статуса аутентификации пользователя и его роли;
* обработка запросов от клиента, что включает проверку пользовательских данных;
* кэширование часто используемых данных;
* оптимизация запросов к базе данных;
* использование алгоритмов для поддержки деятельности пользователя и администратора.

Для доступа к данным ядро производит проверку авторизации. Авторизация производится на основе расширения Flask-Login. Функция login\_user() используется для входа пользователя в систему после аутентификации пользователя. Эта функция устанавливает сессионный файл cookie с идентификатором пользователя.

JavaScript [4] используется для обеспечения связи между клиентской и серверной частями вашего веб-приложения. Клиентский JavaScript-код отправляет запросы серверному коду асинхронно. Затем серверный код обрабатывает запрос и отправляет ответ обратно в клиентский JavaScript-код.

Скорость, с которой происходит этот обмен, может варьироваться в зависимости от различных факторов, таких как скорость интернет-соединения клиента, скорость обработки сервером и объем передаваемых данных. Однако использование AJAX может помочь повысить воспринимаемую скорость работы приложения, позволяя пользователю взаимодействовать с приложением без необходимости ждать полной перезагрузки страницы, что улучшает пользовательский опыт.

Автоматизированная система является частью ядра и предоставляет различные функции. Система отслеживает поведение пользователя, анализируя взаимодействие пользователя с веб-сервисом, например, тип контента, к которому он обращается, сколько времени он проводит на определенных страницах, данные о потоке кликов, отслеживает рассылку спама и идентифицирует пользователей, которые часто генерируют большое количество сообщений о нарушении.



## Описание алгоритмов

Код структурирован с использованием принципов объектно-ориентированного программирования, при этом каждая модель (пользователь, профиль и другие) представлена классом. Классы инкапсулируют данные и поведение моделей и обеспечивают чистый и модульный интерфейс для взаимодействия с базой данных.

Алгоритм фильтрации (рис. 10) используется для рекомендации пользователям публикации на основе их интересов и действий на платформе. В центре рекомендательной системы находится так называемая матрица предпочтений. Это матрица, по одной из осей которой отложены все клиенты сервиса (Users), а по другой – объекты рекомендации (Items). На пересечении некоторых пар (user, item) данная матрица заполнена оценками (Ratings) – это известный нам показатель заинтересованности пользователя в данном контенте, выраженный по заданной шкале (например, от 1 до 5).



Рисунок 10 - Матрица предпочтений

Алгоритм скрытого распределения Дирихле (рис. 11) предлагает теги для сообщений на основе содержания сообщения. С помощью данного алгоритма происходит группировка коллекций тем, чтобы найти более похожие темы и поместить менее похожие темы в разные категории, используя теги. Когда новая тема появляется в наборе данных, есть возможность автоматически помещать её в правильную категорию.

countVectorizer = CountVectorizer(stop\_words='russian')  
 termFrequency = countVectorizer.fit\_transform(docs)  
 featureNames = countVectorizer.get\_feature\_names()  
lda = LatentDirichletAllocation(n\_components=3)  
 lda.fit(termFrequency)  
 for idx, topic in enumerate(lda.components\_):  
 print ("Topic ", idx, " ".join(featureNames[i] for i in topic.argsort()[:-10 - 1:-1]))

Рисунок 11 – Алгоритм скрытого распределения Дирихле

Наивный байесовский алгоритм (рис. 12) используется для обнаружения неподходящих сообщений и комментариев. Алгоритм использует словарь слов, вычисляет вероятность для каждого слова в тексте и фильтрует слова, вероятность которых меньше пороговой. Слова с вероятностью меньше пороговой вероятности являются нерелевантными.

df['message'] = df['message'].apply(lambda x: ' '.join(x))

count\_vect = CountVectorizer()

counts = count\_vect.fit\_transform(df['message'])

transformer = TfidfTransformer().fit(counts)

counts = transformer.transform(counts)

X\_train, X\_test, y\_train, y\_test = train\_test\_split(counts, df['label'], test\_size=0.1, random\_state=69)

model = MultinomialNB().fit(X\_train, y\_train)

Рисунок 12 – Наивный байесовский алгоритм

Алгоритм нечеткого сопоставления (рис. 13). Модуль Fuzzy используется для выполнения нечеткого сопоставления строк. Алгоритм состоит в создании таблицы расстояний между всеми префиксами двух сравниваемых строк, а затем использует эту таблицу для вычисления конечного расстояния. Пороговое значение задано таким образом, чтобы каждое соответствие имело определенное качество, и составляет 70.

def process\_str\_for\_similarity\_cmp(input\_str, normalized=False, ignore\_list=[]):

for ignore\_str in ignore\_list:

input\_str = re.sub(r'{0}'.format(ignore\_str), "", input\_str, flags=re.IGNORECASE)

if normalized is True:

input\_str = input\_str.strip().lower()

#clean special chars and extra whitespace

input\_str = re.sub("\W", "", input\_str).strip()

return input\_str

Рисунок 13 – Алгоритм нечеткого сопоставления

В заключение следует отметить, что код хорошо структурирован и использует принципы объектно-ориентированного программирования для представления моделей в виде классов, инкапсулируя данные и поведение и обеспечивая чистый интерфейс для взаимодействия с базой данных. В системе используется несколько алгоритмов, включая быструю сортировку для сортировки, нечеткое сопоставление для сравнения строк, двоичный поиск для поиска совпадений в списках и фильтрацию для целей рекомендаций на основе матрицы предпочтений. Алгоритм скрытого распределения Дирихле используется для группировки тем, в то время как наивный байесовский алгоритм используется для обнаружения неподходящих сообщений и комментариев. Система представляется всеобъемлющей и эффективной в решении различных функциональных задач, которые она предоставляет.

# 4 Техническое задание

## 4.1 Введение

Приложение "Онлайн редактор мультипликации на основе HTML5" предназначено для использования художниками, нацеленных на веб-разработку, и создателям интерактивной мультипликации. Оно также может быть порекомендовано всем желающим, кто хочет украсить веб-приложение анимационными вставками либо создавать анимацию на основе HTML5.

## 4.2 Назначение разработки

Программа "Редактор мультипликации" должна предоставлять базовые инструменты для редактирования анимации:

1. добавление и удаление анимируемых объектов;
2. редактирование свойств анимируемых объектов;
3. создание анимации за счёт изменения свойств анимируемых объектов с течением времени;

## 4.3 Требования к функциональным характеристикам

Онлайн редактор должен быть оснащён обширным спектром разных функций и инструментов для удовлетворения нужд и требований пользователей.

Реализуемое приложение предоставляет пользователю следующие возможности:

1. экспорт проекта в форматах JSON или HTML;
2. инициализация проекта;
   * импорт проекта из форматов JSON или HTML;
   * создание нового проекта;
3. добавление анимируемых элементов. Таких как: разделитель; изображение; кнопка; поле ввода и др;
4. удаление анимируемых элементов;
5. редактирование анимируемых объектов;
   * добавление, редактирование и удаление атрибутов;
   * редактирование CSS-анимаций;
   * редактирование стиля с помощью изменения значений атрибутов «style» и «class»;
   * редактирование свойств анимации. Таких как: длительность анимаций, задержка перед анимациями, количество повторений анимаций и др;
   * редактирование обработчиков событий. Таких как: нажатие кнопкой мыши на элемент, наведение курсора на элемент, нажатие на клавишу и др;
6. создание таблиц стилей (CSS);
7. создание и редактирование CSS-анимаций;
8. создание JavaScript скриптов;
9. предпросмотр редактируемой анимации;

## 4.4 Требования к составу и параметрам технических средств

Минимальные технические требования:

1. Процессор: Intel Core i3 или аналогичный AMD;
2. Оперативная память: 4 Гб ОЗУ;
3. Место на диске: 256 Мб;

Оптимальные технические требования:

1. Процессор: Intel Core i5 или аналогичный AMD с тактовой частотой не менее 3 ГГц;
2. Оперативная память: 8 Гб ОЗУ;

Место на диске: 256 Мб;

## 4.5 Требования к информационной, программной совместимости

Программа должна функционировать на компьютере, поддерживающим браузер, созданный на одном из указанных движков: «WebKit», «Blink» или «Gecko».

## 4.6 Требования к программной документации

Программная документация должна быть предоставлена руководством пользователя.

# 5 Руководство пользователя

Для запуска необходимо дважды кликнуть по файлу index.py и запустить localhost в любом браузере (рис. 14).

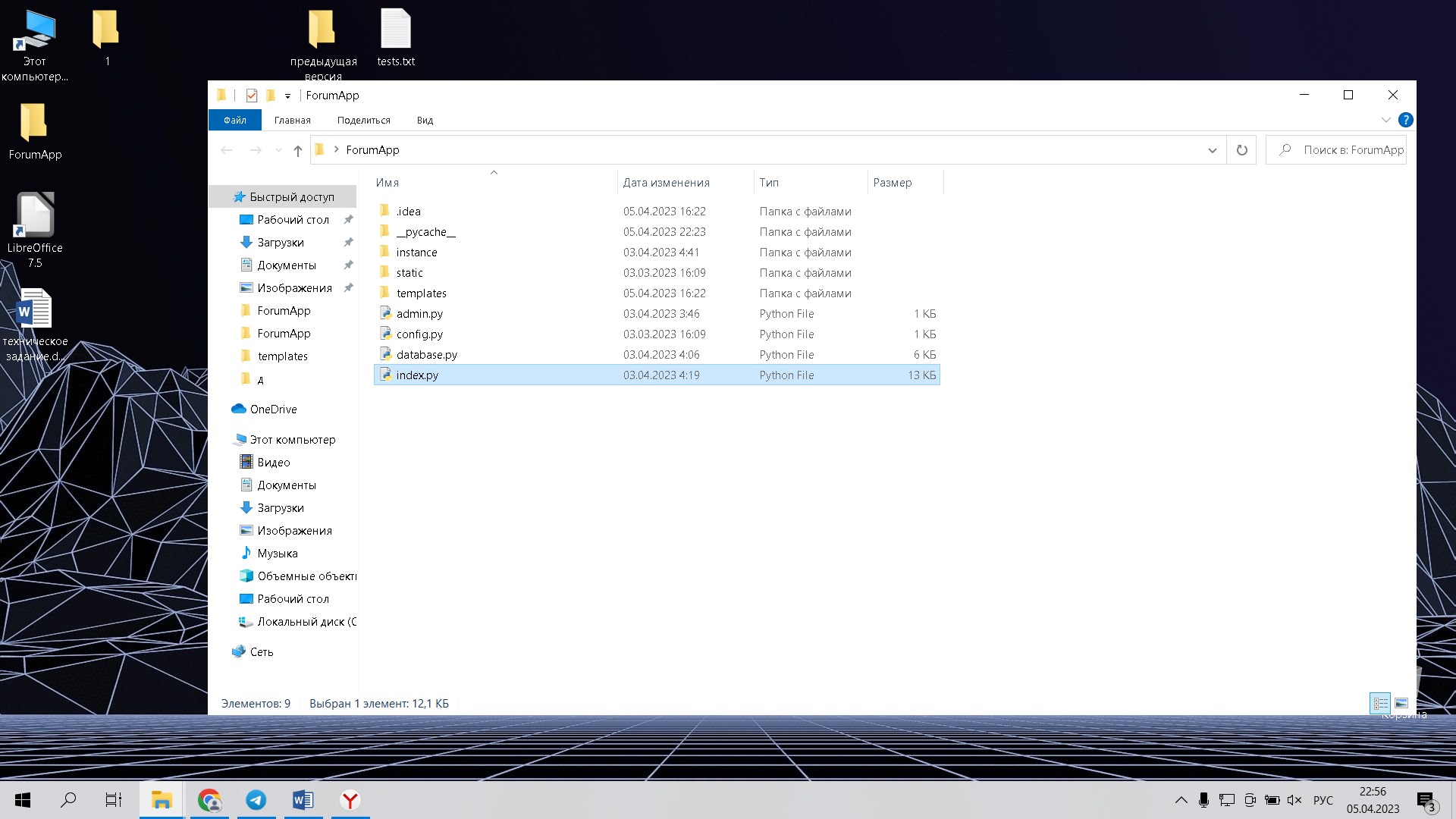


Рисунок 14 – Запуск веб-сервиса

Для работы на веб-сервисе необходимо авторизоваться (рис. 15).

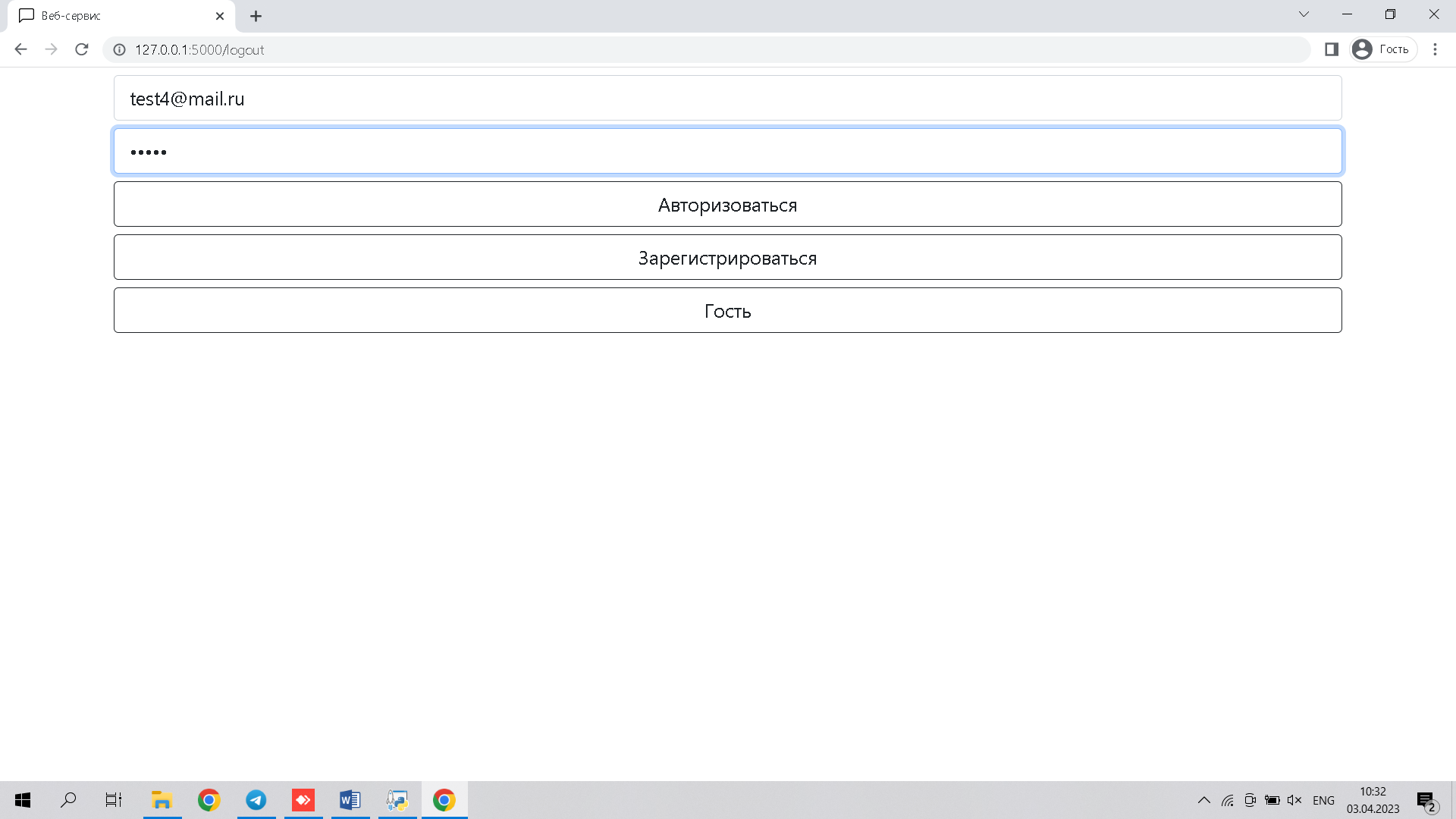


Рисунок 15 - Форма авторизации

После авторизации нам доступен список веток веб-сервиса (рис. 16). Рядом с названием каждой ветки отображается количество тем, которые к ней относятся.

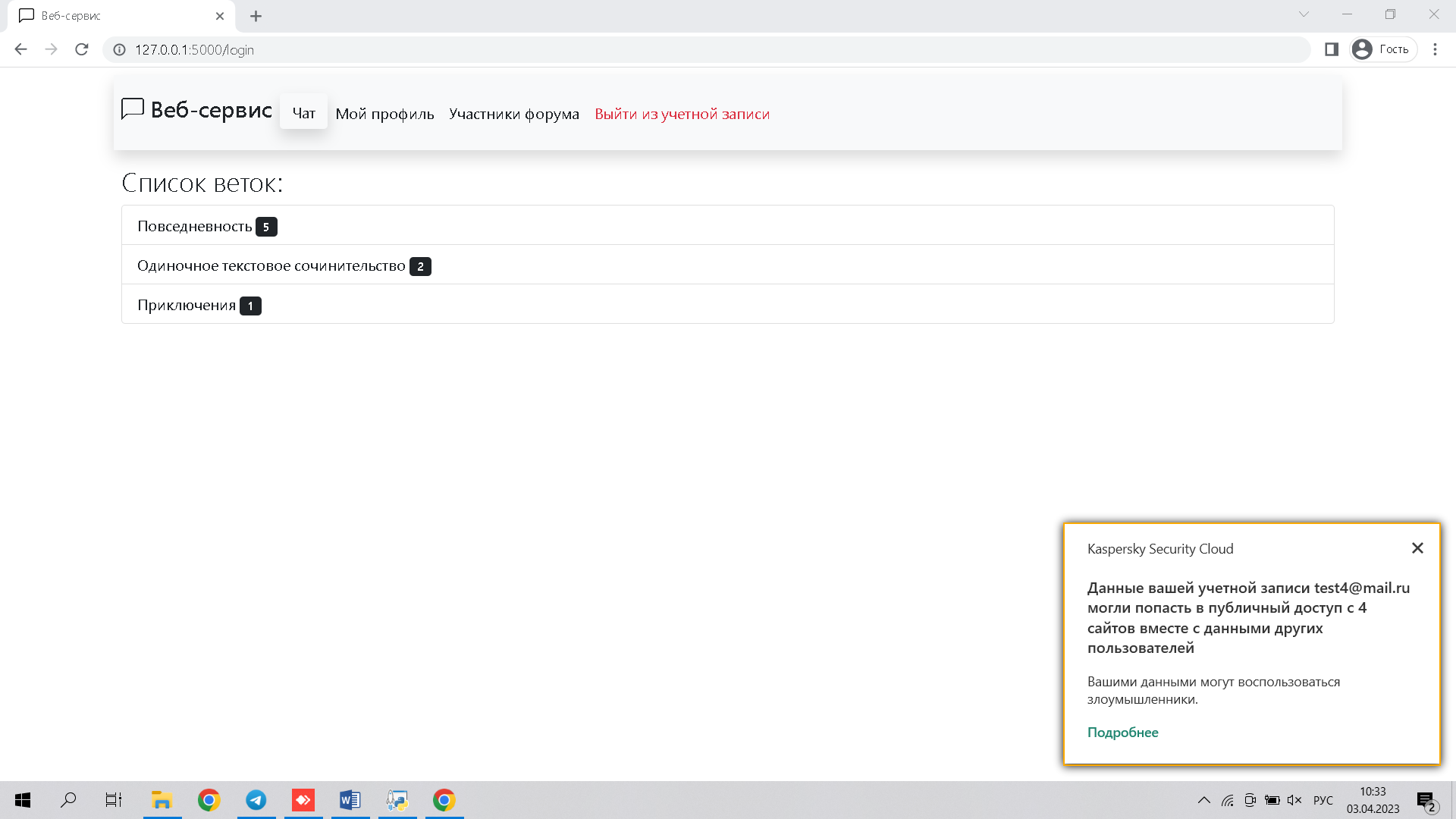


Рисунок 16 - Список веток

Выберем одну из веток (рис. 17). Часть веток не отображается в связи с настройкой в профиле пользователя. Данная настройка отвечает за отображение нежелательного контента.

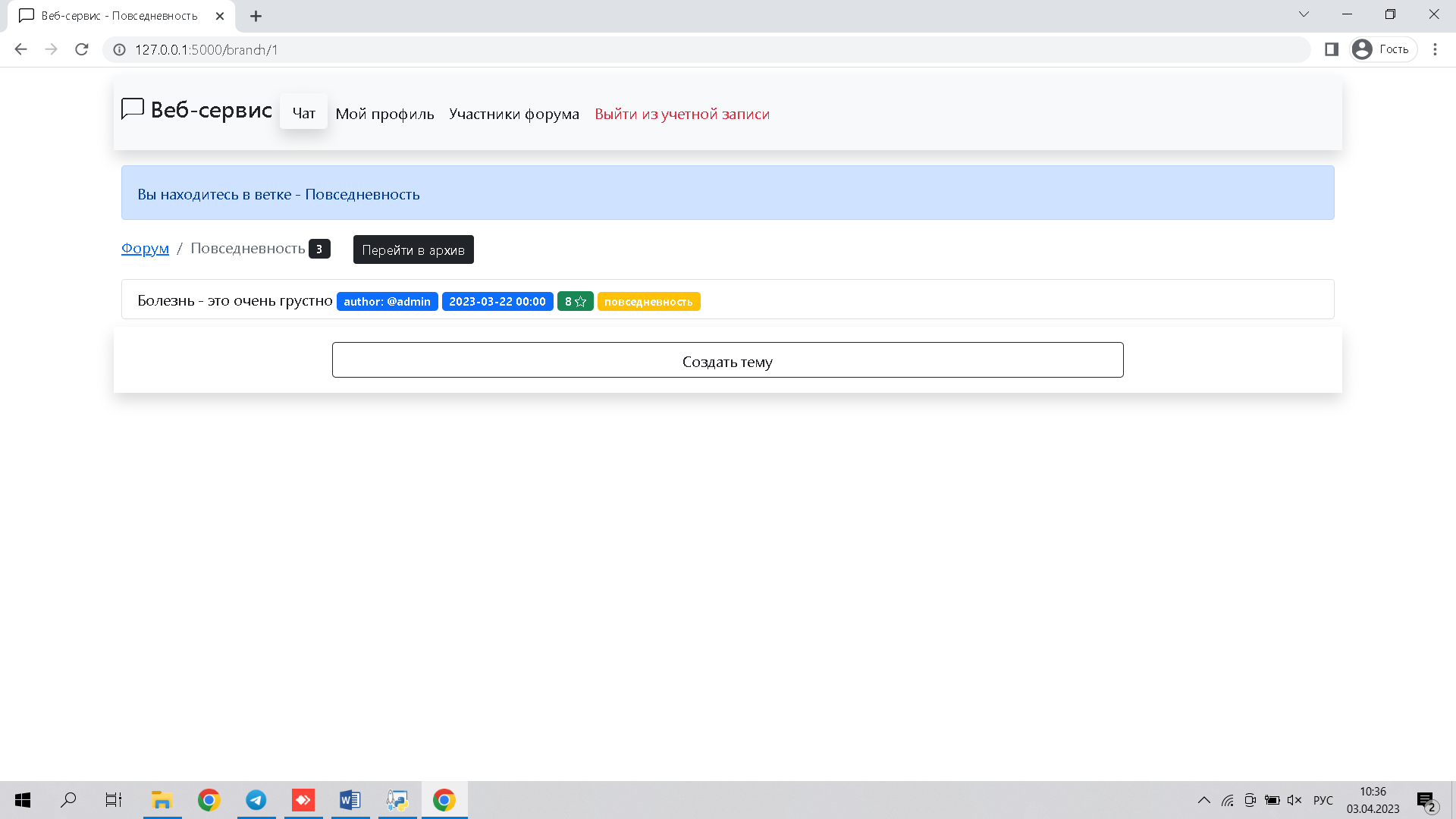


Рисунок 17 - Список тем в ветке

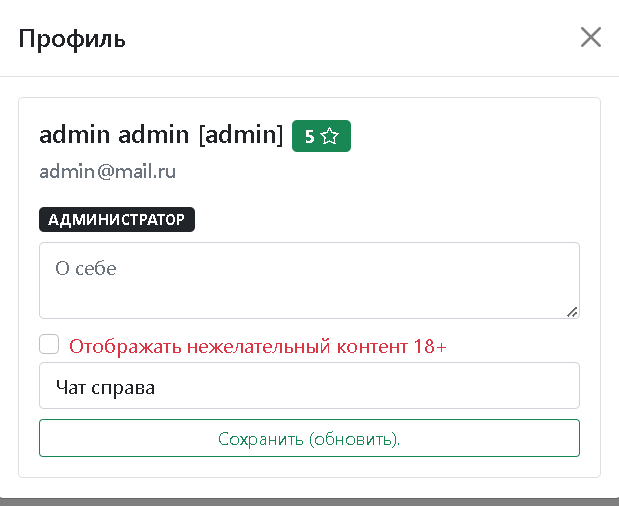


Рисунок 18 - Настройка в профиле

Создадим тему (рис. 19). Допустимо выбрать файл, который будет отображаться в теме, написать теги. Спам-фильтр проверяет на корректность названия темы и описания, а галочка нежелательного контента скроет информацию от пользователя с соответствующим требованием. Готовая тема отобразится в списке (рис. 20).

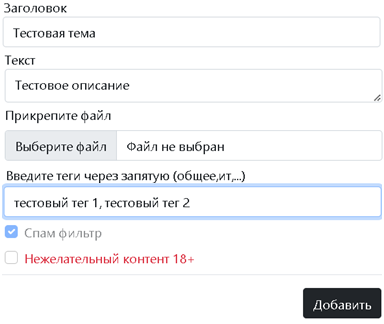


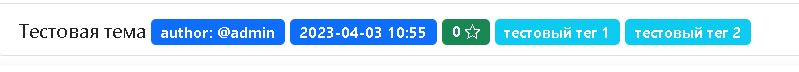
Рисунок 19 - Создание темы

Рисунок 20 - Отображенная тема в списке тем

После чего данная тема будет доступна для использования (рис. 21).

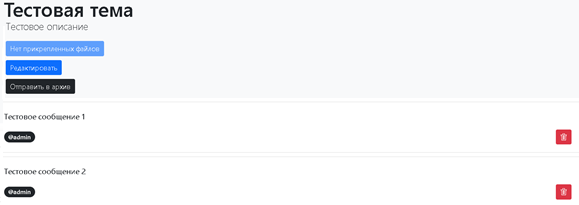


Рисунок 21 - Пример темы

ЗАКЛЮЧЕНИЕ

Реализуемый редактор HTML-анимации в виде веб-приложения является эффективным инструментом для создания мультипликации на основе современных технологий: HTML пятой версии и CSS третьей версии. Редактор позволяет упростить создание анимации для пользователей, с помощью его (редактора) функций: добавление и удаление анимируемых объектов; редактирование свойств анимируемых объектов; редактирование анимации объектов; возможность редактирования таблицы стилей; возможность создания JavaScript скриптов; возможность создания интерактивной анимации; возможность экспорта и импорта проекта в JSON и HTML форматы.

Дальнейшими перспективами реализуемого программного средства являются:

* возможность экспорта анимации в популярные видеоформаты: mp4, gif и др;
* использование технологий «intellisence» для дополнительной помощи пользователю при редактировании различных свойств анимируемых объектов: стиль, обработчики событий, таблица стилей, JavaScript скрипт и т.п;
* Наличие дополнительных интерфейсов при работе с 3D-анимацией для большего удобства пользователя за счёт упрощения работы;

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Википедия. Анимация [Элек тронный ресурс]. URL: https://en.wikipedia.org/wiki/Animation (дата обращения 1.01.2024).
2. W3School. CSS Animations [Электронный ресурс]. URL: https://www.w3schools.com/css/css3\_animations.asp (дата обращения 06.03.2024).
3. W3School. JavaScript HTML DOM animation [Электронный ресурс]. URL: https://www.w3schools.com/js/js\_htmldom\_animate.asp (дата обращения 06.03.2024).
4. Wikipedia. C Sharp [Электронный ресурс]. URL: <https://ru.wikipedia.org/wiki/C_Sharp> (дата обращения 06.03.2024).
5. Wikipedia. Blazor [Электронный ресурс]. URL: https://ru.wikipedia.org/wiki/Blazor (дата обращения 06.03.2024).
6. Wikipedia. WebAssembly [Электронный ресурс]. URL: https://ru.wikipedia.org/wiki/WebAssembly (дата обращения 06.03.2024).
7. Wikipedia. Progressive web app [Электронный ресурс]. URL: https://en.wikipedia.org/wiki/Progressive\_web\_app (дата обращения 06.03.2024).
8. Wikipedia. Native (computing) [Электронный ресурс]. URL: https://en.wikipedia.org/wiki/Native\_(computing) (дата обращения 06.03.2024).
9. Wikipedia. Visual Studio Code [Электронный ресурс]. URL: https://en.wikipedia.org/wiki/Visual\_Studio\_Code (дата обращения 06.03.2024).