|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| |  | | --- | | МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ | | ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ | | **«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»** | | **(МОСКОВСКИЙ ПОЛИТЕХ)** | | Факультет информационных технологий | |
|  |

Кафедра «Прикладная информатика»

Форма обучения: очная

|  |  |  |
| --- | --- | --- |
| **ОТЧЕТ**  **ПО ПРОЕКТНОЙ ДЕЯТЕЛЬНОСТИ** | | |
| Тема: «*Проект информационной системы визуализации данных NASA о метеоритных потоках с использованием СУБД MongoDB»* | | |
|  | | |
| Группа | *181-341* |  |
| Студенты | *\_\_\_\_\_\_\_\_\_\_\_\_\_*  *\_\_\_\_\_\_\_\_\_\_\_\_\_*  *\_\_\_\_\_\_\_\_\_\_\_\_\_*  *\_\_\_\_\_\_\_\_\_\_\_\_\_*  *\_\_\_\_\_\_\_\_\_\_\_\_\_* | А.С.Немнонов  А.М.Волосков  Т.А.Островская  Т.А.Костюк  А.А.Иванчихин |
| Оценка работы  Дата | \_\_\_\_\_\_\_\_\_\_\_\_ | 26 января 2021 |
|  |  |  |
|  |  |  |

МОСКВА 2021

СОДЕРЖАНИЕ

[АННОТАЦИЯ 4](#_Toc62696779)

[1 ОБЩИЕ СВЕДЕНИЯ О ПРОЕКТЕ 5](#_Toc62696780)

[1.1 Актуальность проекта 5](#_Toc62696781)

[1.2 Цель и задачи проекта 6](#_Toc62696782)

[1.3 Теоретическая и практическая значимость работы 7](#_Toc62696783)

[1.4 План и организация работы 7](#_Toc62696784)

[2 ИНДИВИДУАЛЬНЫЕ ПЛАНЫ УЧАСНИКОВ 9](#_Toc62696785)

[2.1 Участники и их роли 9](#_Toc62696787)

[2.2 Индивидуальные планы участников 9](#_Toc62696788)

[3 СОДЕРЖАТЕЛЬНЫЕ ГЛАВЫ 12](#_Toc62696789)

[3.1 Разработка frontend-части приложения 12](#_Toc62696791)

[3.1.1 Основной инструментарий приложения 12](#_Toc62696792)

[3.1.2 Дизайн приложения 12](#_Toc62696793)

[3.1.3 Отрисовка положения МКС 13](#_Toc62696794)

[3.2 Разработка backend-части приложения 14](#_Toc62696795)

[3.2.1 Выбор языка программирования и библиотек для серверной части приложения 14](#_Toc62696796)

[3.2.2 Выбор сервера для размещения приложения 15](#_Toc62696797)

[3.2.3 Содержание сайта 15](#_Toc62696798)

[4 РЕЗУЛЬТАТЫ 17](#_Toc62696799)

[4.1 Доступ к результатам 17](#_Toc62696801)

[ЗАКЛЮЧЕНИЕ 18](#_Toc62696802)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 19](#_Toc62696803)

[ПРИЛОЖЕНИЕ А 20](#_Toc62696804)

# АННОТАЦИЯ

Для разработки текущего проекта была выбрана тема: «Информационная система визуализации данных NASA о метеоритных потоках с использованием СУБД MongoDB». Предметной областью выбранного нами проекта является изучение и реализация возможностей работы с Node.js, а также с NoSQL СУБД MongoDB.

В результате изучения прикладных инструментов, а также космологических статей о метеоритах, астероидах и МКС, нашей командой было создано интернет-приложение. Любой пользователь, имеющий URL-ссылку на текущее интернет-приложение, может ознакомиться с интерактивными картами (МКС, упавшие метеориты), удобными таблицами космологических данных, а также наглядными графиками. С технической точки зрения, данное веб-приложение разрабатывалось с учетом тесной связи веб-сервера и репозитория Git.

Ключевые слова: информационная система, NASA, космос, СУБД, MongoDB, Node.js, МКС, астероиды, метеориты.

Отчет состоит из 19 страниц, содержит 5 литературных источников и одного приложения.

# ОБЩИЕ СВЕДЕНИЯ О ПРОЕКТЕ

## Актуальность проекта

Космос привлек внимание людей еще в древности. Сначала они наблюдали за видимыми кометами и планетами, затем, движимые научным интересом, стали изобретать всевозможные приборы, с помощью которых было возможно увидеть гораздо больше. А в философском смысле термин «космос» начинает употребляться уже в период становления первых философских школ Древней Греции. Холодная война и космическая гонка серьезно подстегнули развитие космической отрасли: запустили первые спутники, люди впервые вышли в открытый космос и отправили экспедиции (пока, конечно, без людей) на другие планеты. На текущий момент технологии позволяют людям изучать космос в бо́льших масштабах: изучать изменения космоса во времени, отправлять экспедиции на ближайшие космические объекты для сбора информации и грунта для последующего анализа на Земле, изменять траектории движения комет и даже отправили космический аппарат («Вояджер-1») за пределы Солнечной системы.

Людям по-отдельности довольно затруднительно исследовать такое большое пространство, как космос, поэтому этим занимаются различные объединения людей – коммерческие и государственные компании. Одной из таких компаний является американское ведомство NASA (National Aeronautics and Space Administration). Сама компания отвечает на вопрос: «Почему мы исследуем?» следующими словами: «Решая проблемы, связанные с освоением космоса человеком, мы расширяем технологии, создаем новые отрасли и помогаем укреплять мирные связи с другими странами. Любопытство и исследования жизненно важны для человеческого духа, и принятие задачи углубиться в космос пригласит граждан мира сегодня и будущие поколения присоединиться к НАСА в этом захватывающем путешествии» [[1](https://www.nasa.gov/exploration/whyweexplore/why_we_explore_main.html#.X9I05miPnIU)].

По заявлению Майкла Э. Липшуца и Людольфа Шульца в «Энциклопедии Солнечной системы», метеориты важны, потому что они содержат самые старые материалы солнечной системы для исследований и являются образцами широкого диапазона родительских тел – внешних и внутренних – некоторые примитивные, некоторые высокоразвитые. Метеориты регистрируют определенные солнечные и галактические эффекты и дают недоступные иным образом данные, относящиеся к генезису, эволюции и составу Земли [[2](https://www.sciencedirect.com/topics/earth-and-planetary-sciences/meteorite#:~:text=Meteorites%2C%20the%20%E2%80%9CPoor%20Man's%20Space,some%20primitive%2C%20some%20highly%20evolved)].

В своей работе мы хотим включить не только исследование комет, но и положение МКС на текущий момент времени. Международная космическая станция служит национальной лабораторией по изучению здоровья человека, биологических исследований и материалов, технологическим испытательным полигоном и ступенькой для дальнейшего продвижения в Солнечную систему. НАСА продолжит беспрецедентную работу с коммерческой отраслью и расширит целую отрасль, поскольку частные компании разрабатывают и эксплуатируют безопасные, надежные и доступные коммерческие системы для перевозки экипажа и грузов на Международную космическую станцию ​​и на низкую околоземную орбиту и обратно [[1](https://www.nasa.gov/exploration/whyweexplore/why_we_explore_main.html#.X9I05miPnIU)].

## Цель и задачи проекта

Основной целью, преследуемой куратором и разработчиками данного проекта, является создание удобной и наглядной информационной системы, которая могла бы знакомить пользователей с космосом и данными NASA.

В соответствии с поставленной целью, были выдвинуты следующие задачи:

1. Разработать техническое задание для проекта.
2. Закрепить теоретические основы в каждой из областей знаний, изучаемых участниками проекта.
3. Найти и структурировать «сырые данные», предоставляемые ведомством NASA.
4. Создать виртуальный кластер MongoDB для размещения подготовленных данных о метеоритах и астероидах.
5. Разработать frontend часть приложения.
6. Разработать backend часть приложения, в частности, внедрить API для отслеживания положения МКС.
7. Реализовать работу приложения на стороннем сервере, организовать связь с Git-репозиторием.
8. Создать комплект отчетных материалов по проекту.

## Теоретическая и практическая значимость работы

С теоретической точки зрения, данный проект может послужить стимулом к изучению космосу как для самих разработчиков проекта, так и для пользователей, работающих с данным интернет-приложением. Несомненно, в интернете существуют подобные приложения, однако, наш проект собрал воедино самые интересующие людей области космологии и предоставил возможность бесплатного доступа к бывшим неструктурированным данным.

С практической точки зрения, разработанное интернет-приложение способно лишний раз продемонстрировать достоинства работы с MongoDB, Node.Js и прикладными библиотеками. Данная работа является усовершенствованной версией многих подобных наработок в сети Интернет.

## План и организация работы

Особенностью работы с проектом в данном семестре была обусловлена эпидемиологической обстановкой в стране, поэтому коммуникация всех участников происходила исключительно в дистанционном формате. Исходя из этого, вместо еженедельных очных встреч, между участниками была организована связь в бесплатном проприетарном мессенджере Discord.

Перед началом работы был составлен план-график работы над проектом, где были указаны основные этапы и предполагаемые сроки их выполнения по времени. Все этапы были поделены на 3 условных группы: подготовительный, основной и заключительный. Ознакомиться с созданным планом можно в Таблице 1.

Таблица

Общий план-график работы над проектом

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № п/п | Наименование работ | Кол-во (объем) работ | Сроки работы (начало/  окончание) | Исполнители | Примечание |
| 1. | Организация коммуникации участников проекта. Первичное распределение обязанностей. | 4 ч. | 23.11.2020 –  26.11.2020 | Немнонов А.С. | Подготовительный этап |
| 2. | Создание и оформление Git-репозитория, обучение участников пользованию СУВ. | 5 ч. | 27.11.2020 –  29.11.2020 | Иванчихин А.А. |
| 3. | Создание и оформление таск-трекера. | 5 ч. | 30.11.2020 –  06.12.2020 | Костюк Т.А. |
| 4. | Налаживание работы с куратором в Teams. | 3 ч. | 30.11.2020 –  06.12.2020 | Немнонов А.С. |
| 5. | Ознакомление участников со своими обязанностями, составление индивидуальных планов. | инд. | 30.11.2020 –  06.12.2020 | Немнонов А.С.,  Островская Т.А.,  Костюк Т.А.,  Иванчихин А.А.,  Волосков А.М. |
| 6. | Написание технического задания. | 10 ч. | 07.12.2020 –  13.12.2020 | Островская Т.А. |
| 7. | Сбор данных. | 13 ч. | 14.12.2020 –  20.12.2020 | Островская Т.А. | Основной этап |
| 8. | Обработка и систематизация найденных данных. Создание кластера MongoDB. | 15 ч. | 21.12.2020 –  27.12.2020 | Островская Т.А.,  Немнонов А.С. |
| 9. | Разработка frontend-части приложения. | 50 ч. | 14.12.2020 –  17.01.2020 | Волосков А.М.,  Костюк Т.А. |
| 10. | Разработка backend-части приложения. | 60 ч. | 21.12.2020 –  25.01.2020 | Иванчихин А.А.,  Немнонов А.С. |
| 11. | Тестирование и отладка интернет-приложения. | 10 ч. | 18.01.2020 –  30.01.2020 | Иванчихин А.А., |
| 12. | Создание отчета. | 15 ч. | 26.01.2020 –  30.01.2020 | Островская Т.А.,  Немнонов А.С. | Заключительный этап |
| 13. | Создание сайта проекта (лендинг). | 25 ч. | 18.01.2020 –  30.01.2020 | Волосков А.М.,  Костюк Т.А. |
| 14. | Создание промо видео и видеопрезентации. | 15 ч. | 28.12.2020 –  30.01.2020 | Костюк Т.А. |
| 15. | Разработка презентации в формате PDF. | 13 ч. | 26.01.2020 –  30.01.2020 | Немнонов А.С. |
| 16. | Дизайн и оформление постера. | 12 ч. | 26.01.2020 –  30.01.2020 | Островская Т.А. |
| 17. | Подготовка проекта к сдаче, окончательная проверка документов и приложения. | 24 ч. | 29.01.2020 –  30.01.2020 | Немнонов А.С.,  Островская Т.А.,  Костюк Т.А.,  Иванчихин А.А.,  Волосков А.М. |

# ИНДИВИДУАЛЬНЫЕ ПЛАНЫ УЧАСНИКОВ



## Участники и их роли

В состав команды входят 5 студентов группы 181-341. Роли участников были распределены следующим образом:

1. Немнонов А.С. – тимлид, backend-разработчик.
2. Островская Т.А. – аналитик, ответственный за документацию и дизайн отчетных материалов.
3. Иванчихин А.А. – backend-разработчик, тестировщик, ответственный за Git-репозиторий.
4. Волосков А.М. – старший frontend-разработчик, дизайнер.
5. Костюк Т.А. – ответственный за видеоконтент, младший frontend-разработчик.

## Индивидуальные планы участников

В соответствии с общим планом, каждый из участников заранее составил свой индивидуальный план. Распределение обязанностей между участниками было совершено таким образом, чтобы суммарное количество затраченных часов на них было примерно одинаковым. План представлен в Таблице 2.

Таблица

Индивидуальный план участников

|  |  |  |
| --- | --- | --- |
| Участник проекта | Кол-во часов | Индивидуальный план |
| Немнонов Андрей Сергеевич | 60 ч. | 1. Организация коммуникации участников проекта.    1. Создание конференции в Telegram.    2. Создание конференции в VK.    3. Создание конференции в Discord, координация и составление расписания еженедельной связи между участниками. 2. Первичное распределение обязанностей между участниками. 3. Налаживание работы с куратором в Teams. 4. Ознакомление со своими обязанностями, составление индивидуального плана. 5. Создание кластера MongoDB на виртуальном облаке.    1. Регистрация на сайте MongoDB.    2. Совместная с аналитиком загрузка данных в кластер. 6. Разработка каркаса приложения, отладка работы простейшего сервера, подключение библиотек и подключение БД к серверу.    1. Инициализация проекта Node.js и подключение библиотек.    2. Написание простого сервера с использованием Express.    3. Создание каркаса с использованием Handlebars. 7. Помощь в написании отчета.    1. Координация команды для того, чтобы каждый из участников отправил свои части отчетных материалов.    2. Написание содержательной части.    3. Окончательная проверка оформления. 8. Создание презентации в формате PDF. 9. Подготовка проекта к сдаче, окончательная проверка документов и приложения. |
| Островская Тамара Александровна | 55 ч. | 1. Ознакомление со своими обязанностями, составление индивидуального плана. 2. Написание и редакция технического задания для текущего проекта.    1. Поиск информации о правилах оформления ТЗ и соответствию ГОСТу.    2. Создание первой версии и последующая редакция. 3. Поиск данных в различных источниках, в частности, на сайте NASA.    1. Поиск данных в источниках, отличных от NASA.    2. Основной поиск данных на официальном ресурсе NASA [[3](https://data.nasa.gov/)].    3. Обоснование выбора конкретных данных. 4. Сбор и последующая обработка найденных данных, помощь в загрузке их в кластер.    1. Загрузка файлов на локальную машину.    2. Унификация данных путем преобразования файлов в формат JSON для работы с MongoDB.    3. Очистка данных от выбросов и пустых строк.    4. Совместная загрузка данных с backend-разработчиками. 5. Совместное написание отчета. 6. Оформление постера. 7. Подготовка проекта к сдаче, окончательная проверка документов и приложения. |
| Иванчихин Андрей Александрович | 60 ч. | 1. Работа с Git-репозиторием.    1. Создание Git-репозитория текущего проекта.    2. Оформление веток и элементов работы Git.    3. Обучение прочих участников основам работы с Git. 2. Ознакомление со своими обязанностями, составление индивидуального плана. 3. Разработка backend-части приложения на основе раннее созданного каркаса.    1. Отладка работы базы данных MongoDB.    2. Подключение API для мониторинга нахождения МКС на карте Земли.    3. Создание таблицы с данными для вывода списка метеоритов и астероидов.    4. Дополнительное создание фильтров для упавших метеоритов.    5. Создание интерактивной карты для метеоритов.    6. Создание графиков для метеоритов.    7. Реализация возможности сортировки неупорядоченных данных. 4. Тестирование и отладка интернет-приложения. 5. Подготовка проекта к сдаче, окончательная проверка документов и приложения. |
| Волосков Артем Михайлович | 55 ч. | 1. Ознакомление со своими обязанностями, составление индивидуального плана. 2. Разработка frontend-части приложения с согласованием тимлида и младшего frontend-разработчика.    1. Создание первичного дизайна сайта в Adobe Photoshop CC 2019.    2. Создание логотипа сайта.    3. Написание каркаса сайта с использованием Bootstrap.    4. Подключение JS-скриптов.    5. Адаптация сайта для кроссплатформенного доступа. 3. Создание сайта проекта (лендинг).    1. Согласование с backend-разработчиками.    2. Создание первичного дизайна сайта, согласование с дизайнером постера.    3. Разработка сайта с использованием CSS и JS. 4. Подготовка проекта к сдаче, окончательная проверка документов и приложения. |
| Костюк Татьяна Александровна | 50 ч. | 1. Создание и оформление таск-трекера для команды с согласованием тимлида и других участников. 2. Ознакомление со своими обязанностями, составление индивидуального плана. 3. Помощь в разработке frontend-части приложения.    1. Поиск данных и интересных фактов о космосе и падении метеоритов.    2. Наполнение главной страницы текстовым контентом.    3. Поиск иллюстраций для сайта. 4. Помощь в разработке сайта проекта (лендинга). 5. Работа с видеоматериалом.    1. Сбор информации для создания промо-ролика.    2. Монтаж и обработка промо-видео.    3. Согласованное с остальными участниками создание видеопрезентации. Монтаж и обработка. 6. Подготовка проекта к сдаче, окончательная проверка документов и приложения. |

# СОДЕРЖАТЕЛЬНЫЕ ГЛАВЫ



## Разработка frontend-части приложения

### Основной инструментарий приложения

Вся frontend-часть приложения выполнена на HTML, CSS и JS. С помощью HTML выполнено построение структуры самого приложения. Для придания более красивого внешнего вида было решено добавить в использование CSS.

JavaScript используется в связке с jQuery, так как многие элементы главной страницы, такие как, например, анимация появления текста или следования шапки (header) за прокруткой, было намного удобнее и проще описывать с данным фреймворком. В приложении были использованы плагины, например Owl Carousel, так как использовать плагины очень удобно и это знатно сокращает время разработки приложения.

Также был использован фреймворк Bootstrap. Выбор данного фреймворка обусловлен не только наличием адаптивности, но и возможностью удобного расположения элементов приложения.

### Дизайн приложения

Первичный дизайн приложения был разработан в Adobe Photoshop CC 2019 и немного отличался от окончательного получившегося варианта. Использование «фотошопа» обусловлено более удобным представлением внешнего вида будущего проекта. После ознакомления с дизайном другими членами нашей команды и последующей его редакции, началась работа над переносом получившегося дизайна в наше приложение.

Выбор цветовой гаммы приложения обусловлен тематикой космоса. Логотип был разработан, учитывая функционал нашего проекта, дизайн логотипа также выполнен в космической стилистике, возможно, чем-то напоминая логотип компании NASA. На логотипе можно заметить такие элементы как летящий метеорит, международную космическую станцию и название нашего проекта. Таким образом, логотип приложения полностью раскрывает его основные опции. Название «OuterSpace» также было выбрано не случайно, «OuterSpace» в переводе с английского означает «космическое пространство». Стоит отметить, что первоначально было использовано другое название – «OuttaSpace», но от него было решено отказаться в силу того факта, что это фраза используется в разговорном английском и является слишком простой для столь серьезного проекта.

В интернет-приложении размещены 4 страницы – главная, МКС, метеориты и астероиды. На главной странице можно ознакомиться с основным функционалом приложения и целями проекта, а также основной информацией о NASA, МКС и небесными телами. На странице МКС находится карта с текущим положением МКС, а также прямая трансляция с камер, находящихся на борту самой МКС. На страницах метеориты и астероиды находятся таблицы с названием и характеристиками небесных тел, причем на странице метеориты находится карта с указанием места падения конкретного метеорита. Все страницы выполнены в одном сочетающемся дизайне. Инструменты разработки дизайна приведены выше.

### Отрисовка положения МКС

Отрисовка МКС происходит полностью с помощью клиентского скрипта с использованием библиотеки Leaflet. Leaflet позволяет создавать на странице интерактивные карты на базе самых разных map API. Библиотека весит всего 39КБ, очень быстрая и поддерживает open-source карты, например, Open Street Maps. Это делает её прекрасным инструментом для выполнения нашей задачи. Так как для нашей задачи (отображение положения МКС на карте) не нужна максимальная точность, то было принято решение использовать Open Street Maps в качестве источника изображений карты. Эти карты не ограничены в количестве запросов к ним, а при желании можно запустить свой сервер тайлов карты. Данные о положении МКС берутся с сайта open-notify.org. Каждые пять секунд эти данные обновляются скриптом, и МКС меняет своё положение на карте. Также для развлечения пользователя при каждом обновлении страницы маркер МКС снабжается небольшим познавательным фактом о станции. В дополнение к отрисовке МКС на карте, на странице присутствует вставка трансляции с камеры, расположенной на самой МКС.

## Разработка backend-части приложения

### Выбор языка программирования и библиотек для серверной части приложения

Для серверной части веб-приложения был выбран язык JavaScript со средой выполнения Node.js. В первую очередь выбор был обоснован широкими возможностями асинхронного выполнения запросов. Для сайта, работающего с большой базой данных, важно не останавливаться на долго выполняющемся запросе, а продолжать выполнять другие запросы или обрабатывать данные. Node.js позволяет очень легко работать с асинхронными запросами и обеспечивать быстрейшую загрузку страниц и данных [4]. Также, поскольку разработка frontend- и backend-частей ведётся на одном и том же языке, то гораздо проще договориться об интерфейсе взаимодействия между этими двумя частями и не тратить время на объяснение особенностей других языков. Node.js обладает огромной базой готовых модулей npm, а также многочисленным сообществом программистов – это серьёзно упрощает и ускоряет разработку.

В связке с Node.js используется библиотека Mongoose. Mongoose позволяет определять объекты со строго-типизированной схемой, соответствующей документу MongoDB [5]. Она позволяет очень удобно работать с MongoDB с помощью схем документов и удобного механизма запросов к базе данных.

### Выбор сервера для размещения приложения

Куратором было рекомендовано размещать свои веб-приложения на хостинге, предоставляемом самим университетом, но из-за проблем с доступом к своему аккаунту, прецедентов взлома аккаунтов и прочих проблем, было принято решение разместить веб-приложение на стороннем хостинге. Размещение на стороннем хостинге даёт не только максимальный контроль над своим приложением, но и позволяет получить ценный опыт настройки своего сервера. Поскольку это проект без чёткой монетизации, то в приоритете стояла невысокая стоимость и минимальное количество сбоев. Для аренды VPS сервера был выбран облачный хостинг RUVDS, который хорошо себя зарекомендовал, он обладает сравнительно низкими ценами и входит в топ-20 российских поставщиков IAAS-услуг. Сам же сервер обладает следующей конфигурацией: Intel Xeon, 2.2ГГц – 1 ядро, 1Гб RAM, 20ГБ HDD. Выбор для операционной системы стоял между Ubuntu и Debian, но был сделан выбор в пользу последней, т.к. она уже довольно долго существует и для неё написано множество руководств.

### Содержание сайта

Для формирования главной страницы использовался простой механизм ответа на запросы через router.get(). В нём, с помощью res.render(), производилась постройка выдаваемой страницы из шаблонов handlebars.

При работе над страницей с информацией о МКС тоже не была использована серверная обработка, кроме, быть может, стандартного формирования и отдачи шаблона через res.render(). Вся информация о местоположении получается и обрабатывается браузером клиента. Вероятно, с увеличением аудитории приложения возникнет необходимость обработки данных на сервере, чтобы не перегружать сервер API, предоставляющего информацию о местоположении МКС. Однако, в данной ситуации клиентская обработка наиболее удобна и проста.

Страница с информацией об упавших метеоритах обладает самой сложной серверной логикой. В процессе разработки для обработки запросов frontend скриптов был выделен отдельный адрес /service/query, на который скрипты могут посылать свои get запросы. При получении запросов по этому адресу сервер извлекает параметры из URL и на основе полученного запроса выполняет различные запросы к базе данных, обрабатывает ответы и возвращает обработанные данные в JSONP формате. Благодаря Mongoose, это делается очень просто. Блок кода, отвечающий за выдачу JSONP, находится в Приложении А. Функция url.parse() парсит URL в объект JS, далее, с помощью ряда условий формируется объект queryFilter, в котором содержится форматированный запрос к MongoDB. В конце блока выполняется запрос к БД, и выполняется выдача полученного материала. При выполнении запроса ставится ограничение на 10000 записей, т.к. при большем количестве клиентский скрипт очень долго отрисовывает полученные результаты, что может негативно сказаться на впечатлении о приложении. Также, в том же Приложении А, в конце можно заметить блок обработки запросов с параметром «asteroid», который отвечает за отдачу результатов запросов таблицы астероидов. Как и в случае с метеоритными потоками, здесь также формируется объект queryFilter для запроса к БД, и выполняется последующая выдача отфильтрованных данных.

# РЕЗУЛЬТАТЫ

В результате работы была создана информационная система для визуализации данных NASA о метеоритных потоках с использованием MongoDB. Также были реализованы дополнительные функции, такие как:

* отслеживание текущего положения МКС на карте Земли;
* дополнительная визуализация в виде интерактивных графиков;
* интересная информация о космосе;
* прямая трансляция с борта МКС;
* вывод астероидов (NEO) в отдельную таблицу.

Получившаяся информационная система имеет большие перспективы развития в будущем. По замыслу разработчиков, далее могут быть реализованы такие функции, как:

* демонстрация космонавтов, находящихся прямо сейчас на борту МКС;
* позиции Солнца и Луны относительно горизонта с учетом местоположения;
* трансляции текущего расположения звезд и планет на небе с учетом местоположения;
* информация о магнитных бурях.



## Доступ к результатам

* Ссылка на Git-репозиторий проекта: <https://github.com/AndreiAvinov/MeteorMap>
* Ссылка на главное интернет-приложение: <http://js.andreiavinov.xyz/>
* Ссылка на сайт о проекте (лендинг): <http://pd-2020-1.std-1273.ist.mospolytech.ru/>
* Ссылка на таск-трекер: <https://trello.com/b/TlnLyQHF/nasa-site>

# ЗАКЛЮЧЕНИЕ

Изначальный выбор темы данного проекта был вызван изучением NoSQL СУБД MongoDB, а также интересом к изучению космоса. При поиске необходимых данных было выяснено, что ведомство NASA содержит в открытом доступе обширное количество данных о различных космических объектах, миссиях и проводимых исследованиях. Поэтому, при углублении в проект, было принято решение расширить количество данных и, соответственно, количество функций, ожидаемых в итоговой информационной системе. Помимо этого, была определена важность изучения космоса и космических объектов в целом.

В процессе реализации данного проекта студенты, на основе пройденных дисциплин, продолжили изучать, закреплять и улучшать полученные ранее навыки: работа с программной платформой Node.js, СУБД MongoDB, применение свободного набора инструментов для создания сайтов Bootstrap, работа с распределенной системой управления версиями Git. Помимо этого, в данном проекте было уделено огромное внимание технической документации, необходимой для создания информационной системы.

В итоге была создана работающая информационная система, позволяющая пользователям в доступной форме ознакомиться с данными о космических объектах, предоставляемыми ведомством NASA, а также поближе познакомиться с МКС и просматривать прямую трансляцию с борта.

На данном этапе работа над проектом не завершена, он имеет хорошую перспективу для развития, о чем было сказано выше. С результатами проекта (основное интернет-приложение, видеоконтент, постер) можно ознакомиться, перейдя по ссылкам, также указанным выше.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Why we explore – NASA – [Электронный ресурс]. Режим доступа: https://www.nasa.gov/exploration/whyweexplore/why\_we\_explore\_main.html#.YBCmreiPnIW (Дата обращения: 27.01.2020)
2. Meteorite – an overview – ScienceDirect – [Электронный ресурс]. Режим доступа: https://clck.ru/SyHnA (Дата обращения: 27.01.2020)
3. Open Data Portal – Nasa – [Электронный ресурс]. Режим доступа: https://nasa.github.io/data-nasa-gov-frontpage/ (Дата обращения: 27.01.2020)
4. About – Node.js – [Электронный ресурс]. Режим доступа: https://nodejs.org/en/about/ (Дата обращения: 27.01.2020)
5. Mongoose – [Электронный ресурс]. Режим доступа: https://mongoosejs.com (Дата обращения: 27.01.2020)

# ПРИЛОЖЕНИЕ А

1. router.get('/service/query', async (req,res) => {
2. **const** queryObject = url.parse(req.url,**true**).query;
3. **var** queryFilter = {}
4. **if** (queryObject.object == 'meteorite'){
5. **if** (!queryObject.recclass){**delete** queryObject.recclass;}
6. **else**{
7. queryFilter.recclass = queryObject.recclass
8. }
9. **if** (!queryObject.fromYear){**delete** queryObject.fromYear;}
10. **else**{
11. **if** (**typeof** queryFilter.year == 'undefined') {queryFilter.year = {};}
12. queryFilter.year['$gte'] = parseInt(queryObject.fromYear)
13. }
14. **if** (!queryObject.toYear){**delete** queryObject.toYear;}
15. **else**{
16. **if** (**typeof** queryFilter.year == 'undefined') {queryFilter.year = {};}
17. queryFilter.year['$lte'] = parseInt(queryObject.toYear)
18. }
19. **if** (!queryObject.fromMass){**delete** queryObject.fromMass;}
20. **else**{
21. **if** (**typeof** queryFilter.mass == 'undefined') {queryFilter.mass = {};}
22. queryFilter.mass['$gte'] = parseFloat(queryObject.fromMass)
23. }
24. **if** (!queryObject.toMass){**delete** queryObject.toMass;}
25. **else**{
26. **if** (**typeof** queryFilter.mass == 'undefined') {queryFilter.mass = {};}
27. queryFilter.mass['$lte'] = parseFloat(queryObject.toMass)
28. }
29. **const** landmets = await landmet.find(queryFilter).limit(10000).lean()
30. res.jsonp(landmets)
31. }**else** **if** (queryObject.object == 'asteroid'){
32. **const** asteroids = await asteroid.find(queryFilter).lean()
33. res.jsonp(asteroids)
34. }
35. })