

**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ  
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук  
Образовательная программа «Программная инженерия»

**СОГЛАСОВАНО**

Старший преподаватель департамента  
больших данных и информационного  
поиска

 В.В. Куренков  
«\_\_\_» \_\_\_\_\_ 2023 г.

**УТВЕРЖДАЮ**

Академический руководитель  
образовательной программы  
«Программная инженерия»  
профессор департамента программной  
инженерии, канд. техн. наук

 В.В. Шилов  
« 10 » \_\_\_\_\_ мая 2023 г.

|              |  |
|--------------|--|
| Подп. и дата |  |
| Инв. № дубл. |  |
| Взам. инв. № |  |
| Подп. и дата |  |
| Инв. № подл. |  |

**Система проверки решений по программированию для Лицея НИУ ВШЭ  
с автоматизированным подсчётом рейтинга**


**Пояснительная записка**

**ЛИСТ УТВЕРЖДЕНИЯ**

**RU.17701729.05.10-01 81 01-1-ЛУ**

Исполнитель

студент группы БПИ219

 / С.О. Котовский/  
«08» мая \_\_\_\_\_ 2023 г.

**Москва 2023**

УТВЕРЖДЕН  
RU.17701729.05.10-01 81 01-1-ЛУ

**Система проверки решений по программированию для Лицея НИУ ВШЭ  
с автоматизированным подсчётом рейтинга**

**Пояснительная записка**

**RU.17701729.05.10-01 81 01-1**

**Листов 21**

|                     |  |
|---------------------|--|
| <i>Подп. и дата</i> |  |
| <i>Инв. № дубл.</i> |  |
| <i>Взам. инв. №</i> |  |
| <i>Подп. и дата</i> |  |
| <i>Инв. № подл</i>  |  |

**Москва 2023**

## ОГЛАВЛЕНИЕ

|           |  |          |
|-----------|--|----------|
| <b>1.</b> | <b>ВВЕДЕНИЕ.....</b>   | <b>4</b> |
| 1.1.      | Наименование программы и объекта, в котором программа используется.....                        | 4        |
| 1.2.      | Краткая характеристика области применения программы и объекта, в котором она используется..... | 4        |
| <b>2.</b> | <b>ОСНОВАНИЯ ДЛЯ РАЗРАБОТКИ .....</b>  | <b>5</b> |
| <b>3.</b> | <b>НАЗНАЧЕНИЕ РАЗРАБОТКИ.....</b>  | <b>6</b> |
| 3.1.      | Функциональное назначение .....  | 6        |
| 3.2.      | Эксплуатационное назначение .....  | 6        |
| <b>4.</b> | <b>ТРЕБОВАНИЯ К ПРОГРАММЕ.....</b>   | <b>7</b> |
| 4.1.      | Постановка задачи на разработку программы .....  | 7        |
| 4.2.      | Описание алгоритма и функционирования веб-приложения .....                                     | 7        |
| 4.2.1.    | Описание алгоритма авторизации пользователя в системе .....                                    | 7        |
| 4.2.2     | Описание личного кабинета пользователя .....   | 7        |
| 4.2.3     | Описание верхнего меню (header) сайта .....  | 8        |
| 4.2.4     | Описание нижнего колонтитула (footer) сайта .....  | 8        |
| 4.2.5     | Описание админ-панели преподавателя.....   | 8        |
| 4.2.6     | Описание создание аккаунта студента .....  | 9        |
| 4.2.7     | Описание импорта контестов из Ejudge через админ-панель.....                                   | 10       |
| 4.2.8     | Описание списка контестов .....  | 10       |
| 4.2.9     | Описание списка посылок.....   | 11       |
| 4.2.10    | Описание страницы с рейтингом .....  | 12       |
| 4.2.11    | Описание страницы с контестом .....  | 12       |
| 4.3.      | Описание и обоснование выбора метода организации входных и выходных данных                     | 14       |
| 4.3.1     | AdminPanel.vue.....  | 14       |
| 4.3.2     | CreateStudentAccount.vue .....   | 14       |

|                         |         |            |        |         |
|-------------------------|---------|------------|--------|---------|
|                         |         |            |        |         |
| Изм.                    | Лист    | № докум.   | Подп.  | Дата    |
| RU.17701729.05.10-01 81 |         |            |        |         |
| Инв. № подл.            | Подп. и | Взам. инв. | Инв. № | Подп. и |

**RU.17701729.05.10-01 81 01–1**

|  |           |
|--|-----------|
| 4.3.3 ImportContest.vue .....  | 14        |
| 4.3.4 Contest.vue .....  | 15        |
| 4.3.5 LoginPage.vue .....  | 15        |
| 4.3.6 GlobalRating.vue .....   | 15        |
| 4.3.7 MyContestsList.vue:.....   | 16        |
| 4.3.8 ParcelsList.vue: .....   | 16        |
| 4.3.9 StudentProfile.vue:.....   | 16        |
| 4.3.10 SiteHeader.vue .....  | 16        |
| 4.3.11 SiteFooter.vue .....  | 16        |
| 4.4. Описание и обоснование состава технических и программных средств .....                            | 17        |
| <b>5. ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ.....</b>  | <b>17</b> |
| 5.1. Предполагаемая потребность .....  | 17        |
| 5.2. Экономические преимущества разработки по сравнению с отечественными и зарубежными аналогами ..... | 18        |
| <b>ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ.....</b>   | <b>20</b> |

|                         |         |            |        |         |
|-------------------------|---------|------------|--------|---------|
|                         |         |            |        |         |
| Изм.                    | Лист    | № докум.   | Подп.  | Дата    |
| RU.17701729.05.10-01 81 |         |            |        |         |
| Инв. № подл.            | Подп. и | Взам. инв. | Инв. № | Подп. и |

## 1. ВВЕДЕНИЕ

### 1.1. Наименование программы и объекта, в котором программа используется

Наименование темы разработки: «Система проверки решений по программированию для Лицея НИУ ВШЭ с автоматизированным подсчётом рейтинга».

Наименование темы разработки на английском языке: «HSE Lyceum Programming Testing System with Automated Rating Estimation».

Наименование объекта, в котором программа используется: Лицей НИУ ВШЭ.

### 1.2. Краткая характеристика области применения программы и объекта, в котором она используется

«Система проверки решений по программированию для Лицея НИУ ВШЭ с автоматизированным подсчётом рейтинга» - набор программных инструментов для оценивания решений задач на разных языках программирования для учащихся Лицея НИУ ВШЭ с рейтингом для каждого учащегося.

Лицей НИУ ВШЭ – общеобразовательное учреждение города Москва.

|                         |         |            |        |         |
|-------------------------|---------|------------|--------|---------|
|                         |         |            |        |         |
| Изм.                    | Лист    | № докум.   | Подп.  | Дата    |
| RU.17701729.05.10-01 81 |         |            |        |         |
| Инв. № подл.            | Подп. и | Взам. инв. | Инв. № | Подп. и |

**2. ОСНОВАНИЯ ДЛЯ РАЗРАБОТКИ**

Основанием для разработки является учебный план подготовки бакалавров по направлению 09.03.04 «Программная инженерия» и утвержденная академическим руководителем тема курсового проекта.

|                         |         |            |        |         |
|-------------------------|---------|------------|--------|---------|
|                         |         |            |        |         |
| Изм.                    | Лист    | № докум.   | Подп.  | Дата    |
| RU.17701729.05.10-01 81 |         |            |        |         |
| Инв. № подл.            | Подп. и | Взам. инв. | Инв. № | Подп. и |

### 3. НАЗНАЧЕНИЕ РАЗРАБОТКИ

#### 3.1. Функциональное назначение

Программа содержит в себе систему, тестирующую решения задач по программированию Лицея НИУ ВШЭ и автоматически подсчитывающую рейтинг каждого учащегося, исходя из его оценок и количества решенных задач. Имеет учебно-методическое назначение.

#### 3.2. Эксплуатационное назначение

Программа направлена на решение учащимися Лицея НИУ ВШЭ задач по программированию, которые им выдал тот или иной преподаватель, просмотр учащимися своего рейтинга по оценкам и количеству решенных задач среди других учащихся. Также программа направлена на просмотр преподавателями решений задач учащимися, получения их рейтинга в виде файлов.

Конечные пользователи – учащиеся и преподаватели информатики/программирования Лицея НИУ ВШЭ.

|                         |         |            |        |         |
|-------------------------|---------|------------|--------|---------|
|                         |         |            |        |         |
| Изм.                    | Лист    | № докум.   | Подп.  | Дата    |
| RU.17701729.05.10-01 81 |         |            |        |         |
| Инв. № подл.            | Подп. и | Взам. инв. | Инв. № | Подп. и |

## 4. ТРЕБОВАНИЯ К ПРОГРАММЕ

### 4.1. Постановка задачи на разработку программы

Целью проекта является предоставление пользователю готового к использованию веб-приложения на Vue.js. Приложение должно предоставлять пользователям возможность входа в систему, просмотра личных данных, отправки решений на задачи и получения вердикта, а также участия в конкурсах. Приложение должно быть удобным и простым в использовании. Приложение должно соответствовать основным требованиям безопасности и защиты данных пользователей.

### 4.2. Описание алгоритма и функционирования веб-приложения

#### 4.2.1. Описание алгоритма авторизации пользователя в системе

На странице расположены поля для ввода email и пароля, а также кнопка "Войти". При успешной авторизации происходит сохранение токена в localStorage и переход на соответствующую страницу в зависимости от роли пользователя (студент или преподаватель). Если авторизация не удалась, то отображается сообщение об ошибке в соответствующем блоке.

Код также содержит функции для получения информации о преподавателе при авторизации с ролью "преподаватель". Это делается с помощью запроса к API, передавая токен в заголовке запроса.

Код также включает обработку ошибок, которые могут возникнуть при попытке авторизации. В случае ошибки от сервера выдается сообщение о проблемах на стороне сервера, а при ошибке сети - сообщение о проблемах с соединением.

Интерфейс авторизации отображается в окне браузера в центре страницы. Кроме того, на странице присутствует логотип лица и нижний колонтитул.

#### 4.2.2 Описание личного кабинета пользователя

Страница личного кабинета пользователя отображает информацию о профиле студента, его рейтинге и личной информации. На странице присутствует заголовок "Мой профиль", под которым располагаются поля с именем, направлением и группой студента. Далее идет блок с личной информацией, который включает в себя ID, электронную почту, класс, программу и группу студента. В блоке рейтинга отображаются два блока - "Место в рейтинге по оценкам" и "Место в рейтинге по задачам", каждый из которых содержит название рейтинга и соответствующее место студента в нем. Страница также содержит футер сайта.

|                         |         |            |        |         |
|-------------------------|---------|------------|--------|---------|
| Изм.                    | Лист    | № докум.   | Подп.  | Дата    |
| RU.17701729.05.10-01 81 |         |            |        |         |
| Инв. № подл.            | Подп. и | Взам. инв. | Инв. № | Подп. и |



### 4.2.3 Описание верхнего меню (header) сайта

Header компонент является частью интерфейса веб-приложения и отвечает за отображение шапки сайта. Он содержит элементы управления, такие как меню навигации, имя пользователя и кнопки для логина и логаута. Код компонента написан на языке Vue.js и использует библиотеку axios для выполнения HTTP запросов к серверу, а также роутер Vue Router для перехода между страницами.

Метод `created()` используется для выполнения определенных действий при создании компонента. В данном случае, метод `created()` используется для проверки авторизации пользователя при помощи вызова функции `checkAuth()`. Если пользователь не авторизован, то он будет перенаправлен на страницу логина.

Метод `handleLogout()` вызывается при нажатии на кнопку "Выйти". Он очищает локальное хранилище и перенаправляет пользователя на страницу логина при помощи вызова функции `logout()`.

Метод `fetchTeacherInfo()` и метод `fetchStudentInfo()` используются для получения информации о пользователе с сервера, в зависимости от того, является ли пользователь преподавателем или студентом. Они отправляют GET запрос на сервер при помощи библиотеки axios, и полученные данные сохраняются в локальном состоянии компонента. Эти методы вызываются в методе `created()` и используются для отображения имени пользователя в шапке сайта.

В компоненте также содержатся различные условные операторы, которые отображают различные элементы шапки в зависимости от того, является ли пользователь студентом или преподавателем. Например, кнопка "Админ-панель" отображается только для преподавателей, а кнопка "Личный кабинет" только для студентов.

Компонент определяет переменную `userRole`, которая хранит роль пользователя. Переменная получается из хранилища приложения Vuex.

### 4.2.4 Описание нижнего колонтитула (footer) сайта

Это компонент футера сайта, отображающий копирайт и прочую информацию о сайте. Он содержит тег `<footer>` и имеет класс `footer-container`, задающий его высоту, цвет фона, выравнивание и позиционирование. Также в компоненте определен пустой объект `SiteFooter`, что может быть использован в дальнейшем для добавления другой функциональности.

### 4.2.5 Описание админ-панели преподавателя.

Этот блок представляет компонент "Админ-панель преподавателя" на веб-странице.

Он имеет следующую структуру:

|                         |         |            |        |         |
|-------------------------|---------|------------|--------|---------|
|                         |         |            |        |         |
| Изм.                    | Лист    | № докум.   | Подп.  | Дата    |
| RU.17701729.05.10-01 81 |         |            |        |         |
| Инв. № подл.            | Подп. и | Взам. инв. | Инв. № | Подп. и |

## RU.17701729.05.10-01 81 01–1

- В секции `script` импортируются необходимые библиотеки и компоненты `SiteHeader` и `SiteFooter`.
- Далее определяется объект `export default`, который содержит:
  - Определение имени компонента `AdminPanel`.
  - Конфигурацию компонента, включающую:
    - Определение объекта данных `teacher`, который содержит информацию о преподавателе, включая `id`, `first_name`, `last_name`, `middle_name`, `email`, `department` и `group`.
    - Создание метода `fetchTeacherInfo`, который использует библиотеку `axios` для получения информации о преподавателе из API и сохранения ее в объекте данных `teacher`.
    - Создание метода `created`, который вызывает метод `fetchTeacherInfo` при создании компонента.
  - Определение шаблона в секции `template`, который содержит:
    - Включение компонентов `SiteHeader` и `SiteFooter`.
    - Заголовок главной части "Админ-панель преподавателя".
    - Информацию о преподавателе, отображаемую в виде списка элементов.
    - Две кнопки для перехода на страницы "Создать аккаунт студента" и "Импорт контеста из Ejudge".
  - Определение стилей в секции `style`, которые включают некоторые правила для расположения элементов на странице и стилизации кнопок.

Когда компонент `AdminPanel` загружается, метод `fetchTeacherInfo` вызывается автоматически для получения информации о преподавателе. Компонент затем отображает эту информацию в своем шаблоне и предоставляет две кнопки для выполнения дополнительных действий на сайте.

### 4.2.6 Описание создание аккаунта студента

Это компонент `Vue.js` с именем `CreateStudentAccount.vue`. Он включает в себя импорт компонентов `SiteFooter.vue`, `SiteHeader.vue` и модуля `axios`, используемого для отправки HTTP-запросов.

Определен объект `export default`, который имеет свойства `name`, `components` и `data`, а также метод `register()`. Свойство `name` задает имя компонента. Свойство `components` содержит компоненты, которые будут использоваться в данном компоненте. Свойство `data` возвращает объект с полями для каждого поля формы.

Метод `register()` выполняет отправку данных формы на сервер, используя метод `axios.post()`. После отправки запроса на сервер, результаты выводятся в консоль с помощью `console.log()`.

Шаблон `template` отображает форму с полями для ввода данных, и кнопкой отправки формы. Кнопка имеет обработчик события `@submit.prevent`, который вызывает метод `register()`. Внутри каждого поля ввода используется директива `v-model`, чтобы связать данные формы с объектом `data` компонента.

|                         |         |            |        |         |
|-------------------------|---------|------------|--------|---------|
|                         |         |            |        |         |
| Изм.                    | Лист    | № докум.   | Подп.  | Дата    |
| RU.17701729.05.10-01 81 |         |            |        |         |
| Инв. № подл.            | Подп. и | Взам. инв. | Инв. № | Подп. и |

## RU.17701729.05.10-01 81 01–1

В блоке `style` определяется стиль формы, включая размер и отступы полей ввода, цвет кнопки и радиусы углов элементов формы.

В целом, данный код представляет собой компонент `Vue.js`, который отображает форму создания профиля студента и выполняет отправку данных формы на сервер при нажатии на кнопку.

### 4.2.7 Описание импорта контекстов из Ejudge через админ-панель

Данный компонент `Vue.js`, реализует функционал импорта конкурса из системы Ejudge в базу данных нашего сервиса.

В начале файла импортируются компоненты `SiteFooter`, `SiteHeader` и библиотека `axios`, используемая для отправки HTTP-запросов. Затем экспортируется объект, содержащий определение компонента с именем `"ImportContest"`.

Определение компонента содержит свойства:

- `processing (boolean)` - флаг, показывающий, находится ли компонент в процессе обработки данных;
- `contestAdded (boolean)` - флаг, показывающий, был ли конкурс успешно добавлен;
- `contestId (string)` - идентификатор конкурса в системе Ejudge;
- `startTime (string)` - дата и время начала конкурса;
- `endTime (string)` - дата и время окончания конкурса;
- `contestName (string)` - название конкурса.

Далее определены методы:

- `importContest ()` - метод, который отправляет запрос на авторизацию в системе Ejudge, получает список задач, и затем отправляет данные о конкурсе на сервер управления конкурсами;
- `sendContestToServer ()` - метод, который отправляет запрос на добавление конкурса на сервер управления конкурсами.

В шаблоне компонента определена форма, содержащая поля ввода для идентификатора конкурса, даты и времени начала и окончания конкурса и названия конкурса. При отправке формы вызывается метод `importContest ()`. При обработке данных на странице отображается текст "Обработка...", а при успешном добавлении конкурса - "Конкурс успешно добавлен!".

Также в компоненте определены стили, включая стили для формы и текста статусов.

### 4.2.8 Описание списка конкурсов

Это `Vue.js` компонент, который отображает список конкурсов (соревнований) для пользователей. Компонент содержит следующие основные части:

|                         |         |            |        |         |
|-------------------------|---------|------------|--------|---------|
|                         |         |            |        |         |
| Изм.                    | Лист    | № докум.   | Подп.  | Дата    |
| RU.17701729.05.10-01 81 |         |            |        |         |
| Инв. № подл.            | Подп. и | Взам. инв. | Инв. № | Подп. и |

## RU.17701729.05.10-01 81 01–1

1. **Script:** В этой части импортируются зависимости, такие как `SiteFooter`, `SiteHeader` и `axios`. Затем определяется новый компонент с именем `ContestsList`. В объекте компонента определены свойства `components`, `data`, `methods` и `created`.
2. **Template:** В этой части определена разметка компонента, включая таблицу, которая отображает список конкурсов с деталями каждого конкурса.
3. **Style:** В этом разделе определены стили для кнопки "Перейти" в таблице.

Давайте разберем каждую часть подробнее:

### Script:

- `components`: Включает в себя `SiteHeader` и `SiteFooter`.
- `data`: Содержит состояние компонента, включая массив конкурсов.
- `methods`: Включает методы для сохранения идентификатора конкурса в локальное хранилище (`saveContestId`) и получения списка конкурсов с сервера (`fetchContests`).
- `created`: Жизненный цикл `Vue.js`, который вызывается при создании компонента. В данном случае вызывается метод `fetchContests`.

### Template:

- Разметка состоит из основных блоков, таких как шапка сайта, контейнер с основным содержимым и подвал сайта.
- Внутри контейнера с основным содержимым есть заголовок, отображающий количество конкурсов в системе и таблица с деталями каждого конкурса.
- В таблице присутствует кнопка "Перейти", которая сохраняет идентификатор конкурса в локальное хранилище и перенаправляет пользователя на страницу конкурса.

### Style:

- В этом разделе определены стили для кнопки "Перейти", включая цвет фона, отступы, радиус границы, ширину и другие свойства.
- Стили также включают плавные переходы при наведении курсора, нажатии и фокусировке на кнопке.

### 4.2.9 Описание списка посылок

Это `Vue.js` компонент, который отображает список посылок (`parcels`) для пользователей с разными ролями (учитель или студент). Компонент содержит следующие основные части:

1. **Script:** В этой части импортируются зависимости, такие как `SiteFooter`, `SiteHeader` и `axios`. Затем определяется новый компонент с именем `AdminPanel`. В объекте компонента определены свойства `components`, `data`, `computed`, `methods` и `created`.
2. **Template:** В этой части определена разметка компонента, включая таблицу, которая отображает список посылок с деталями каждой посылки. Также присутствует модальное окно, отображающее код посылки.
3. **Style:** В этом разделе определены стили для модального окна и его содержимого.

|                         |         |            |        |         |
|-------------------------|---------|------------|--------|---------|
|                         |         |            |        |         |
| Изм.                    | Лист    | № докум.   | Подп.  | Дата    |
| RU.17701729.05.10-01 81 |         |            |        |         |
| Инв. № подл.            | Подп. и | Взам. инв. | Инв. № | Подп. и |

**RU.17701729.05.10-01 81 01–1**

Давайте разберем каждую часть подробнее:

**Script:**

- **components:** Включает в себя `SiteHeader` и `SiteFooter`.
- **data:** Содержит состояние компонента, включая массив посылок, переменные для модального окна и выбранного кода.
- **computed:** Включает вычисляемое свойство `mainHeadline`, которое определяет, какой заголовок отображать на основе роли пользователя.
- **methods:** Включает методы для получения посылок с сервера (`fetchParcels`), открытия модального окна (`openCodeModal`), закрытия модального окна (`closeModal`).
- **created:** Жизненный цикл `Vue.js`, который вызывается при создании компонента. В данном случае вызывается метод `fetchParcels`.

**Template:**

- Разметка состоит из основных блоков, таких как шапка сайта, контейнер с основным содержимым и подвал сайта.
- Внутри контейнера с основным содержимым есть заголовок, отображающий количество посылок и таблица с деталями каждой посылки.
- Для каждой посылки в таблице есть кнопка, которая открывает модальное окно с кодом посылки.

**Style:**

- В этом разделе определены стили для модального окна и его содержимого, включая расположение, размер, цвет фона и другие свойства.

**4.2.10 Описание страницы с рейтингом**

Это `Vue.js` компонент, который отображает рейтинг пользователей по двум критериям: по оценкам и по количеству решенных задач.

Компонент состоит из трех основных разделов: `script`, `template` и `style`.

**Script:** • Импортируются зависимости: `SiteFooter`, `SiteHeader` и `axios`. • Создается компонент с именем `"GlobalRating"` и определяются свойства `components`, `data`, `methods`, и `mounted`. • **components:** включает `SiteHeader` и `SiteFooter`. • **data:** содержит состояние компонента: `ratingType` (тип рейтинга), `ratingData` (данные рейтинга) и `error` (сообщение об ошибке). • **methods:** определяет методы `getRatingData` (получение данных рейтинга), `changeRating` (изменение типа рейтинга) и `downloadCSV` (выгрузка рейтинга в формате CSV). • **mounted:** Жизненный цикл `Vue.js`, который вызывается при монтировании компонента. В данном случае вызывается метод `getRatingData`.

**Template:** • Разметка состоит из основных блоков: шапка сайта, контейнер с основным содержимым и подвал сайта. • Внутри контейнера с основным содержимым размещены

|                         |         |            |        |         |
|-------------------------|---------|------------|--------|---------|
|                         |         |            |        |         |
| Изм.                    | Лист    | № докум.   | Подп.  | Дата    |
| RU.17701729.05.10-01 81 |         |            |        |         |
| Инв. № подл.            | Подп. и | Взам. инв. | Инв. № | Подп. и |

**RU.17701729.05.10-01 81 01–1**

кнопки для переключения типа рейтинга, кнопка для загрузки рейтинга в формате CSV, а также таблица с информацией о пользователях и их рейтинге.

Style: • В этом разделе определены стили для кнопок переключения типа рейтинга, таблицы и ее элементов (заголовков и ячеек). • Стили также определяют анимацию кнопок при наведении курсора и стиль сообщения об ошибке.

Компонент позволяет пользователю выбирать между двумя типами рейтинга, а затем отображает список пользователей с соответствующими данными рейтинга в виде таблицы. Пользователь также может загрузить рейтинг в формате CSV. Для получения и обновления данных рейтинга используется библиотека `axios` для выполнения HTTP-запросов.

**4.2.11 Описание страницы с контекстом**

Это Vue.js компонент с именем "Contest". Компонент использует два дочерних компонента, `SiteHeader` и `SiteFooter`, а также импортирует библиотеку `axios` для выполнения HTTP-запросов.

Внутри компонента определены данные (`data`), вычисляемые свойства (`computed`), методы жизненного цикла (`created`), и методы (`methods`).

Вычисляемое свойство `"contestName"` получает название конкурса из локального хранилища.

В данных (`data`) компонента определены следующие поля:

- `processing`: флаг, указывающий, идет ли обработка решения в данный момент.
- `failureTest`: номер теста, на котором произошла ошибка.
- `solutionStatus`: статус последнего отправленного решения.
- `selectedFile`: выбранный файл с решением.
- `tasks`: массив с информацией о задачах.
- `currentTask`: текущая задача.
- `taskTitle` и `taskDescription`: название и описание текущей задачи.
- `contestName`: название конкурса.
- `selectedLanguage`: выбранный язык программирования.

Методы жизненного цикла и методы компонента включают:

- `fetchTasks`: получает список задач с сервера и обновляет текущую задачу.
- `onLanguageChange`: обновляет выбранный язык программирования.
- `updateTask`: обновляет текущую задачу и ее описание.
- `authenticate`: выполняет авторизацию на сервере и возвращает данные авторизации.
- `onFileChange`: обрабатывает выбор файла с решением.
- `convertFileToBase64` и `readFileAsText`: конвертируют файл в Base64 и читают его как текст, соответственно.
- `submitSolution`: отправляет решение на сервер, получает результат и вызывает метод `fetchResult`.

|                         |         |            |        |         |
|-------------------------|---------|------------|--------|---------|
|                         |         |            |        |         |
| Изм.                    | Лист    | № докум.   | Подп.  | Дата    |
| RU.17701729.05.10-01 81 |         |            |        |         |
| Инв. № подл.            | Подп. и | Взам. инв. | Инв. № | Подп. и |

**RU.17701729.05.10-01 81 01–1**

- `submitSolutionToServer`: отправляет решение на другой сервер с информацией о результатах тестирования.
- `fetchResult`: получает результат тестирования решения и вызывает метод `submitSolutionToServer`.

В разделе `<template>` определена структура HTML для отображения компонента, включая заголовок, описание задачи, статус последнего решения и кнопку отправки решения.

Также на странице есть таблица с информацией о последних 4 посылках, где для каждой посылки указан номер, язык и вердикт, а также форма отправки своего решения, где пользователь может выбрать файл с решением, язык программирования и отправить его на сервер для проверки.

Важно отметить, что на странице используется механизм двусторонней привязки данных `v-model`, который связывает значения на странице с данными в компоненте. Также есть обработчики событий, которые отслеживают изменения в форме и отправляют данные на сервер для проверки.

В разделе `<style>` определены стили CSS для оформления компонента.

### **4.3. Описание и обоснование выбора метода организации входных и выходных данных**

#### **4.3.1 AdminPanel.vue**

- Входные данные: информация о преподавателе, полученная с сервера при загрузке страницы.
- Выходные данные: Запросы на сервер для создания нового аккаунта студента / импорта конкурса из Ejudge.

#### **4.3.2 CreateStudentAccount.vue**

- Входные данные: пользователь вводит информацию о студенте (имя, фамилия, отчество, почта, роль, пароль, направление, группа) через текстовые поля.
- Выходные данные: при отправке формы, данные о студенте отправляются на сервер для регистрации нового аккаунта студента.

#### **4.3.3 ImportContest.vue**

- Входные данные: пользователь вводит ID конкурса, время начала, время окончания и название конкурса через текстовые поля и поля выбора времени.

|                         |         |            |        |         |
|-------------------------|---------|------------|--------|---------|
|                         |         |            |        |         |
| Изм.                    | Лист    | № докум.   | Подп.  | Дата    |
| RU.17701729.05.10-01 81 |         |            |        |         |
| Инв. № подл.            | Подп. и | Взам. инв. | Инв. № | Подп. и |

- Выходные данные: при отправке формы, данные отправляются на сервер для импорта конкурса из Ejudge и создания нового конкурса в системе с указанными данными.

#### 4.3.4 Contest.vue

На данной странице осуществляется работа с задачами и посылками решений. Входными данными являются выбранное решение пользователя и выбранный язык программирования. Выходными данными являются результаты посылки, статус решения и номер теста, на котором произошла ошибка.

- Организация входных данных:
  1. Выбор файла решения: `<input type="file" id="solution" name="solution" @change="onFileChange">`
  2. Выбор языка программирования: `<select name="language" id="language" @change="onLanguageChange">`
- Организация выходных данных:
  1. Результат последней посылки: `<h3>Последняя посылка: <span class="last-status">{{ solutionStatus }}</span></h3>`
  2. Ошибка на тесте: `<h3>Ошибка на тесте: <span class="last-status">{{ failureTest }}</span></h3>`
  3. Список последних 4 посылок: `<ol class="submissions">...</ol>`

При нажатии на кнопку "Отправить", вызывается метод `submitSolution`, который преобразует выбранный файл в Base64, читает его содержимое и отправляет его на сервер для обработки. Затем получает результаты обработки, обновляет статус решения и номер теста с ошибкой, и отправляет решение на другой сервер для сохранения.

#### 4.3.5 LoginPage.vue

Эта страница предназначена для авторизации пользователей. Входные данные включают email и пароль, которые указываются в соответствующих полях формы. Когда пользователь отправляет форму, выполняется метод `submitForm`, в котором отправляется запрос на сервер для проверки email и пароля. Если авторизация проходит успешно, сервер возвращает токен, который сохраняется в `localStorage`. Затем выполняется запрос для получения роли пользователя, и основываясь на роли, происходит переход на соответствующую страницу (StudentProfile или AdminPanel). В случае ошибки, на странице отображается сообщение об ошибке.

#### 4.3.6 GlobalRating.vue

Эта страница предназначена для отображения рейтинга студентов в зависимости от выбранного типа (по оценкам или по задачам). Кнопки "Рейтинг по оценкам" и "Рейтинг по задачам" позволяют пользователю выбирать тип рейтинга. При изменении типа рейтинга вызывается метод `changeRating`, который обновляет данные на странице.

|                         |         |            |        |         |
|-------------------------|---------|------------|--------|---------|
|                         |         |            |        |         |
| Изм.                    | Лист    | № докум.   | Подп.  | Дата    |
| RU.17701729.05.10-01 81 |         |            |        |         |
| Инв. № подл.            | Подп. и | Взам. инв. | Инв. № | Подп. и |



**RU.17701729.05.10-01 81 01–1**

- Входные данные на этой странице — это тип рейтинга (marks или tasks).
- Выходные данные — это таблица с рейтингом студентов, включающая Имя, Фамилию, Отчество, Email, Роль, Факультет и Группу.

**4.3.7 MyContestsList.vue:**

- Входные данные: Токен пользователя, который хранится в локальном хранилище.
- Выходные данные: Список конкурсов, который выводится в виде таблицы с информацией о каждом конкурсе (ID в Ejudge, название, дата начала, дата окончания и ссылка на конкурс).

**4.3.8 ParcelsList.vue:**

- Входные данные: Токен пользователя и роль пользователя, которые хранятся в локальном хранилище и Vuex Store соответственно.
- Выходные данные: Список посылок, который выводится в виде таблицы с информацией о каждой посылке (ID, просмотр кода, язык, статус, время работы, использованная память, тест, название конкурса, название задачи).

**4.3.9 StudentProfile.vue:**

- Входные данные: Токен пользователя, который хранится в локальном хранилище.
- Выходные данные: Информация о профиле студента, включая имя, направление, группу, ID, электронную почту, класс, программу и место в рейтинге по оценкам и задачам. Эти данные выводятся в виде текста на странице профиля студента.

**4.3.10 SiteHeader.vue**

- Входные данные: Роль пользователя (из хранилища Vuex, `this.$store.state.userRole`) и токен авторизации (из `localStorage`, `localStorage.getItem("token")`).
- Выходные данные: Фамилия и имя пользователя (из `user.first_name` и `user.middle_name`). Они выводятся в шапке сайта.

В процессе создания компонента `SiteHeader.vue` выполняются следующие действия:

1. С помощью функции `checkAuth` проверяется, авторизован ли пользователь.
2. В зависимости от роли пользователя (учитель или студент) выполняется запрос к серверу для получения информации о пользователе.
3. Отображается меню навигации и информация о пользователе.

**4.3.11 SiteFooter.vue**

|                         |         |            |        |         |
|-------------------------|---------|------------|--------|---------|
|                         |         |            |        |         |
| Изм.                    | Лист    | № докум.   | Подп.  | Дата    |
| RU.17701729.05.10-01 81 |         |            |        |         |
| Инв. № подл.            | Подп. и | Взам. инв. | Инв. № | Подп. и |

**RU.17701729.05.10-01 81 01–1**

На странице SiteFooter.vue нет входных и выходных данных. Это простой компонент, который отображает статическую информацию в подвале сайта.

Все страницы используют axios для отправки запросов к серверу и получения данных. Входные данные обрабатываются с помощью Vue.js и отправляются на сервер для дальнейшей обработки.

**4.4. Описание и обоснование состава технических и программных средств**

Для обеспечения надежной работы проекта необходим следующий состав технических и программных средств:

## 1. Серверная часть:

- Операционная система: Linux-based (например, Ubuntu 20.04 LTS).
- Web-сервер: Nginx.
- Среда выполнения: Node.js (версия 14.x или выше).

## 2. Клиентская часть:

- Веб-браузеры: Google Chrome (версия 70 и выше), Mozilla Firefox (версия 63 и выше), Safari (версия 12 и выше), Microsoft Edge (версия 79 и выше).
- Разрешение экрана: минимальное 1024x768 пикселей.
- JavaScript: должен быть включен для корректной работы приложения.

Выбор технических и программных средств обосновывается следующими соображениями:

- Linux-based операционная система обеспечивает высокую стабильность и безопасность сервера, а также широкие возможности настройки и управления.
- Nginx в качестве веб-сервера обеспечивает высокую производительность и надежность, хорошо справляется с большим количеством одновременных запросов.
- Node.js позволяет создавать высокопроизводительные серверные приложения с использованием языка программирования JavaScript.
- Поддержка современных веб-браузеров и разрешений экрана позволяет достичь максимальной совместимости и удобства использования приложения для большинства пользователей.
- Включенный JavaScript обеспечивает корректную работу интерактивных элементов пользовательского интерфейса и обработку запросов на стороне клиента.

**5. ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ****5.1. Предполагаемая потребность**

Программа будет использоваться учащимися Лицея НИУ ВШЭ для решения задач по программированию и просмотра рейтинга учащихся в режиме реального времени, а также

|                         |         |            |        |         |
|-------------------------|---------|------------|--------|---------|
|                         |         |            |        |         |
| Изм.                    | Лист    | № докум.   | Подп.  | Дата    |
| RU.17701729.05.10-01 81 |         |            |        |         |
| Инв. № подл.            | Подп. и | Взам. инв. | Инв. № | Подп. и |

**RU.17701729.05.10-01 81 01–1**

преподавателями Лицея НИУ ВШЭ для отслеживания количества решенных задач учащимися, контроля рейтинга, получения информации об оценках.

### **5.2. Экономические преимущества разработки по сравнению с отечественными и зарубежными аналогами**

(текст или таблица – Яндекс Контест, Informatics, Timus Online judge, nlogn)

|   | Timus<br>Online Judge | Яндекс<br>Контест | Informatics | NLOGN | Разрабатываемое<br>решение |
|---|-----------------------|-------------------|-------------|-------|----------------------------|
| Возможность<br>регистрации без<br>приглашения   | +                     | +                 | +           | -     | -                          |
| Рейтинг по<br>определенному<br>соревнованию   | +                     | +                 | +           | ?     | +                          |
| Общий рейтинг<br>пользователей<br>системы   | +                     | -                 | +           | -     | +                          |
| Возможность<br>автоматического<br>выставления<br>оценки за<br>соревнование                            | -                     | -                 | -           | -     | +                          |
| Возможность<br>выгрузки таблицы<br>с оценками<br>(рейтингом) всех<br>пользователей за<br>соревнования | ?                     | +                 | -           | -     | +                          |
| Наличие архива<br>задач, из которых<br>можно создавать<br>соревнование                                | +                     | -                 | +           | -     | +                          |

|                         |         |            |        |         |
|-------------------------|---------|------------|--------|---------|
| Изм.                    | Лист    | № докум.   | Подп.  | Дата    |
| RU.17701729.05.10-01 81 |         |            |        |         |
| Инв. № подл.            | Подп. и | Взам. инв. | Инв. № | Подп. и |

## RU.17701729.05.10-01 81 01–1

|  |    |    |    |    |   |
|--|----|----|----|----|---|
| Количество<br>поддерживаемых<br>языков<br>программирования | 18 | 30 | 16 | 16 | 14*<br>*на момент<br>защиты<br>доступно 14<br>языков, точный<br>список будет<br>обсуждаться с<br>заказчиком<br>проекта. |
|--|----|----|----|----|---|

|                         |         |            |        |         |
|-------------------------|---------|------------|--------|---------|
|                         |         |            |        |         |
| Изм.                    | Лист    | № докум.   | Подп.  | Дата    |
| RU.17701729.05.10-01 81 |         |            |        |         |
| Инв. № подл.            | Подп. и | Взам. инв. | Инв. № | Подп. и |

[illegible]

|                         |         |            |        |         |
|-------------------------|---------|------------|--------|---------|
|                         |         |            |        |         |
| Изм.                    | Лист    | № докум.   | Подп.  | Дата    |
| RU.17701729.05.10-01 81 |         |            |        |         |
| Инв. № подл.            | Подп. и | Взам. инв. | Инв. № | Подп. и |