Департамент образования и науки города Москвы

ГАПОУ Колледж предпринимательства №11

Центр информационно-коммуникационных технологий

**Аналитический отчёт**

по дисциплине: «Исследовательская и проектная деятельность»

на тему: Информационный сайт: «Проекты студентов ГАПОУ КП 11 Центра ИКТ по дисциплине «Исследовательская и проектная деятельность»»

|  |  |
| --- | --- |
| Выполнили:  студенты группы ИСиП-33  Комаров Алексей – менеджер  Дорина Валентина – зам. менеджера  Красюк Евгений | Проверил:  Преподаватель  Е.Ю. Ильина |

Москва, 2023 г.

# **Введение**

В центре информационно-коммуникационных технологий колледжа предпринимательства № 11 по дисциплине «Исследовательская и проектная деятельность» создается много интересных проектов, которые можно публиковать, что позволит расширить аудиторию, заинтересованную в студенческих проектах. Это поможет студентам понять на что ориентироваться при выборе темы, а также что из себя должен представлять проект. Кроме этого, сайт предоставит другим пользователям возможность увидеть, осуществляемые студентами колледжа, проекты.

Актуальность выбранной темы: проектная группа хочет улучшить свои знания в веб-сфере, создав сайт, на котором будут публиковаться проекты. При работе над проектом команда планирует улучшить такие навыки, как ответственность, планирование, креативность.

Цель проекта - разработать сайт для хранения проектов по дисциплине «Исследовательская и проектная деятельность»

Задачи:

1. Выбрать тему исследования
2. Собрать участников в команду
3. Распределить обязанности в команде
4. Собрать информацию по сайту
5. Продумать дизайн сайта
6. Создать макет
7. Продумать что будет публиковаться на сайте
8. Определиться с функциями сайта
9. Ознакомиться со способами реализации функций сайта
10. Разработать сайт
11. Протестировать обучающий сайт
12. Написать рекомендации и выводы
13. Защитить аналитический отчет

Объект исследования: информационный сайт.

Предмет исследования: адаптированное информационное пространство для хранения проектов по дисциплине «Исследовательская и проектная деятельность».

Субъект исследования: проектная группа

Методы исследования, которые будут применяться в работе: метод экспертных оценок, кабинетный, разведочный, моделирование.

# **Глава 1. Натуральное описание**

В этой главе команда опишет процесс выполнения аналитического отчета.

**13.01.2023 – 27.01.2023**

Создали исследовательскую группу: Комаров Алексей, Дорина Валентина, Красюк Евгений.

Выбрали тему проекта.

Определили роли в команде.

Определились с целями и задачами проекта.

Написали главу аналитического отчета – введение.

**27.01.2023 – 10.02.2023**

Валентина нашла источник дополнительной информации:

https://www.youtube.com/watch?v=GQ\_pTmcXNrQ

Ознакомилась с полученной информацией и приступила к разработке сайта.

Алексей приступил к описанию разработки информационного сайта нашел источники дополнительной информации и ознакомился с полученной информацией:

https://habr.com/ru/company/ruvds/blog/428077/

https://habr.com/ru/company/ruvds/blog/422893/

Евгений нашел источник дополнительной информации:

<https://it-dev-journal.ru/articles/node-js-i-react-kak-napisat-fulstek-prilozhenie-polnoe-rukovodstvo>

**10.02.2023 - 18.02.2023**

Валентина продолжила разработку информационного сайта.

Алексей продолжил описание разработки информационного сайта и добавил вспомогательные картинки (Рисунок 1, Рисунок 2, Рисунок 3, Рисунок 4) и нашел источники дополнительной информации:

<https://www.geeksforgeeks.org/how-to-create-a-website-in-react-js/>

<https://www.techomoro.com/how-to-create-a-multi-page-website-with-react-in-5-minutes/>

Евгений нашел источники дополнительной информации:

<https://translated.turbopages.org/proxy_u/en-ru.ru.3f649025-63efd0ab-ff02599f-74722d776562/https/www.geeksforgeeks.org/create-relationship-in-mongodb/>

https://habr.com/ru/company/ruvds/blog/321104/

**18.02.2023 – 10.03.2023**

Валентина закончила разработку сайта.

Алексей продолжил описание разработки информационного сайта и добавил вспомогательные картинки (Рисунок 5, Рисунок 6, Рисунок 7, Рисунок 8, Рисунок 9, Рисунок 10).

Евгений более подробно описал назначение используемых библиотек, фреймворков и технологий, используемых в проекте.

# **Глава 2. Натуральное описание**

Разработка информационного сайта делится на два основных пункта – это Frontend и Backend.

Frontend-разработка – это создание пользовательского интерфейса на клиентской стороне веб‑сайта или приложения. Это всё, что видит пользователь, когда открывает веб-страницу, и с чем он взаимодействует: кнопки, баннеры и анимация.

Backend – это разработка бизнес-логики продукта (сайта или веб-приложения). Backend отвечает за взаимодействие пользователя с внутренними данными, которые потом отображает frontend. Попросту говоря, это то, что скрыто от глаз пользователя и происходит вне его браузера и компьютера.

**Используемые технологии**

Определимся с используемыми технологиями разработки. Для frontend:

* ReactJS – это фреймворк JavaScript с открытым кодом для создания внешних пользовательских интерфейсов. Фреймворки JavaScript это библиотеки программирования JavaScript, в которых есть предварительно написанный код, который упрощает разработку программного продукта, определяет структуру проекта и помогает удобно объединять в нём разные компоненты. ReactJS является одним из самых простых и используемых фреймворков, поэтому он был выбран для разработки сайта.
* Redux Toolkit – Пакет Redux Toolkit предназначен для стандартного способа написания логики Redux, которая предназначена для управления состоянием приложения. Содержит ряд инструментов, позволяющих значительно упростить передачу данных хранилища через контекст.
* React Hook Form – библиотека для обработки элементов ввода формы в экосистеме React. Она позволяет оптимизировать код за счет уменьшения количества повторных рендеров.
* React Router v6 – это стандартная библиотека для маршрутизации в React. Она обеспечивает навигацию между представлениями различных компонентов в приложении React, позволяет изменять URL браузера и синхронизировать пользовательский интерфейс с URL. Другими словами, данная библиотека позволяет перемещаться между страницами и компонентами на сайте без обновления страницы.

URL (Uniform Resource Locator – унифицированный указатель ресурса) – это адрес, который выдан уникальному ресурсу в интернете.

* React Markdown / Simple Editor – библиотека React Markdown будет отвечать за рендер информации на информационном сайте, в свою очередь Simple Editor будет отвечать за рендер редактора, в котором будут создавать статьи.
* Axios – это библиотека необходима, для того чтобы посетители информационного сайта смогли взаимодействовать с backend, то есть с технической частью сайта. Помимо этого Axios можно использовать для перехвата HTTP-запросов и ответов, а также включить защиту на стороне клиента от XSRF.

XSRF — это атака на веб-приложения, в результате которой вредоносное веб-приложение может повлиять на взаимодействие между клиентским браузером и веб-приложением, которое доверяет браузеру.

Технологии для Backend-разработки:

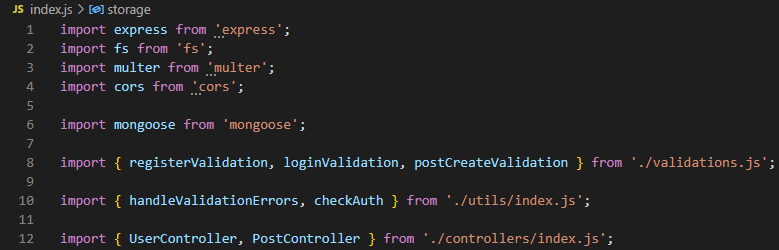
* NodeJS / ES6 – NodeJS - кроссплатформенная среда исполнения с открытым исходным кодом, которая позволяет разработчикам создавать всевозможные серверные инструменты и приложения используя язык JavaScript. Для функционирования NodeJS понадобится также установить и ECMAScript 2015 (или ES6), предназначенный для обеспечения совместимости веб - страниц в различных браузерах.
* Express + Validator – Express - это платформа веб-приложений, которая предоставляет простой API для создания веб-сайтов, веб-приложений и серверных частей. А validator является инструмент статического анализа кода, обеспечивающий проверку, если JavaScript исходный код содержит синтаксические ошибки.

API – Application Programming Interface, что значит программный интерфейс приложения. Содержит описание способов взаимодействия одной компьютерной программы с другими.

* MongoDB / Mongoose – для хранения всевозможной информации будет использоваться база данных MongoDB. Для подключения к MongoDB используем библиотеку Mongoose.
* JSON Web Token – это открытый стандарт для создания токенов доступа, основанный на формате JSON. Используется для передачи данных для аутентификации в клиент-серверных приложениях.
* Multer – библиотека, с помощью которой выполняется загрузка файлов.
* BCrypt – библиотека, используемая для защищенного хранения паролей.

**Разработка backend**

Первым делом создадим основной файл сайта index.js и импортируем необходимые библиотеки.



**Рисунок 1. Импорт библиотек**

Далее создадим файл package.json. Для того необходимо написать в консоль команду:

$ npm init

Npm - это менеджер пакетов, который управляет модулями и зависимостями проекта.

Node.js npm также избавляет разработчика от необходимости хранить все используемые сторонние модули в удаленных репозиториях. За это отвечает файл package.json. В npm файл package.json используется для описания модулей, от которых зависимо ваше приложение, и содержит ряд метаданных о приложении.

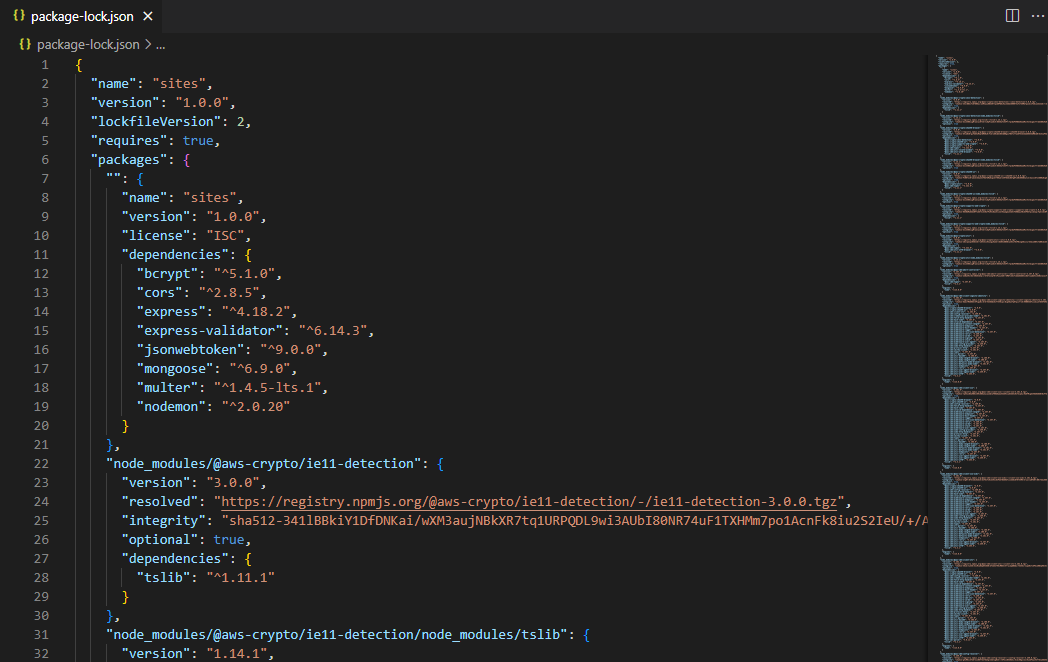


**Рисунок 2. Файл package.json**

Сразу после установки хотя бы одного Node.js npm модуля вместе с package.json будет создан файл package-lock.json, который будет обновляться каждый раз при добавлении новой зависимости.

В package-lock.json содержится описание состояния текущей иерархии используемых в приложении модулей: их точные версии и точные версии используемых ими модулей и так до конца. При развертывании Node.js приложения в новой среде наличие файла package-lock.json гарантирует, что будут установлены версии тех пакетов, которые использовались в разработке в последний раз.

Описание файла package-lock.json имеет более высокий приоритет в сравнении с описанием зависимостей package.json.



**Рисунок 3. Файл package-lock.json**

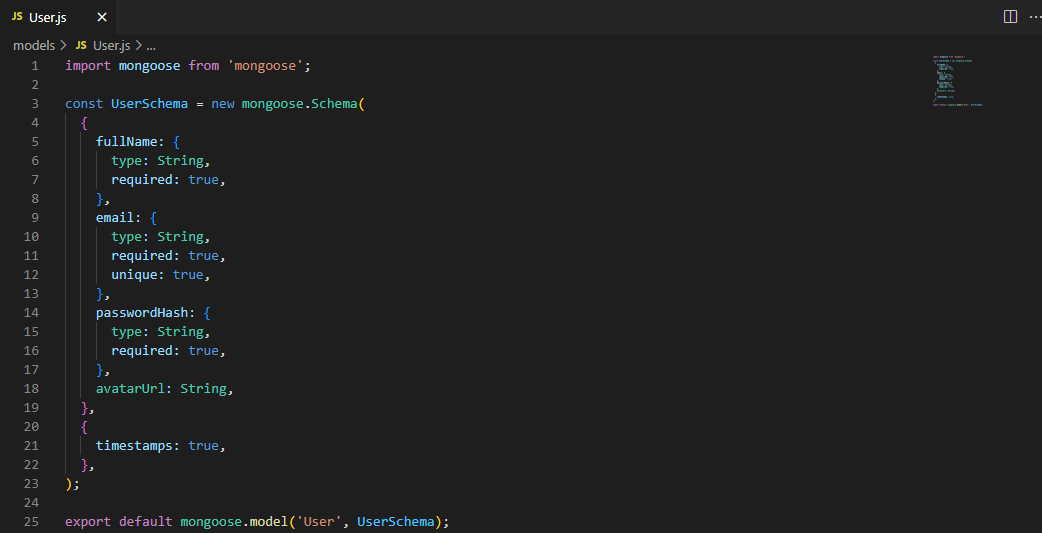
Далее реализуем авторизацию на информационном сайте. Для того, чтобы проверялась корректность введенных логина и пароля, понадобится библиотека express validator. Дальше создаем файл validations-js, в котором создаем переменные. Переменные будут обрабатывать введенные данные.



**Рисунок 4. Файл validations.json**

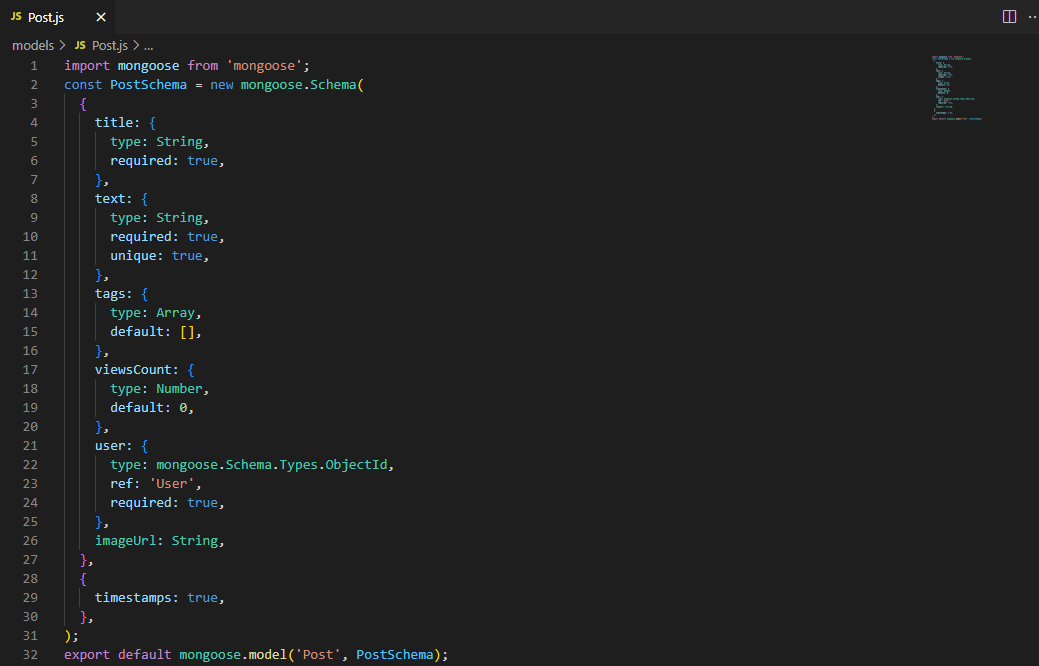
Приступим к созданию пользователя. Создадим директорию models, в которой будут находиться два файловых объекта: Post.js и User.js.

Файл User.js описывает конструкцию таблицы, содержащей в себе информацию о пользователях, зарегистрированных в системе.

****

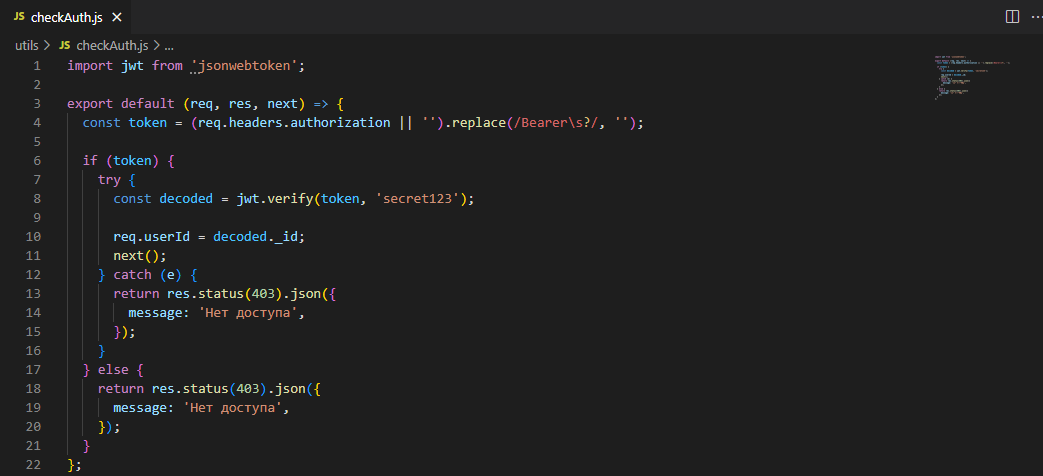
**Рисунок 5. Файл User.js**

Файл Post.js представляет собой структурную модель, описывающую основные элементы и атрибуты статей, размещаемых на информационном сайте.

****

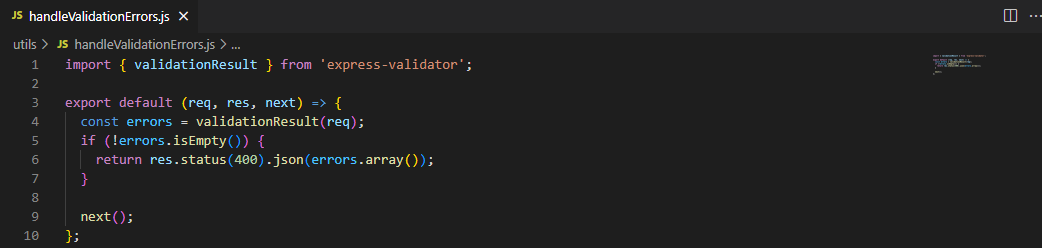
**Рисунок 6. Файл Post.js**

Далее создадим папку utils, в которую поместим функции. Первой такой функцией станет файл checkAuth.js, который будет проводить проверку доступа пользователя к запросу на авторизацию на информационном сайте.

****

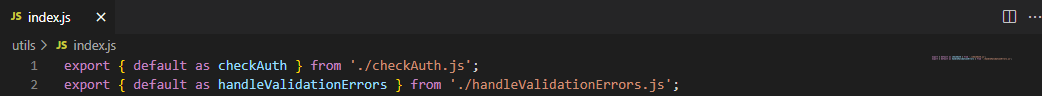
**Рисунок 7. Файл checkAuth.js**

Следующим файлом, содержащим функцию, станет handleValidationErrors.js. Эта функция позволит выдавать ошибку с более верным описанием при неверно введенных данных.

****

**Рисунок 8. Файл handleValidationErrors.js**

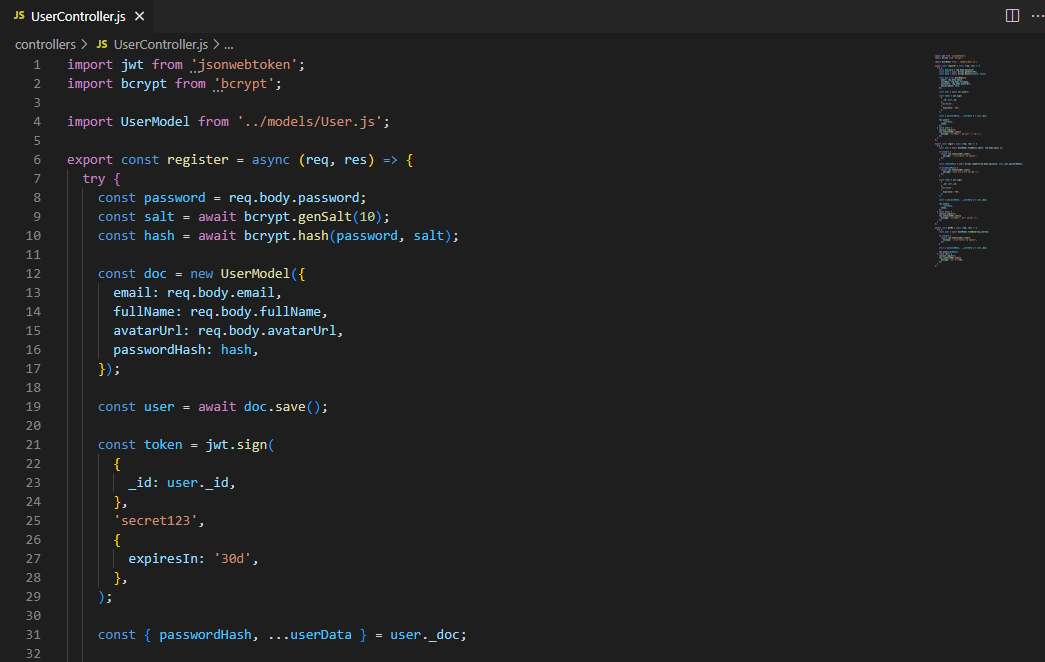
Последним файлом в данной папке будет index.js, отвечающий за экспорт функций/модулей, что позволит в дальнейшем легче обратиться к нужной функции.

****

**Рисунок 9. Файл index.js в папке utils**

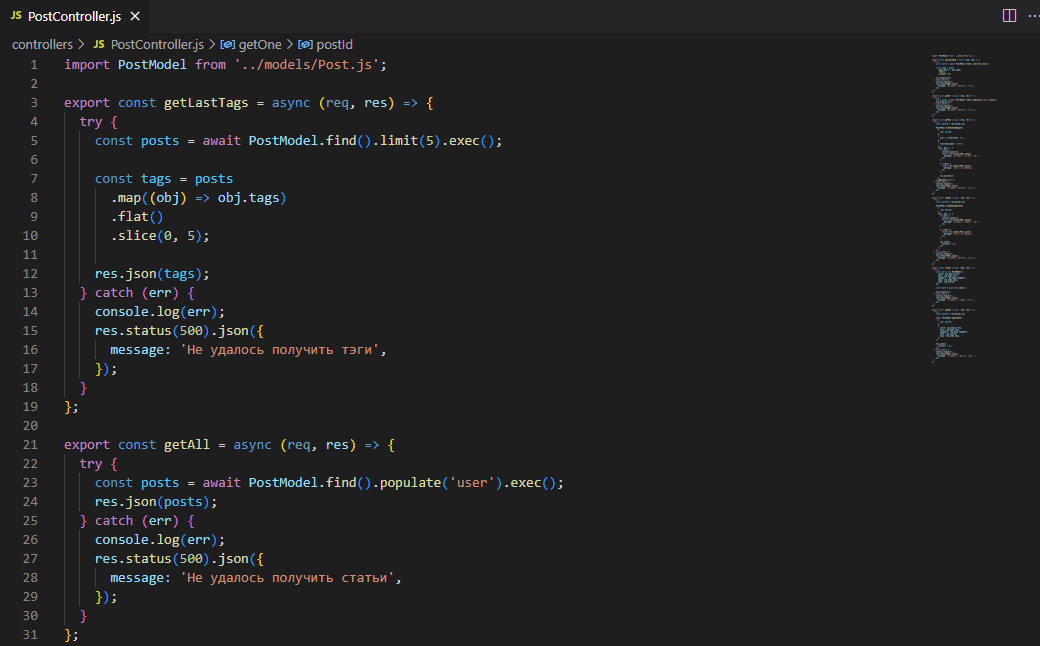
По мере создания информационного сайта объем кода растет. Для удобства создадим папку controllers, в которой файлы-контроллеры будут выполнять свою специализированную функцию.

В файле userController.js создаем асинхронную функцию авторизации (можно переименовать функцию, либо оставить под наименованием - register).

****

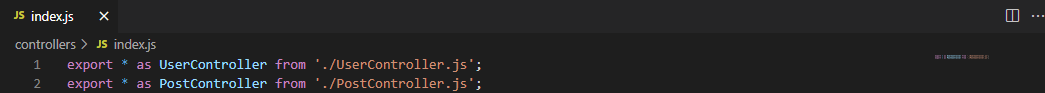
**Рисунок 10. Файл userController.js**

Файл postController.js реализует процедуры, связанные с управлением статьями на информационном веб-сайте. В частности, контроллер обеспечивает функциональность получения, создания, обновления и удаления статей.

****

**Рисунок 11. Файл postController.js**

Последним файлом в данной папке будет index.js, отвечающий за экспорт функций/модулей, что позволит в дальнейшем легче обратиться к нужной функции.

****

**Рисунок 12. Файл index.js в папке controllers**

**Разработка frontend**