|  |
| --- |
| **МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ** |
| **ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕХНОЛОГИЙ И УПРАВЛЕНИЯ ИМЕНИ К.Г. РАЗУМОВСКОГО (ПЕРВЫЙ КАЗАЧИЙ УНИВЕРСИТЕТ)»**  **(ФГБОУ ВО «МГУТУ ИМ. К.Г. РАЗУМОВСКОГО (ПКУ)»)** |
|  |
| **УНИВЕРСИТЕТСКИЙ КОЛЛЕДЖ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ** |

|  |  |
| --- | --- |
|  | ДОПУСКАЕТСЯ К ЗАЩИТЕ |
|  | Председатель ПЦК специальности 09.02.07 Информационные системы и программирование |
|  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_А.И. Глускер |
|  | «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2023 г. |

|  |
| --- |
| **ДИПЛОМНЫЙ ПРОЕКТ** |
|  |
| на тему: Разработка электронного образовательного ресурса для использования в образовательной организации (на примере темы «Молекулярная физика. Термодинамика» дисциплины «Физика») |
|  |
| студентки группы 090207-9о-19/1  специальности 09.02.07 Информационные системы и программирование |
| Сарапулова Даниила Владиславовича |

|  |  |  |
| --- | --- | --- |
| Студента |  | Д.В. Сарапулов |
| Руководитель |  | Л.В. Салахова |
| Консультанты: |  |  |
| Нормоконтроль |  | И.Г. Дзюба |

|  |  |  |
| --- | --- | --- |
| Дата защиты «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2023 г. | | |
| Оценка:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | |
| Председатель ГЭК |  | П.Р. Сафиканов |

Москва

2023

**СОДЕРЖАНИЕ**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  | | | | | |
|  |  |  |  |  |
| Изм. | Лист | № докум. | Подп. | Дата |
| Разраб. | | Сарапулов Д.В. |  |  |  | Лит. | | | Лист | Листов |
| Пров. | | Салахова Л.В. |  |  |  |  |  | 2 |  |
|  | |  |  |  |  | | | | |
| Н. контр. | | Дзюба И.Г. |  |  |
| Утв. | |  |  |  |

[**ВВЕДЕНИЕ** 3](#_Toc136437859)

[**Глава 1. ТРЕБОВАНИЯ ЭЛЕКТРОННО ОБРАЗОВАТЕЛЬНОМУ РЕСУРСУ НА ОСНОВЕ ТЕОРИТИЧЕСКИХ АСПЕКТАХ РАЗРАБОТКИ ВЕБ-ИНТЕРФЕЙСОВ** 6](#_Toc136437860)

[**Глава 2. РЕСУРСЫ И ИНСТУМЕНТЫ ИСПОЛЬЗУЕМЫЕ ДЛЯ РАЗРАБОТКИ ЭЛЕКТРОННО ОБРАЗОВАТЕЛЬНОГО РЕСУРСА** 8](#_Toc136437861)

[2.1 Языки программирования 8](#_Toc136437862)

[2.2 Среды разработки 10](#_Toc136437863)

[2.3 Разработка и прототипирование дизайна 11](#_Toc136437864)

[2.4 Базы данных 11](#_Toc136437865)

[2.5 Фреймворки 12](#_Toc136437866)

[2.5.1 Фреймворк для серверной части (BackEnd, RESTAPI) 12](#_Toc136437867)

[2.5.2 Фреймворк для клиентской части (FrontEnd) 13](#_Toc136437868)

[2.6 Интегрированная среда разработки 13](#_Toc136437869)

[2.7 Проектировщик ER–диаграмм 14](#_Toc136437870)

[**Глава 3. РАЗРАБОТКА ЭЛЕКТРОННОГО ОБРАЗОВАТЕЛЬНОГО РЕСУРСА** 15](#_Toc136437871)

[3.1 Разработка названия электронного образовательного ресурса 15](#_Toc136437872)

[3.2 Разработка логотипа электронного образовательного ресурса 15](#_Toc136437873)

[3.3 Разработка дизайна электронного образовательного ресурса 17](#_Toc136437874)

[3.4 Проектировка базы данных 20](#_Toc136437875)

## **ВВЕДЕНИЕ**

В настоящее время наличие электронных образовательных ресурсов (ЭОР) является неотъемлемой частью обучения в различных образовательных учреждениях, а также в иных сферах деятельности. Электронные образовательные ресурсы входят в обиход образовательных учреждений ввиду их доступности с любого устройства, с любой точки планеты.

Электронные образовательные ресурсы легко заменяют книги и учебники, а также позволяют лучше донести информацию до пользователя из-за интерактивности ресурсов, информация легче усваивается пользователем в более непринужденной форме.

Сфера образования нуждается в такой же оцифровке, легкодоступности и мобильности как другие сферы жизни человека. Данная ситуация доказывается потребностью целевой аудитории, которая большую часть своего времени проводит в смартфонах и персональных компьютерах, в следствии которой уже существует множество различных электронных образовательных ресурсов, направленных как и на общую тематику, так и на узкую (к примеру электронный образовательный ресурс в котором изучают только «Математику»).

Современная молодежь в большинстве своем мало осведомлена в области физики что создает ошибочное восприятие о данном направлении, и его деятельности, поэтому возникает малая вовлеченность в нужные и фундаментальные профессии.

В связи с этим актуальна разработка узконаправленного электронного образовательного ресурса, который поможет учебному заведению продемонстрировать и ознакомить обучающихся с дисциплиной «Физика», а также некоторыми ее направлениями.

Целю этой работы является полная разработка электронного образовательного ресурса для использования в образовательной организации (на примере темы «Молекулярная физика. Термодинамика» дисциплины «Физика»).

**Были поставлены задачи:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  | Лист |
|  |  |  |  |  |  |
| Изм. | Лист | № докум. | Подпись | Дата |

1. Cпроектировать и разработать базы данных.
2. Разработать серверный компонент REST API.
3. Внедрить функцию регистрации и авторизации пользователя.
4. Реализовать шифрование данных пользователя (пароль).
5. Реализовать выдачу данных по токену пользователя.
6. Разработать интерфейс веб-приложения (фронтенд spa).
7. Реализовать динамический фон.
8. Реализовать замедленное моделирование теплового движения молекул в зависимости от давления, температуры, объема.
9. Реализовать моделирование адиабатических, изохорных, изобарных, изотермических процессов.
10. Провести расчет экономической эффективности разработки.

**Гипотезы выпускной квалификационной работы:**

1. Предполагается, что целевая аудитория повысит уровень знаний и осведомленности в дисциплине «Физика».

2. Предполагаются положительные отзывы о разработанном электронном образовательным ресурсом.

3. Предполагается, что электронный образовательный ресурс будет использоваться учебным заведением для обучения учеников и повышения осведомленности.

# **Глава 1. ТРЕБОВАНИЯ ЭЛЕКТРОННО ОБРАЗОВАТЕЛЬНОМУ РЕСУРСУ НА ОСНОВЕ ТЕОРИТИЧЕСКИХ АСПЕКТАХ РАЗРАБОТКИ ВЕБ-ИНТЕРФЕЙСОВ**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  | Лист |
|  |  |  |  |  |  |
| Изм. | Лист | № докум. | Подпись | Дата |

* 1. В соответствии с ГОСТ Р 52653-2006 Электронный образовательный ресурс должен включать образовательный контент, метаданные и программные компоненты.

Образовательный контент – это структурированный предметный контент, используемый в образовательном процессе, информационно-релевантный контент электронный образовательного ресурса. Программные компоненты обеспечивают представление элементов контента пользователю в определенных комбинациях, а также обеспечивают интерактивный способ работы с контентом.

Метаданные – структурированные данные, задача которых описывать характеристики электронного образовательного ресурса.

* 1. Ресурс должен быть интерактивным и модифицируемым.

Интерактивность – способность компонентов (в данном случае электронно-образовательного ресурса) активно реагировать на те или иные действия со стороны пользователя (к примеру, изображение при наведении на нее начинает двигаться или делать любые иные действия).

Модифицируемость – свойство программного обеспечения, характеризующая какие усилия необходимо приложить для улучшения и исправления программного обеспечения, в связи с найденными дефектами или подстраиванию под новые стандарты в сфере.

* 1. Ресурс должен соответствовать современному представлению и правилам дизайна.

Следует придерживаться следующих правил в дизайне:

1. Единство стиля на всех страницах
2. Удобство работы с навигацией сайта
3. Любой текстовый материал должен представляться пользователю максимально лаконично

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  | Лист |
|  |  |  |  |  |  |
| Изм. | Лист | № докум. | Подпись | Дата |

1. Интерфейс должен быть юзерфрендли
2. Страницы не должны бать переполнены информацией если того не требуется
3. Любая аудио и медиа информация на сайте должна быть обоснована

Юзерфрендли – современное понятие в сфере разработки обозначающее легкость и простоту использования интерфейса или функций пользователем.

* 1. Ресурс должен быть понятен разработчикам с технической точки зрения.

В связи с тем что рынок профессий в сфере разработки наполняется новыми кадрами, для любой компании свойственна «текучка» кадров и распределение большего количества человек для разработки и поддержания одного проекта. В связи с этим должен быть выбран более распространённый язык программирования для работы в команде.

# **Глава 2. РЕСУРСЫ И ИНСТУМЕНТЫ ИСПОЛЬЗУЕМЫЕ ДЛЯ РАЗРАБОТКИ ЭЛЕКТРОННО ОБРАЗОВАТЕЛЬНОГО РЕСУРСА**

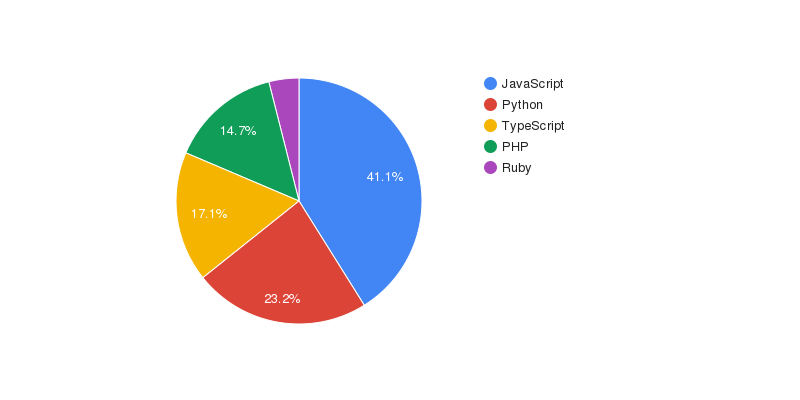
## 2.1 Языки программирования

В настоящее время существует большое множество языков программирования каждый язык по-своему уникален, но не все они «гибкие» и универсальные. Поэтому стоит подбирать язык программирования, который легко освоить и максимально может закрыть потребности при разработке.

Очень часто такие языки являются самыми популярными в мире, список таких языков публикуется на сайте для разработчиков GitHub основанным в 2008 году, использующий и развивающий систему GIT каждый год публикуется статистика самых используемых языков, популярными языками программирования в сфере веб разработки являются (Рисунок 1, Рисунок 2):

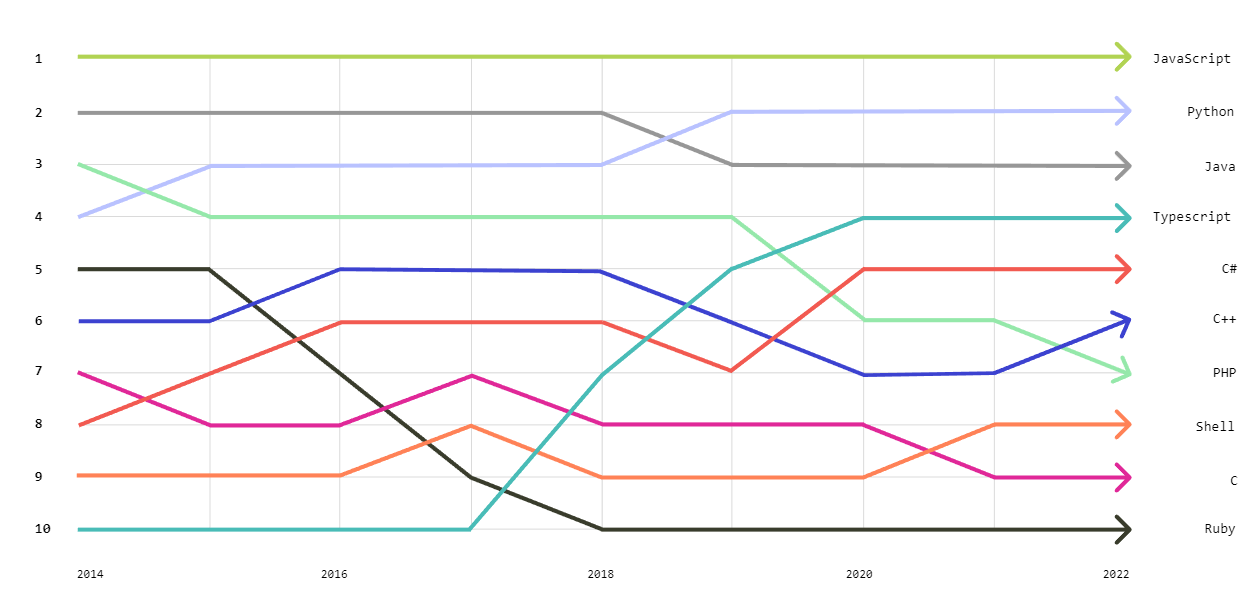
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  | Лист |
|  |  |  |  |  |  |
| Изм. | Лист | № докум. | Подпись | Дата |

1. JavaScript
2. Python
3. TypeScript
4. PHP
5. Ruby



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  | Лист |
|  |  |  |  |  |  |
| Изм. | Лист | № докум. | Подпись | Дата |
|  |  |  |  |  |  | Лист |
|  |  |  |  |  |  |
| Изм. | Лист | № докум. | Подпись | Дата |

Рисунок 1 – диаграмма часто используемых языков веб-программирования

Рисунок 2 – График часто используемых языков в сервисе GitHub

Исходя из этих данных лучшим языком для разработки электронного образовательного ресурса является JavaScript.

JavaScript может использоваться для разработки, как и клиента, так и серверного компонента (RESTAPI).

## 2.2 Среды разработки

Среда разработки (IDE) выбирается самим разработчиком исходя из собственных нужд. Существуют универсальные среды разработки, которые поддерживают некоторое множество языков или все. Так же существуют среды разработки под определенные языки такие как PyCharm, который нацелен на работу только с Python. Для данной задачи лучше всего подходит Visual Studio и Visual Studio Code, так как они имеют поддержку огромного количества языков программирования, что позволяет пользоваться одним редактором для разных задач, и разных языков. В Visual Studio и Visual Studio Code так же есть встроенные подсказки синтаксиса, что упрощает изучение языка и разработку проекта. В данных продуктах есть интеграция с Git и GitHub.

2.3 Система контроля версий

В разработке любых проектов разработчики используют систему контроля версий, такие как:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  | Лист |
|  |  |  |  |  |  |
| Изм. | Лист | № докум. | Подпись | Дата |

1. Git
2. Mercurial
3. Subversion
4. Perforce

Данные системы контроля версий записывают и сохраняют все изменения в файлах проекта, позволяют делать несколько различных ответвлений проекта, чтобы в них можно было вносить изменения без ущерба основной версии проекта, а после успешного обновления проекта на другом его ответвлении можно «слиять» вторичную и основную ветви проектов. Так же системы контроля версий позволяют вернуться к предыдущей версии проекта.

Самой популярной системой контроля версий является Git выпущенный в 2005 году, который в свою очередь имеет интеграцию с различными веб-сервисами для хостинга различных проектов и их совместной разработки. Такие веб-сервисы обеспечивают доступ к проектам с любого устройства, и из любой точки планеты.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  | Лист |
|  |  |  |  |  |  |
| Изм. | Лист | № докум. | Подпись | Дата |

Существует множество таких веб-сервисов, все они работают и выглядят по одному принципу, самым же популярным из них является GitHub.

Исходя из этого лучшим вариантом является система контроля версий Git, и веб-сервис GitHub.

## 2.3 Разработка и прототипирование дизайна

Для облегчения разработки дизайна были созданы специальные прило­жения и сервисы, с помощью которых веб-дизайнеры делают представление сайта по техническому заданию. В дальнейшем разработчики при создании используют эти представления для создания клиентской части сайта.

В данном направлении разработки так же существует много сервисов и приложений, например:

1. Figma
2. Blocs Website Builder
3. InVision
4. Turbologo
5. Zeplin

Для задач этой работы лучше всего подходит Figma.

## 2.4 Базы данных

Существует два типа баз данных: реляционные и нереляционные, их основное отличие в том что в реляционных базах данных все данные хранятся в виде столбцов и строк. В нереляционных базах данных в свою очередь информация храниться в определенных типах, таких как: JSON, графы, ключи значений.

Для задач данной работы подходят реляционные базы данных, популярными представителями являются:

1. MySQL
2. Oracle RDBMS
3. PostgreSQL
4. Microsoft SQL Server

Из этих баз данных лучше всего подходит MySQL, из-за большой популярности, простоты освоения и использования, так же будет использоваться ответвление MariaDB.

## 2.5 Фреймворки

Фреймворк – специальное «дополнение» к языку программирования, которое определяет структуру проекта, предоставляет некоторые инструменты, для разработки которых нет в «чистом» языке. Так же фреймворки могут объединять несколько функций в одну для упрощения и удобства разработки.

### Почти для каждого языка программирования существуют фреймворки, ранее для данной работы был выбран язык программирования JavaScript. Из-за того, что для этой работы был выбран JavaScript, для написания клиентской части и серверной, фреймворки для задач используются разные.

### 2.5.1 Фреймворк для серверной части (BackEnd, RESTAPI)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  | Лист |
|  |  |  |  |  |  |
| Изм. | Лист | № докум. | Подпись | Дата |

Для разработки серверного компонента зачастую используют следующие фреймворки:

1. Express
2. Spring Boot
3. Node.js
4. GraphQL

Для разработки электронного образовательного ресурса больше всего подходит Node.js, эта система предназначена для асинхронной обработки, поэтому серверный компонент лучше работает с большим количеством пользователей. Этот фреймворк использует синтаксис JavaScript, упрощается его освоение и использования. Node.js имеет полную поддержку JSON формата файлов, что пригодиться в дальнейшей разработке.

### 

### 2.5.2 Фреймворк для клиентской части (FrontEnd)

1. React JS
2. Vue JS
3. Angular
4. Ember JS
5. Preact

Лучшим выбором является Vue.js, потому что имеет большую скорость отрисовки чем другие, имеет реактивную модель, которая обновляет представление при изменении данных, и при установке имеет встроенные критически важные компоненты.

## 2.6 Интегрированная среда разработки

Интегрированная среда разработки – комплекс программных средств для разработки программного обеспечения. Такие среды работают по принципу «все в одном», то есть одна программа содержит в себе одно или несколько средств для развертывания баз данных, запуска локальных серверов. Так же такие среды могут содержать в себе встроенные редакторы кода.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  | Лист |
|  |  |  |  |  |  |
| Изм. | Лист | № докум. | Подпись | Дата |

1. Laragon
2. Open Server Panel
3. WampServer
4. XAMPP
5. MAMP

Для данной разработки электронного образовательного ресурса подходит OpenServerPanel – интегрированная среда разработки имеет много­функ­циональность, поддержку большого количества языков интерфейса, просмотр логов в реальном времени, и конечно же быстроту запуска.

## 

## 2.7 Проектировщик ER–диаграмм

ER–диаграммы применяются для проектировки структур реляционных баз данных: как будут взаимодействовать таблицы между собой, какой компонент будет первичным ключом по которому можно однозначно определить каждую строку в таблице. Есть множество программ, при помощи которых можно создать ER–даиграммы.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  | Лист |
|  |  |  |  |  |  |
| Изм. | Лист | № докум. | Подпись | Дата |

1. Er assistant;
2. Software Ideas Modeler;
3. Edraw.

Самой подходящей является Er assistant – она наиболее приспособлена для создания ER–даиграмм баз данных, имеет простой не нагруженный интерфейс, в котором легко разобраться.

## 2.8 Архитектура REST API

API – Application Programming Interface. Это программный интерфейс, который позволяет связывать между собой различные приложения. На примере этой дипломной работы API будет связывать серверный компонент и базу данных, что бы отправляя запрос на серверный компонент можно было получить данные из базы данных.

REST – это архитектура API, которая работает с помощью протокола HTTP, и имеет возможность предоставлять данные с сервера.

## 2.9 SPA SPA (Single Page Application) – дословно переводя это одностраничное приложение, вся суть данной архитектуры заключается в том что веб–приложение работает на одной странице html, отрисовывая нужные компоненты. Данная архитектура обладает рядом преимуществ в отличие от переадресации пользователя на отдельные html документы:

1) Быстродействие – не надо при каждом запросе загружать страницу, все уже заранее отрисованно.

2) Гибкость – зачастую применяется такой паттерн программирования при котором существует главный родительский компонент, и на нем уже размещаются дочерние, к примеру блок с навигацией это отдельный компонент, блок с контентом это тоже отдельный компонент и т.д., эти компоненты легко настраивать по отдельности.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  | Лист |
|  |  |  |  |  |  |
| Изм. | Лист | № докум. | Подпись | Дата |

# **Глава 3. РАЗРАБОТКА ЭЛЕКТРОННОГО ОБРАЗОВАТЕЛЬНОГО РЕСУРСА**

## 3.1 Разработка названия электронного образовательного ресурса

Имя продукта одна из важнейших деталей, так как имя будет напрямую будет ассоциироваться с продуктом и с его производителем.

Так же если имя уникально и необычно, то оно лучше запоминается у потребителя.

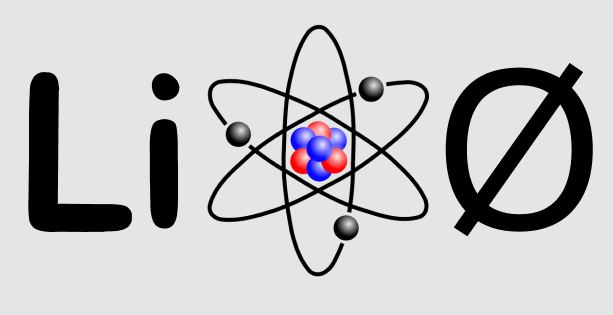
Отталкиваясь от темы выпускной квалификационной работы, за основную идею имени был выбран оксида лития, это лучше всего ассоциирует электронный образовательный ресурс для использования в образовательной организации (на примере темы «Молекулярная физика. Термодинамика» дисциплины «Физика») с наукой. Для комфортного и современного произношения, название было переведено на английский язык: «Lithium oxide».

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  | Лист |
|  |  |  |  |  |  |
| Изм. | Лист | № докум. | Подпись | Дата |

## 3.2 Разработка логотипа электронного образовательного ресурса

Логотип является неотъемлемой частью любой компании и продукта, логотип крепко ассоциируется с продуктом или компанией. Благодаря ему можно выделиться на рынке среди конкурентов. Логотип обеспечивает визуальный контакт с потребителем. Если логотип индивидуален и не похож на другие, то это показывает отношение компании к своему продукту, и показывает пользователю, что производитель/разработчик заботятся о своих клиентах. Логотип – это прямое продолжение имени продукта или компании.

В первую версию логотипа (Рисунок 3) легло имя оксида, атом физики, и символ пустого множества (Рисунок 4).

Рисунок 3 – первая версия логотипа

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  | Лист |
|  |  |  |  |  |  |
| Изм. | Лист | № докум. | Подпись | Дата |

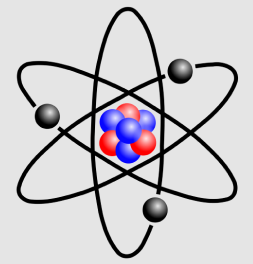
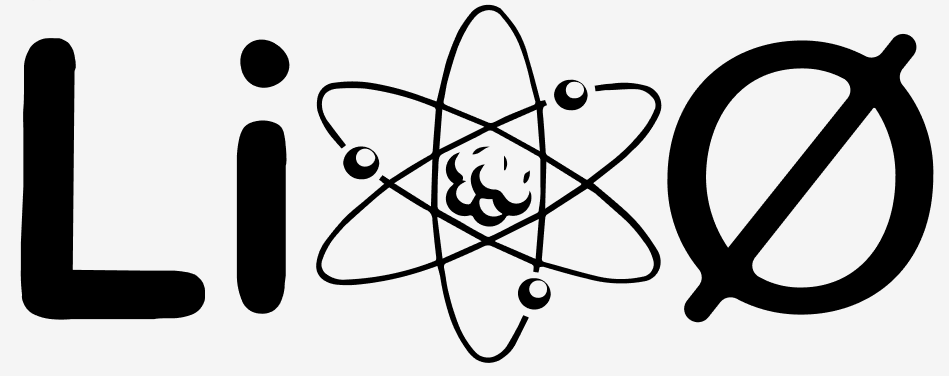


Рисунок 4 – атом физики

После более подробного изучения современных логотипов электронных образовательных ресурсов, была создана новая версия логотипа, который был изменен и осовременен: были убраны острые углы, атом физики был сделан черно-белым, из-за чего весь логотип стал полностью черным (Рисунок 5)



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  | Лист |
|  |  |  |  |  |  |
| Изм. | Лист | № докум. | Подпись | Дата |

Рисунок 5 – обновленная версия логотипа

3.3 Разработка рекламной продукции

Рекламная про

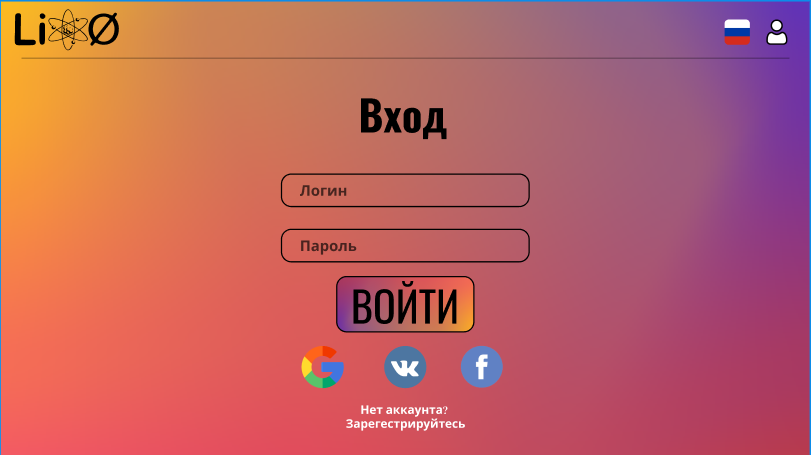
На основе новой версии логотипа была разработана некоторая рекламная продукция:

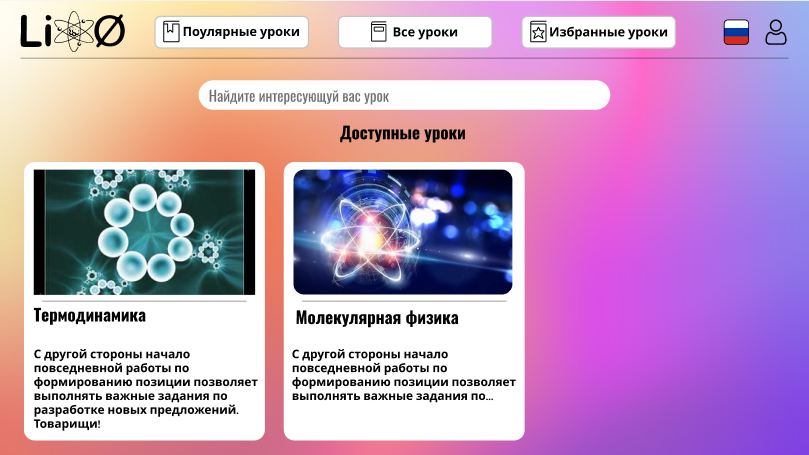
## 3.3 Разработка дизайна электронного образовательного ресурса

Для разработки дизайна интерфейса будет использоваться ранее выбранная Figma. Работа над дизайном и прототипом были созданы следующие страницы электронного образовательного ресурса:

1. Страница на которой реализована авторизация (Рисунок 6)
2. Главная страница, на которой можно выбрать доступные уроки (Рисунок 7)
3. Страница урока по теме «Термодинамика» (Рисунок 8)
4. Страница урока по теме «Молекулярная физика» (в рамках работы над прототипом, функционал страницы не разрабатывался)

Ограниченный по функционалу прототип сайта нужен для понимания дальнейшей разработки электронного образовательного ресурса .

Рисунок 6 – страница авторизации



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  | Лист |
|  |  |  |  |  |  |
| Изм. | Лист | № докум. | Подпись | Дата |

Рисунок 7 – страница доступных уроков

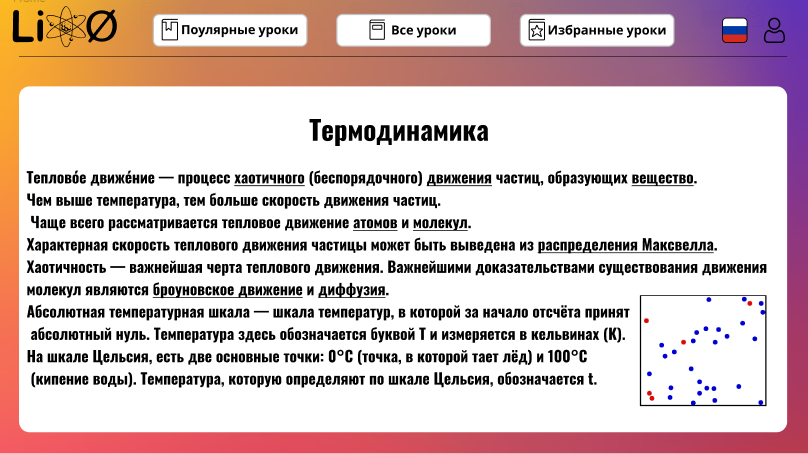


Рисунок 8 – Страница урока по теме «Термодинамика»

Дизайн был вдохновлен несколькими ресурсами:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  | Лист |
|  |  |  |  |  |  |
| Изм. | Лист | № докум. | Подпись | Дата |

<https://duolingo.com> (Рисунок 9)

https://stepik.org (Рисунок 10)

<https://razoom.mgutm.ru> (Рисунок 11)



Рисунок 9 – Электронный образовательный ресурс “duolingo”

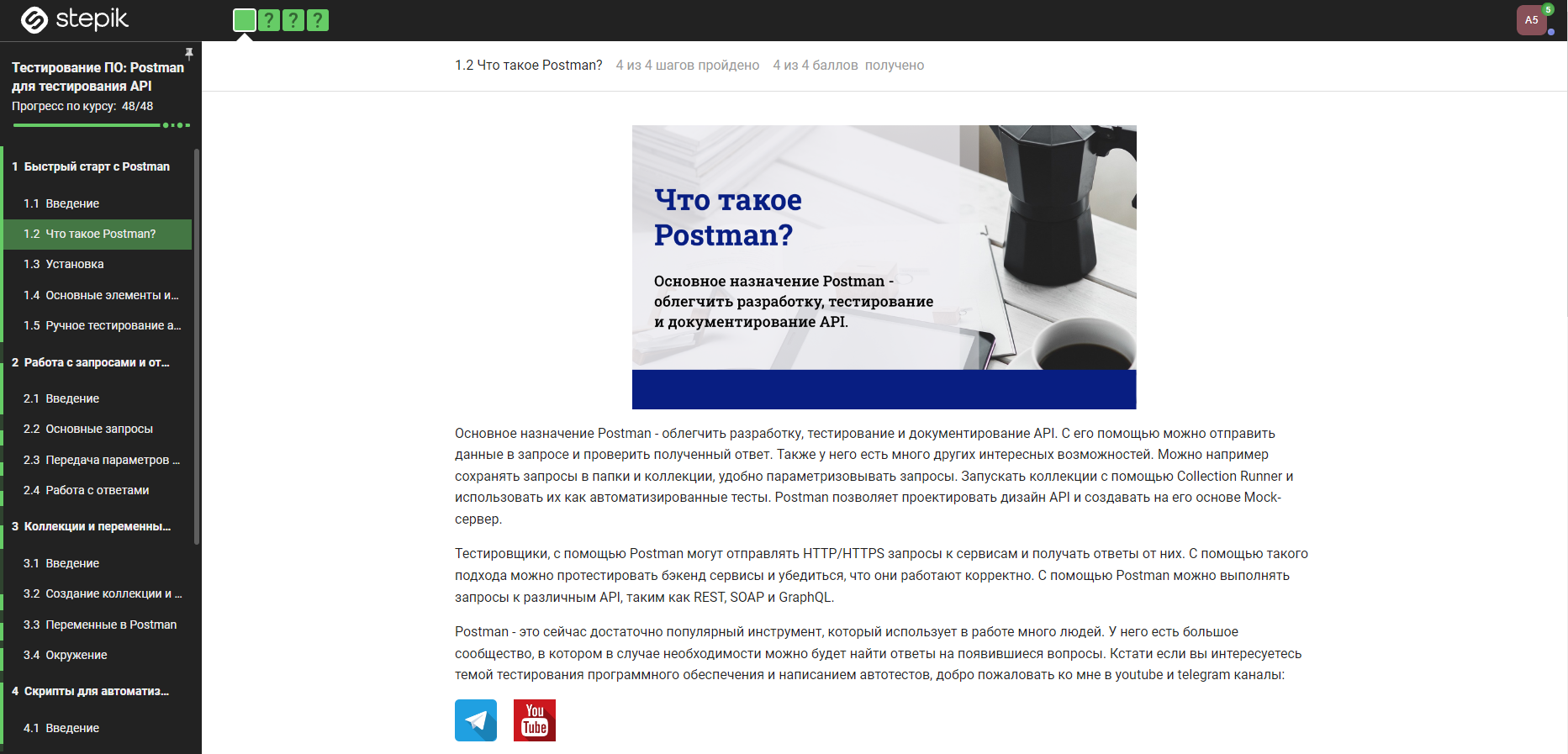
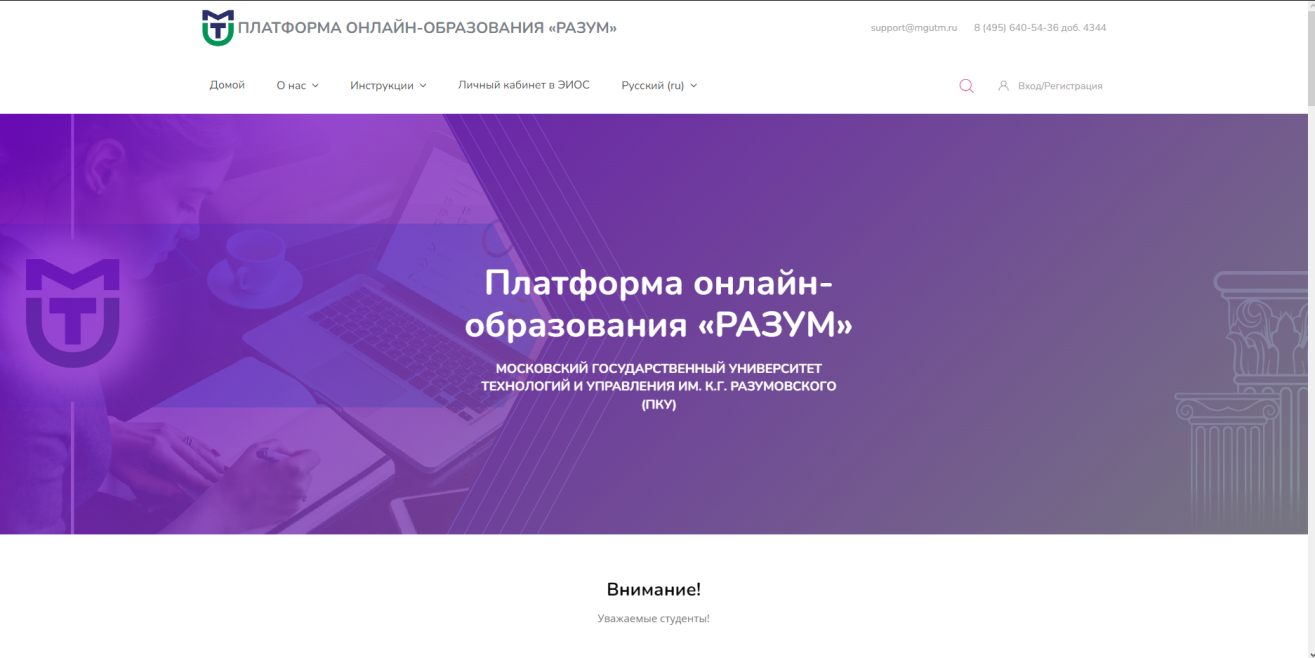


Рисунок 10 – Электронный образовательный ресурс “stepik”



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  | Лист |
|  |  |  |  |  |  |
| Изм. | Лист | № докум. | Подпись | Дата |

Рисунок 11 – Платформа онлайн-образования «РАЗУМ»

## 3.4 Проектировка базы данных

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  | Лист |
|  |  |  |  |  |  |
| Изм. | Лист | № докум. | Подпись | Дата |

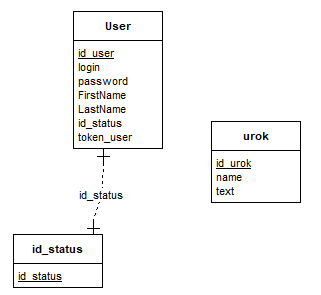
Для создания структуры будущей базы данных была сделана ER–диаграмма (Рисунок 12).

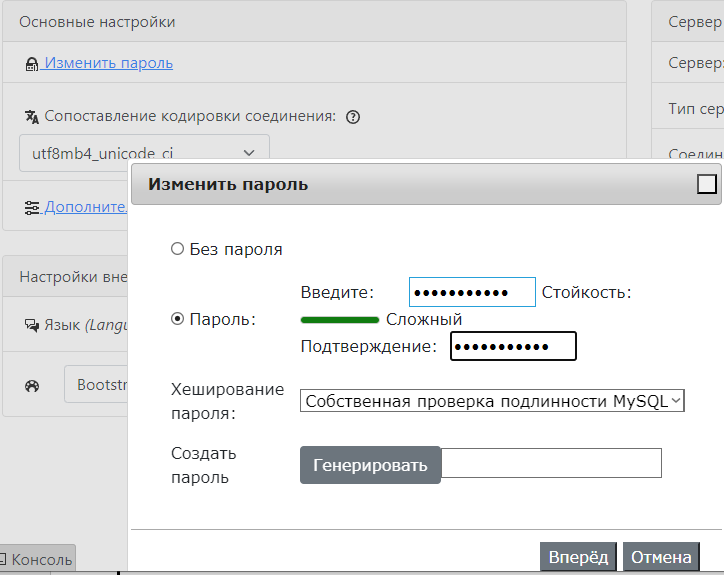
Рисунок 12 – ER–диаграмма базы данных

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  | Лист |
|  |  |  |  |  |  |
| Изм. | Лист | № докум. | Подпись | Дата |

Так как в разрабатываемом электронном образовательном ресурсе все уроки доступны для всех зарегистрированных пользователей, таблицы «User» и «Urok» никак не связаны. У каждого пользователя есть статус: пользователя или администратора, который может создавать новые уроки.

## 3.5 Создание базы данных

Для создания базы данных будет использоваться компонент OpenServerPanel под названием PhpMyAdmin. После запуска OpenServerPanel и открытия PhpMyAdmin, система предлагает нам авторизоваться, по умолчанию стоит логин «root», а пароля нет. В перспективе разработки проекта это небезопасно, поэтому в главном меню нужно сменить логин и пароль для доступа к базе данных (Рисунок 13).

Рисунок 13 – смена пароля для доступа к базе данных

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  | Лист |
|  |  |  |  |  |  |
| Изм. | Лист | № докум. | Подпись | Дата |

После этого можно приступать к созданию самой базы данных по ER–диаграмме, сделанной ранее (Рисунок 12).

Были созданы таблицы: «user» (Рисунок 14), «urok» (Рисунок 15), «id\_status» (Рисунок 16).

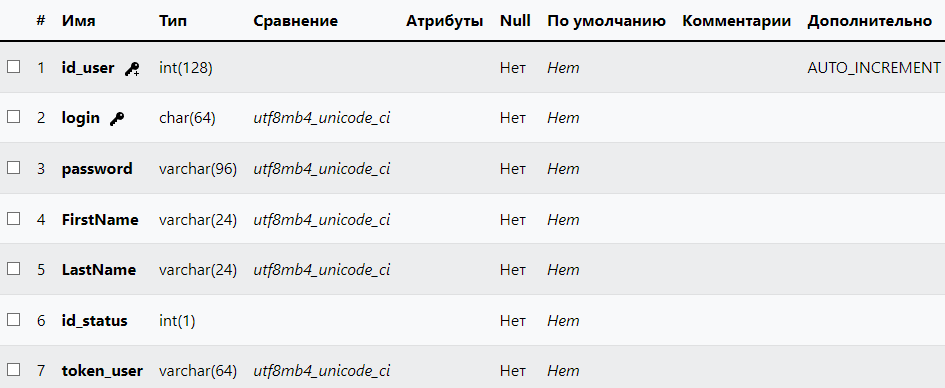


Рисунок 14 – структура таблицы “user”

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  | Лист |
|  |  |  |  |  |  |
| Изм. | Лист | № докум. | Подпись | Дата |

Рисунок 15 – структура таблицы “urok”

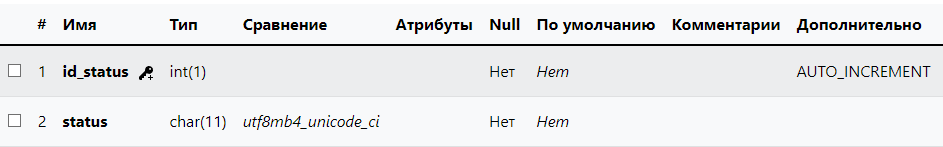
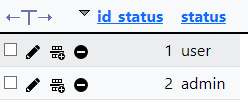


Рисунок 16 – структура таблицы “id\_status”

В таблице «user» (Рисунок 14) присутствует строка «login», ей присвоен уникальный ключ, который не дает создать повторяющиеся логины пользователей. В базе данных создано разделение прав пользователей, данная функция реализована в таблице «id\_user» (Рисунок 16), вручную добавлены две строки, которые отвечают за права пользователей (Рисунок 17).



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  | Лист |
|  |  |  |  |  |  |
| Изм. | Лист | № докум. | Подпись | Дата |

Рисунок 17 – Разграничение прав в базе данных

По умолчанию всем пользователям присваивается значение статуса 1, означает – обычный пользователь и не имеет прав на редактирование контента.

Функции администратора пока не реализованы в проекте, и все пользователи являются простыми читателями, но в будущем планируется добавление функционала администратора.

## 3.6 Разработка серверного компонента REST API

После создания базы данных идет разработка серверного компонента с архитектурой REST API, для реализации потребуется два файла, в одном будут прописаны URL для обращения к серверу, а так же будет производиться парсинг данных (преобразование данных в формат JSON). Второй же файл будет содержать сами методы получения данных из базы данных в которых будут указаны SQL запросы.

Разработка серверного компонента была начата с объявления констант подключения к базе данных (Рисунок 18)



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  | Лист |
|  |  |  |  |  |  |
| Изм. | Лист | № докум. | Подпись | Дата |

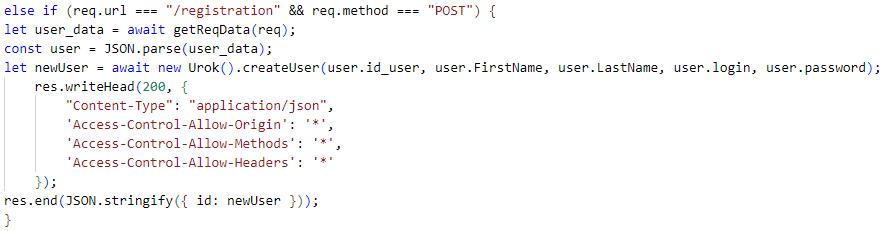
Рисунок 18 – константы подключения к базе данных

Такое решение позволяет избежать нагромождение кода, и в случае чего, данные константы можно спрятать в части DOM’a проекта недоступную обычному пользователю в целях безопасности.

Так же была создана константа «pool» она практически идентична с константой подключения, однако имеет ограничение на подключение, сделано это специально, что бы пользователь мог запрашивать определенное количество раз одни и те же данные.

Далее создается метод, который будет отправлять SQL запрос на создание новой записи в таблицу «user» в базу данных (Рисунок 19), так же что бы этот метод корректно работал нужно во втором файле, в котором парсятся данные, создать. Для дополнительной безопасности данных, пароли хэшируются (шифруются).



Рисунок – 19 метод регистрации нового пользователя

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  | Лист |
|  |  |  |  |  |  |
| Изм. | Лист | № докум. | Подпись | Дата |

Рисунок – 20 путь для обращения к серверу и парсинг данных при регистрации

Теперь надо создать методы авторизации пользователя, которые при успешной авторизации будут генерировать токены пользователей и записывать их базу данных (Рисунок 21, Рисунок22)

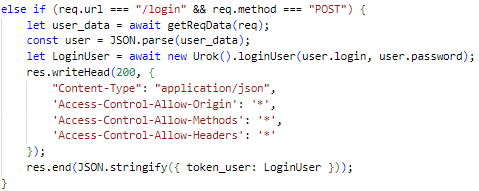
Рисунок 22 – метод авторизации пользователя

Рисунок 23 – путь для обращения к серверу и парсинг данных при авторизации

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  | Лист |
|  |  |  |  |  |  |
| Изм. | Лист | № докум. | Подпись | Дата |

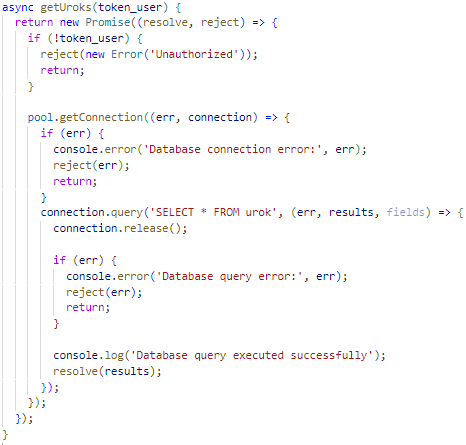
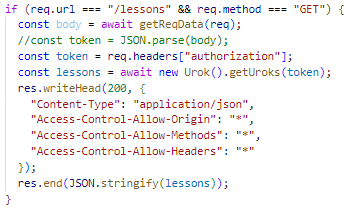
Далее были созданы методы для получения уроков из таблицы «urok» (Рисунок 24, Рисунок 25)

Рисунок 24 – метод получения уроков



Рисуонк 25 – путь для обращения к серверу и парсинг данных при получении уроков

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  | Лист |
|  |  |  |  |  |  |
| Изм. | Лист | № докум. | Подпись | Дата |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  | Лист |
|  |  |  |  |  |  |
| Изм. | Лист | № докум. | Подпись | Дата |