

Министерство просвещения Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования

«Чувашский государственный педагогический университет
им. И.Я. Яковлева»

Факультет физико-математического образования информатики и технологий

Кафедра информатики и технологий

УТВЕРЖДАЮ
Заведующий кафедрой
_____ К. Н. Фадеева
«__» _____ 2023 г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ (БАКАЛАВРСКАЯ) РАБОТА

Направление подготовки бакалавров
09.03.03 Прикладная информатика
Профиль: Прикладная информатика в государственном и муниципальном
управлении

Виртуальная 3D-экскурсия по образовательной организации

Научный
руководитель

подпись, дата

канд. физ.-
мат. наук,
должность,
ученая степень

А.В. Горский
инициалы,
фамилия

Выпускник

подпись, дата

Т.Е. Пигальцев

инициалы,
фамилия

Содержание

Введение	3
1. Теоретические основы создания виртуальных 3D-экскурсий по образовательным организациям.....	5
1.1 Понятие панорамы	5
1.2 Понятие система управления содержимым.....	6
1.3 Обзор и сравнение готовых программных решений для создания виртуальных 3D-экскурсий	7
1.4. Постановка и обоснование задачи проекта	10
1.7. Анализ предметной области и требований к программному продукту	11
2. Проектирование и разработка виртуальной 3D-экскурсии по образовательной организации	18
2.1. Выбор средств разработки и архитектуры программного продукта	18
2.2. Разбиение задачи на подзадачи и определение алгоритмов решения	19
2.3. Реализация клиентская часть	23
2.4. Реализация серверной части.....	31
3. Тестирование виртуальной 3D-экскурсии по образовательной организации.....	39
4. Расчет себестоимости и экономической эффективности виртуальной 3D-экскурсии по образовательной организации	41
4.1. Расчет трудоемкости разработки.....	41
4.2. Расчет затрат на электроэнергию	41
4.3 Составление сметы всех затрат.....	42
Заключение.....	43
Список использованных источников	44

Введение

Образование является краеугольным камнем развития общества, давая людям знания и навыки, которые помогают им ориентироваться в сложных ситуациях современного мира. Из-за стремления образовательных организаций обеспечить значимый опыт обучения, они должны постоянно изучать инновационные подходы к использованию технологий с растущим уровнем. В последние годы виртуальная реальность (VR) и 3D-моделирование стали инструментами, способными произвести революцию в сфере образования.

Данная концепция виртуального 3D-тура предлагает уникальную возможность преодоления разрыва между цифровой и физической средой обучения. Создание иммерсивных и интерактивных копий учебных залов позволяет учебным заведениям предлагать обучающимся не только самим исследовать учебные аудитории, но также их сопровождают преподаватели. Такой вид виртуального исследования может имитировать опыт физического присутствия, позволяя людям передвигаться по классам, лабораториям и библиотекам разного назначения, не выходя из своих устройств.

Важность вовлеченности и доступности в обучении подчеркивает студенческий опыт. Из-за таких ограничений, как расстояние и ограниченные ресурсы традиционные методы образования часто ограничивают доступ к физическим пространствам. Более того, метод пассивного обучения часто не может привнести интерес и любопытство студентов в предмет. Это препятствует полной погруженности их знаний.

У образовательных организаций появилась возможность преодоления этих ограничений с появлением виртуальных 3D-туров. Применяя эту технологию, они имеют возможность создавать динамичную и инклюзивную среду обучения, отвечающую разнообразным потребностям учеников. Помимо этого, виртуальные 3D-туры способствуют стимуляции вовлеченности и подогревают любопытство. Они способны увеличить общий уровень обучения

за счет более глубокого понимания образовательных пространств посредством своего визуального эффекта.

Потому что границы между реальным и цифровым миром все еще стираются, образовательным организациям крайне важно внедрять инновационные решения позволяющие преодолеть этот разрыв. В настоящее время образовательные технологии используют виртуальный 3D-тур, который является захватывающим рубежом в технологиях обучения.

Объект исследования: Разработка виртуального 3D тура

Предмет исследования: Образовательная организация

Цель выпускной квалификационной работы- создание программного продукта “Виртуальная 3D экскурсия по образовательной организации”, который позволит объединить практический опыт с теоретическими знаниями, что даст возможность создать иммерсивный и информативный виртуальный личный опыт. Работа посвящена использованию технологий для продвижения образовательных организаций, повышения вовлеченности пользователей и внесения вклада в развитие технологических средств данного направления.

Для решения поставленной цели необходимо решить следующие задачи:

- изучить предметную область и инструментальную среду разработки
- обосновать выбор программной среды
- разработать техническое задание
- разработать программный продукт
- провести тестирование программного продукта

1. Теоретические основы создания виртуальных 3D-экскурсий по образовательным организациям

1.1 Понятие панорамы

Понятие "панорама" относится к широкому, беспрепятственному виду пейзажа или сцены, обычно запечатленному на фотографии, картине или в цифровом представлении. Цель панорам - обеспечить полное и всеобъемлющее восприятие, охватывая большое поле зрения, редко превышающее то, что возможно запечатлеть в одном кадре.

Создание панорамы в фотографии производится путем соединения нескольких отдельных изображений, которые создадут цельный панорамный вид. Данная техника даёт возможность охватить более обширную перспективу и представить больше захватывающего изображения сцены.

Задача концепции панорамы выйти за рамки фотографии и может быть применена в других видах искусства. Именно панорамные картины были популярны в 19 веке. Художники, создавая масштабнейшие произведения искусства, изображающие величественные виды природы или исторических событий вызывают у зрителей ощущение присутствия на сцене.

С приходом в нашу жизнь цифровых технологий все большее распространение получают виртуальные панорамы. Изучайте и создавайте захватывающие панорамы с помощью платформ виртуальной реальности (VR) или дополненной действительности, разработанных для AR. У пользователей появится возможность перемещения по виртуальным средам и взаимодействовать с панорамными видами, тем самым увеличивая их вовлеченность.

Панорамы могут быть использованы для разных целей, среди которых туризм, архитектурная визуализация игр и образовательные цели. Они предлагают максимально увлекательный, интерактивный способ изучения и

восприятия разнообразных мест действия. Это дает ощущение погружения в материал постановки с максимальной реалистичностью.

В общем, панорама — это обширный вид сцены или ландшафта, который получают при помощи фотографии, живописи либо цифрового представления. Её цель - обеспечить полный и захватывающий визуальный опыт, дающий возможность зрителю исследовать изображенную среду через контакт с ней.

1.2 Понятие система управления содержимым

Предоставляемая концепция системы управления контентом (CMS) представляет собой динамичную и универсальную платформу, позволяющую различным частным лицам или организациям эффективно управлять своим цифровым продуктом. В своей сути CMS – это надёжное программное обеспечение, предназначенное для упрощения всего жизненного цикла контента от создания и организации до публикации на свет.

Главной отличительной чертой CMS является то, что она может предоставлять удобный и централизованный интерфейс управления контентом. Пользователи, не зависимо от их технических знаний могут легко создавать и форматировать контент. Загружать файлы в медиа формат, организовывать групповой подбор информации соответствуя своим запросам. Этот подход не только повышает производительность, способствует сотрудничеству и обеспечивает последовательную доставку контента.

Еще одно из достоинств CMS – разделение содержания и представления. Разделяя контент от дизайна, CMS даёт возможность пользователям сконцентрироваться исключительно на создании и курировании контента оставляя нюансы верстки или презентации в стороне. Этот модульный подход дает возможность легко настраивать сайт, получая при этом гибкий и отзывчивый дизайн, ориентированный к конкретной целевой аудитории.

Комплексная CMS имеет в себе ряд функциональных возможностей и дополнительных функций. В их перечень включаются управление пользователями и правами доступа, контроль версий, возможности поиска в многоязычной поддержке инструменты поисковой оптимизации. Эти функции дают эффективное управление контентом, контроль доступа пользователей и редактирование контента. Они также обеспечивают высокую поисковую видимость в поисковых системах.

На сегодняшний день CMS способна адаптироваться к различным отраслевым требованиям и условиям эксплуатации. Она может удовлетворить самые разнообразные потребности, начиная от персональных блоковых платформ и заканчивая системами корпоративного уровня. Они способны обеспечить работу веб-сайтов с большой посещаемостью, а также площадок электронной коммерции. Масштаб и ширина CMS делают ее адаптивным решением, которое может расширяться вместе с растущими потребностями организации в контенте.

Исходя из сути, концепция системы управления контентом представляет собой комплексную и понятливую платформу для пользователей. Она позволяет эффективно управлять своим цифровым продуктом. Грамотный интерфейс, разграничивающий контент с помощью специальных возможностей CMS способствует продуктивной публикации контента в современном цифровом ландшафте.

1.3 Обзор и сравнение готовых программных решений для создания виртуальных 3D-экскурсий

Рынок программ, которые связаны с созданием панорам, разнообразен и предлагает ряд решений предназначенных для разных нужд пользователей. Функциональные инструменты этих программ предоставляют возможность создания панорамных видов, сшивания нескольких изображений вместе и повышения качества результата.

Преимущества программ для создания панорам:

- **Возможность сшивания:** программы для создания панорам отлично справляются со склеиванием нескольких изображений и созданием бесшовных панорам. Новейшие алгоритмы используются для сопоставления содержимого изображений, при этом происходит точное выравнивание и минимальное искажение.
- **Возможности настройки:** Эти программы часто предлагают широкий спектр возможностей настройки, позволяя пользователям регулировать такие параметры, как фокусное расстояние, тип проекции и коррекция перспективы. Такая гибкость обеспечивает творческий контроль и возможность подгонять панорамы под конкретные предпочтения или художественное видение.
- **Инструменты улучшения изображения:** Программы для создания панорам часто включают инструменты для улучшения качества конечного результата. Эти инструменты могут включать корректировку экспозиции, цветокоррекцию, шумоподавление и повышение резкости, позволяя пользователям точно настроить внешний вид панорамы.
- **Эффективность рабочего процесса:** Автоматизируя процесс сшивания и предоставляя интуитивно понятные пользовательские интерфейсы, программы для создания панорам оптимизируют рабочий процесс и экономят время пользователей. Они упрощают задачу создания панорам даже для людей с ограниченными техническими знаниями, позволяя быстро и эффективно создавать потрясающие панорамные изображения.

Минусы программ для создания панорам:

- **Кривая обучения:** Некоторые программы для создания панорам могут иметь кривую обучения, особенно для пользователей, которые только начинают работать с программным обеспечением или техниками сшивания

изображений. Понимание различных настроек и рабочих процессов может потребовать некоторых первоначальных усилий и экспериментов.

- Ресурсоемкость: Создание панорам, особенно с высоким разрешением или крупномасштабных, может быть ресурсоемким с точки зрения вычислительной мощности и использования памяти. Пользователи могут столкнуться с замедлением производительности или потребовать более надежного оборудования для обработки сложных вычислительных требований при сшивании и рендеринга панорамных изображений.

- Ограничения в сложных сценариях: Программы для создания панорам могут столкнуться с трудностями при работе со сложными сценами, такими как сцены с движущимися объектами или сцены с непостоянными условиями освещения. Эти проблемы могут привести к некачественной сшивке или артефактам в конечном результате.

- Стоимость: хотя существуют бесплатные или недорогие программы для создания панорам, некоторые продвинутые или профессиональные программные решения могут иметь свою цену. Эти программы премиум-класса часто предлагают дополнительные функции и расширенные возможности, но они могут потребовать финансовых вложений.

Цены на программы, которые нужны для создания панорам могут существенно различаться. Ниже представлена общая разбивка ценовых сегментов, охватывая примерные минимальные и максимальны расходы:

- Бесплатное/программное обеспечение с открытым исходным кодом: Эти программы для создания панорам доступны бесплатно, что делает их очень доступными для пользователей. Примеры: Hugin, Microsoft ICE (Image Composite Editor) и Autostitch.

- Потребительское программное обеспечение начального уровня/премиум-класса: В среднем ценовом сегменте программы для создания панорам обычно стоят от 30 до 150 долларов. Эти программы предлагают баланс между доступностью и расширенными возможностями. В

качестве примера можно привести PanoramaStudio, PTGui и Serif PanoramaPlus.

– Программное обеспечение профессионального уровня: Профессиональное программное обеспечение для создания панорам обычно стоит дороже из-за расширенных возможностей и целевой аудитории. Цены могут варьироваться от 200 до 1000 долларов и выше, в зависимости от конкретного программного обеспечения и вариантов лицензирования. В качестве примера можно привести Kolor Autopano, Pano2VR и Adobe Photoshop с его возможностями сшивки панорам.

Стоит помнить о том, что указанная цена является приблизительной и может изменяться в зависимости от таких факторов как: скидки, акции географическое положение, а также конкретные лицензионные соглашения. Также некоторые программы для создания панорам имеют возможность предлагают различные модели подписки или уровня цен, что позволяет пользователям выбирать уровень функциональности и поддержки соответственный их потребностям.

1.4. Постановка и обоснование задачи проекта

Необходимо выделить клиентскую и серверную часть виртуальной 3D экскурсии по образовательной организации.

Метод отображения панорам образовательной организации и метод навигации для перемещения отображаются в клиентской части.

Отображение всей системы управления контентом обновление, добавление изменений и удаление с помощью авторизаций.

В процессе виртуального знакомства должны отображаться кабинеты учебных заведений. Местоположение системы навигации должно быть удобным и охватывать все здание. Кнопка управления должна иметь способы редактирования контента. Оптимизация интерфейса под мобильные устройства.

Чтобы разработать программный продукт, нужны навыки и знания разработки сайта, которые получены в ходе обучения или же практики, проведенной на предприятии. Возможно, некоторые дополнительные сведения в области программирования среде HTML, CSS, Bootstrap, JQuery, Ajax, Node.js и Strapy. Так же потребовался смартфон с установленным приложением «Google Camera» и хостинг для размещения ПП на сайте в общем пользовании.

1.7. Анализ предметной области и требований к программному продукту

В учебной области «3D виртуальный тур по образовательной организации» внимание фокусируется на создании зрительного образа учебного заведения, такого как школа или университет в 3Д формате с использованием трехмерной технологии. Виртуальный тур дает возможность пользователям исследовать учебные аудитории, библиотеки и другие места организации без физического присутствия на месте.

Ключевые компоненты и особенности:

1. 3D-моделирование: Предметная область включает создание 3D-моделей физических пространств и структур образовательной организации. Это могут быть здания, классы, аудитории, коридоры, открытые площадки и т.д. 3D-модели должны точно отображать реальные пространства и обеспечивать реалистичный и захватывающий опыт для пользователей.

2. Навигация и взаимодействие: Виртуальный тур должен предоставлять интуитивно понятные возможности навигации, позволяя пользователям легко перемещаться по виртуальной среде. Пользователи должны иметь возможность исследовать различные области, перемещаться между этажами или зданиями.

3. Доступность и совместимость: Виртуальный тур должен быть доступен для широкого круга пользователей, включая людей с ограниченными возможностями. Он должен соответствовать стандартам доступности, позволяя пользователям перемещаться и взаимодействовать с виртуальной

средой с помощью вспомогательных технологий. Виртуальный тур также должен быть совместим с различными устройствами и платформами, включая настольные компьютеры, ноутбуки, планшеты и мобильные устройства.

4. Интеграция с существующими системами: Виртуальный тур может быть интегрирован с другими существующими системами или платформами, используемыми образовательной организацией. Например, он может быть связан с веб-сайтом учебного заведения, системой управления обучением или информационной системой для студентов, чтобы обеспечить беспрепятственный пользовательский опыт и доступ к дополнительным ресурсам.

Преимущества и применение:

1. Удаленное привлечение потенциальных студентов: Будущие студенты могут дистанционно изучить учебное заведение, получить представление о его помещениях, ресурсах и учебной среде. Это может помочь им принять обоснованное решение о поступлении.

2. Ориентация для новых студентов: Виртуальный тур может послужить инструментом ориентации в учебном заведении для новых студентов, позволяя им ознакомиться с планом образовательной организации, объектами и ключевыми местами до физического посещения.

3. Рекламный и маркетинговый инструмент: Виртуальный тур может быть использован в качестве рекламного и маркетингового инструмента для демонстрации инфраструктуры образовательной организации и привлечения потенциальных студентов, родителей или партнеров.

Вызовы и соображения:

1. Масштабируемость и обновления: со временем образовательные организации могут расширяться или подвергаться реконструкции. Виртуальный тур должен быть масштабируемым и легко обновляемым, чтобы учитывать изменения в физической среде.

2. Производительность и время загрузки: Виртуальный тур должен быть оптимизирован по производительности, чтобы обеспечить плавную и

отзывчивую работу пользователя. Эффективная загрузка больших 3D-моделей и мультимедийного контента имеет решающее значение.

3. Доступность и удобство использования: Виртуальный тур должен учитывать рекомендации по доступности, чтобы обеспечить возможность его использования людьми с ограниченными возможностями. Тестирование удобства использования и отзывы пользователей помогут выявить и решить проблемы, связанные с удобством использования.

4. Интеграция и обслуживание: Интеграция виртуального тура с существующими системами и платформами, а также обеспечение его постоянного обслуживания и обновления требуют координации и сотрудничества с соответствующими заинтересованными сторонами в образовательной организации.

В общем, предметная область 3D виртуального тура по образовательной организации предоставляет инновационный и захватывающий способ демонстрации учебных заведений. Он даст возможность для вовлечения, распространения информации и совершенствования коммуникации между студентами (включая преподавателей), коллегами по работе или потенциальными работниками.

Требования к функциональным характеристикам.

Программа должна обеспечивать возможность выполнения перечисленных ниже функций:

Функциональные требования:

- Создать виртуальный 3D-тур, точно представляющий университетский корпус и его объекты;
- Обеспечить удобную интерактивную платформу, позволяющую пользователям перемещаться по корпусу и получать доступ к соответствующей информации о каждом объекте;
- Клиент-серверная архитектура приложения;
- Создание и хранение информации;

- Добавление, редактирование и удаление информации;
- Использование гироскопа на смартфонах для перемещения камеры;

- Оптимизация под мобильные устройства;
- Поддержка VR и AR технологий

Нефункциональные требования:

- Сайт должен быть максимально удобным для использования и иметь интуитивно понятный интерфейс;
- Сайт должен работать быстро и при минимальных задержках;
- Сайт должен быть безопасным.

Спецификация

Клиентская часть:

- Версия JQuery: 3.6.4 или выше;
- Использование библиотеки Ajax для работы с API сервера;
- Дизайн сайта должен быть выполнен в едином стиле с учетом всех требований к интерфейсу.

Серверная часть:

- Использование Node.js для разработки серверной части;
- Разработка REST API для взаимодействия с клиентской частью;
- Использование фреймворка Strapi для работы с базой данных SQLite;

База данных:

- Использование SQLite для хранения информации;

Требования к надежности:

Сайт не должен содержать:

- Собирать данные о пользователе
- Информацию рекламного характера
- Информацию, запрещённую к распространению

Условия эксплуатации:

Программа должна быть проста в обращении и понятна среднему пользователю компьютера.

Требования к составу и параметрам технических средств

Для работы сайта с панорамами необходимы следующие параметры:

- Установленный браузер с поддержкой HTML 5, CSS, JS
- Подключение к интернету
- Процессор: Intel Celeron G1610 2.60 ГГц или аналогичный
- ОЗУ: 2 гб
- Свободного дискового пространства: 500мб
- Устройство под управлением OS Android 4.0 и выше, IOS 11.0 и

ниже.

Требования к информационной и программной совместимости

Требования к информационным структурам (файлов) на входе и выходе, а также к методам решения не предъявляются.

Специальные требования

Совместимость с мобильными устройствами: Виртуальный тур должен быть оптимизирован для мобильных устройств, чтобы обеспечить беспрепятственный и отзывчивый опыт для пользователей, просматривающих тур на смартфонах или планшетах. Пользовательский интерфейс и навигация должны быть адаптированы к небольшим экранам и сенсорному взаимодействию.

Интеграция виртуальной реальности (VR): при желании виртуальный тур может быть совместим с устройствами виртуальной реальности для обеспечения еще более захватывающего впечатления. Для этого необходима поддержка гарнитур и контроллеров VR, а также возможность навигации и взаимодействия в виртуальной среде с помощью технологии VR.

Оптимизация производительности: Виртуальный тур должен быть оптимизирован по производительности для обеспечения плавной навигации и

времени загрузки. Для повышения производительности можно использовать такие методы, как оптимизация активов, кэширование и потоковая передача.

Требования к программной документации

С программой должно поставляться руководство оператора.

Стадии и этапы разработки

Разработка должна быть проведена в пять стадий:

1. Разработка технического задания
2. Эскизный проект
3. Технический проект
4. Рабочий проект
5. Отладка, тестирование

На стадии «Техническое задание» должен быть выполнен этап разработки, согласования и утверждения настоящего технического задания.

На стадии «Эскизный проект» должны быть выполнены перечисленные ниже этапы работ:

- Предварительная разработка структуры входных и выходных данных
- Уточнение методов решения задачи
- Разработка общего описания алгоритма решения задачи

На стадии «Технический (и рабочий) проект» должны быть выполнены перечисленные ниже этапы работ:

- Разработка программы;
- Разработка программной документации;
- Испытания программы.

На стадии «Отладка или тестирование» должны быть выполнены перечисленные ниже этапы работ:

- Проведение испытаний.

Корректировка программы и программной документации по результатам испытаний

2. Проектирование и разработка виртуальной 3D-экскурсии по образовательной организации

2.1. Выбор средств разработки и архитектуры программного продукта

HTML: HTML — это стандартный язык разметки для создания структуры веб-страниц. Он обеспечивает основу для организации и представления контента в рамках 3D виртуального тура. HTML позволяет создавать семантические элементы и обеспечивает структурированный макет для других технологий.

CSS: CSS необходим для создания стиля и визуального улучшения 3D виртуального тура. Он позволяет настраивать цвета, шрифты, макеты и анимацию, обеспечивая визуально привлекательный и интересный пользовательский опыт. CSS обеспечивает гибкость для создания отзывчивого дизайна, который адаптируется к различным размерам экрана и устройствам.

JavaScript: JavaScript — это мощный язык программирования, который добавляет интерактивность и динамическую функциональность веб-страницам. В контексте 3D виртуального тура JavaScript можно использовать для создания интерактивных элементов, обработки взаимодействия с пользователем, анимации объектов и управления поведением тура. Это позволяет создать более захватывающий и интерактивный опыт для пользователей.

jQuery: jQuery — это популярная библиотека JavaScript, которая упрощает обход HTML-документов, обработку событий и манипулирование DOM. Она обеспечивает лаконичный синтаксис и кросс браузерную совместимость, что делает ее полезной для выполнения общих задач и взаимодействий в рамках виртуального тура. jQuery может ускорить разработку и оптимизировать усилия по кодированию.

Ajax: Ajax обеспечивает асинхронную связь между веб-браузером и сервером, позволяя беспрепятственно получать данные и обновлять содержимое без перезагрузки всей страницы. Это может быть полезно при загрузке динамического контента в виртуальном туре, например, при

получении дополнительной информации, изображений или мультимедийных ресурсов, что обеспечивает более плавное взаимодействие с пользователем.

Node.js: Node.js — это среда выполнения JavaScript на стороне сервера, которая облегчает разработку масштабируемых и высокопроизводительных веб-приложений. Она может быть использована для обработки логики на стороне сервера, управления хранением и получением данных, а также обеспечения коммуникационных возможностей в реальном времени, что делает ее подходящей для сложных 3D виртуальных туров, требующих обработки на стороне сервера или взаимодействия в реальном времени.

Bootstrap: Bootstrap — это широко используемый CSS-фреймворк, который предоставляет набор предварительно разработанных компонентов, сеток и стилей. Он предлагает систему отзывчивого дизайна, гарантируя, что виртуальный тур адаптируется и оптимально отображается на различных устройствах и размерах экрана. Bootstrap позволяет ускорить разработку, улучшить согласованность и обеспечить прочную основу для создания отзывчивых и визуально привлекательных интерфейсов.

A-Frame.js: A-Frame.js — это веб-фреймворк для создания виртуальной реальности (VR) с помощью HTML. Он упрощает создание VR-контента, предоставляя декларативный синтаксис и высокоуровневые абстракции. A-Frame.js интегрируется с популярными платформами и устройствами VR, позволяя проводить виртуальные экскурсии в захватывающей 3D-среде, повышая общую вовлеченность пользователей.

2.2. Разбиение задачи на подзадачи и определение алгоритмов решения

Основная задача содержит следующие подзадачи:

Планирование и исследование:

- **Собрать требования:** определить конкретные цели, особенности и целевую аудиторию для виртуального тура.
- **Провести исследование:** изучите существующие виртуальные туры и изучите лучшие практики создания захватывающего 3D опыта.

- Разработать план проекта: составить график, распределить ресурсы и определить основные этапы проекта.

Сбор и подготовка данных:

- Организовать и оптимизировать данные: очистить и обработать снятые данные, обеспечив их соответствующий формат и разрешение для визуализации в виртуальном туре.

- Создание 360-градусных панорам: сшивание нескольких изображений для создания бесшовных панорамных видов ключевых мест в организации.

Разработка интерфейса виртуального тура:

- Разработать пользовательский интерфейс: создать интуитивно понятный и удобный интерфейс для навигации по виртуальному туру, позволяющий пользователям исследовать различные места.

- Реализовать навигацию: обеспечить плавную навигацию в виртуальном туре, включая панорамирование, масштабирование и перемещение между различными областями.

Программирование и интеграция:

- Используя HTML, CSS и JavaScript: написать код для создания виртуального тура, используя HTML для структуры, CSS для стилизации и JavaScript для интерактивности и динамического поведения.

- Реализовать 3D-рендеринг: использовать такие библиотеки, как Three.js или A-Frame.js для рендеринга и отображения 3D-контента в виртуальном туре.

- Интегрировать внешние инструменты или API: использовать внешние инструменты, API или фреймворки (например, jQuery, Ajax или Node.js) для улучшения функциональности, интерактивности и обработки данных в виртуальном туре.

- Обеспечить кросс браузерную совместимость: протестировать и оптимизировать виртуальный тур, чтобы обеспечить его бесперебойную работу в различных веб-браузерах и устройствах.

Обеспечение качества и тестирование:

- Протестировать виртуальный тур на различных устройствах, браузерах и операционных системах, чтобы выявить и устранить любые проблемы совместимости или ошибки.

- Оценить впечатления пользователей: получить отзывы от пользователей или заинтересованных сторон, чтобы оценить удобство использования, навигацию и общее впечатление от виртуального тура.

- Оптимизировать производительность: доработать производительность виртуального тура, оптимизировав время загрузки, минимизировав использование ресурсов и обеспечив плавные переходы между различными местами.

Краткое описание средств разработки:

Visual Studio Code (VS Code) - популярный редактор исходного кода, разработанный компанией Microsoft. Он предоставляет легкую, но мощную среду для разработки программного обеспечения на различных платформах. Ниже приводится краткое описание среды разработки Visual Studio Code:

- Пользовательский интерфейс: VS Code имеет чистый и интуитивно понятный пользовательский интерфейс. Он состоит из боковой панели проводника файлов, центральной области редактора и различных панелей для вывода, отладки и расширений. Пользовательский интерфейс является настраиваемым, что позволяет разработчикам располагать и настраивать макет в соответствии со своими предпочтениями.

- Редактирование кода: VS Code предлагает богатый набор функций для редактирования кода. Он поддерживает подсветку синтаксиса для широкого спектра языков программирования и обеспечивает интеллектуальное авто завершение, фрагменты кода и форматирование кода.

Он также включает встроенную интеграцию Git для контроля версий и поддерживает совместную работу над кодом через Live Share.

- **Расширения и рынок:** Одним из ключевых достоинств VS Code является его расширяемость. Он имеет обширную экосистему расширений, созданных сообществом, что позволяет разработчикам улучшить опыт разработки и поддерживать различные языки программирования, фреймворки и инструменты. VS Code Marketplace обеспечивает централизованный центр для поиска и установки расширений.

- **Интегрированный терминал:** VS Code включает интегрированный терминал, позволяющий разработчикам выполнять задачи командной строки, не выходя из редактора. Он поддерживает различные среды оболочки и обеспечивает удобный способ запуска процессов сборки, выполнения команд и взаимодействия со средой разработки.

- **Отладка:** VS Code обеспечивает надежную отладку с поддержкой нескольких языков программирования. Он предоставляет точки останова, пошаговую отладку, проверку переменных и другие функции отладки, помогающие разработчикам выявлять и устранять проблемы в коде.

- **Совместная работа:** VS Code поддерживает совместную работу в режиме реального времени с помощью Live Share, который позволяет разработчикам делиться своей средой разработки с другими, что дает возможность совместного редактирования, отладки и обзора кода. Эта функция особенно полезна для удаленного сотрудничества или парного программирования.

- **Интегрированный контроль исходных текстов:** VS Code интегрируется с Git, обеспечивая беспрепятственный контроль исходного кода. Он предлагает такие функции, как история коммитов, управление ветвями и просмотр различий "бок о бок", что позволяет разработчикам легко управлять и отслеживать изменения в своей базе данных.

- Палитра команд и сочетания клавиш: VS Code включает палитру команд, которая обеспечивает быстрый доступ к различным командам и функциям. Он также предлагает широкий спектр сочетаний клавиш, которые могут значительно повысить производительность и оптимизировать рабочий процесс разработки.
- Кроссплатформенная поддержка: Visual Studio Code создан как кроссплатформенный продукт, работающий на Windows, macOS и Linux. Это позволяет разработчикам работать в предпочитаемой операционной системе и плавно переключаться между различными средами без ущерба для функциональности и согласованности.
- В целом, Visual Studio Code представляет собой мощную и универсальную среду разработки с большим количеством функций, широкими возможностями настройки и яркой экосистемой расширений, созданной сообществом. Она предоставляет разработчикам гибкий и эффективный инструмент для кодирования, отладки, совместной работы и контроля версий, что делает ее популярным выбором для разработки программного обеспечения для широкого спектра проектов и языков программирования.

2.3. Реализация клиентская часть

Ниже представлена реализация клиентской части.

Описание Index.html:

1. Структура HTML:

- HTML-документ начинается с `<!doctype html >` объявление и укажите язык как русский (`lang="ru"`).
- Раздел `<head>` содержит мета-теги для кодировки символов и настроек видового экрана.
- В заголовке страницы указано "Pano".
- Связаны внешние файлы CSS и JavaScript, включая Bootstrap CSS framework, пользовательскую таблицу стилей и необходимые библиотеки для Bootstrap, A-Frame и jQuery.

- Раздел "`<body>`" разделен на разделы верхнего колонтитула, содержимого и нижнего колонтитула.

2. Заголовок:

- Заголовок содержит панель навигации (`<nav>`) с элементом фирменного стиля (``) и кнопку переключения (`<button>`) для сворачивания навигационного меню на экранах меньшего размера.

3. Содержание:

- Раздел содержимого содержит 3D-сцену, отрисованную с использованием A-Frame, представленную элементом `<a-scene>`.

- Внутри `<a-scene>` элемент `<a-sky>` используется для отображения панорамного изображения по умолчанию, указанного атрибутом `src`.

4. Карта:

- Раздел карты представлен элементом `<div>` с идентификатором "renderMap".

- Он содержит список вкладок (`<ul class="nav nav-pills">`) и контейнер содержимого вкладок (`<div class="tab-content">`).

- Вкладки динамически генерируются на основе полученных данных и заполняются номерами этажей.

- Содержимое вкладки динамически загружается и отображает карты этажей и связанные с ними комнаты.

5. Нижний колонтитул:

- Раздел нижнего колонтитула пуст и не содержит никакого содержимого.

6. JavaScript-код:

- Код встроен в тег `<script>`.
- Он объявляет переменную `dataRooms` для хранения извлеченных данных о комнате.

- AJAX-запрос выполняется с использованием `$.ajax()` для получения данных о номере по указанному URL-адресу.
- При успешном извлечении полученные данные присваиваются переменной `dataRooms`.
- Определена функция `renderMain(el)`, которая вызывается при щелчке по элементу room.
- Внутри функции извлекается атрибут `value` щелкнутого элемента.
- Массив `dataRooms` повторяется для поиска соответствующего идентификатора комнаты.
- Если найдено совпадение, создается URL-адрес для соответствующего панорамного изображения.
- Атрибут `src` элемента `<a-sky>` обновляется новым URL-адресом, эффективно изменяя отображаемое панорамное изображение.
- Функция также регистрирует идентификатор выбранной комнаты и массив `dataRooms` на консоли.

Код настраивает структуру HTML, стилизует страницу с помощью Bootstrap, визуализирует 3D-сцену виртуального тура с помощью A-Frame и обрабатывает динамическую загрузку карт этажей и панорам комнат. Код JavaScript извлекает данные асинхронно и обновляет сцену на основе взаимодействий пользователя. Выглядит следующим образом. См. таблицу 5.1.1.

Таблица 5.1.1.

```
<!doctype html>
<html lang="ru">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Пано</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-
GLh1TQ8iRABdZLL603oVMWsktQOp6b7In1Zl3/Jr59b6EGGoI1aFkw7cmDA6j6gD"
crossorigin="anonymous">
```

```

    <link
href="https://slowbag.github.io/Diplom/src/content/css/style.css"></link>

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha1/dist/js/bootstrap.bundle.min.js" integrity="sha384-
w76AqPfDkMBDXo30jS1Sgez6pr3x5MlQ1ZAGC+nuZB+EYdgRZgiwxhTBtkF7CXvN"
crossorigin="anonymous"></script>
    <script
src="https://slowbag.github.io/Diplom/src/content/js/aframe.js"></script>
    <script
src="https://slowbag.github.io/Diplom/src/content/js/jquery.js"></script>
</head>
<body class="d-flex flex-column min-vh-100">
    <header style="z-index: 1;">
        <nav class="navbar navbar-expand-lg bg-body-tertiary">
            <div class="container">
                <div class="col-sm-3">
                    <a class="navbar-brand" href="#">Папо</a>
                    <button class="navbar-toggler" type="button" data-bs-
toggle="collapse" data-bs-target="#navbarNav" aria-controls="navbarNav" aria-
expanded="false" aria-label="Toggle navigation">
                        <span class="navbar-toggler-icon"></span>
                    </button>
                </div>
                <div class="col-sm-3"></div>
            </div>
        </nav>
    </header>
    <content>
        <a-scene style="position: static;">
            <a-sky
src="https://slowbag.github.io/Diplom/src/content/image/6561-dockalnd-
appart.jpg" rotation="0 -130 0"></a-sky>
        </a-scene>

        <!-- Map -->
        <div id="renderMap" class="opacity-75" style="position: fixed;
z-index: 1;background: gray;bottom: 0px;left: 0px;max-width: 20%;">
            <ul class="nav nav-pills mb-3" id="pills-tab"
role="tablist">

                </ul>
                <div class="tab-content" id="pills-tabContent">

                    </div>
            </div>

    </content>
    <footer class="mt-auto">

```

```

    </footer>
  </body>
  <script
src="https://slowbag.github.io/Diplom/src/content/js/index/index.js"></script>
  <script>

    let dataRooms;

    // ModelRooms
    $.ajax({
      url: recvestUrl + '/api/rooms/?populate=*',
      method: 'get',
      dataType: 'json',
      success: function(data)
      {
        dataRooms = data.data;
      }
    });

    async function renderMain(el)
    {
      let valueId = $(el).attr('val');

      for(let x in dataRooms)
      {
        let idRoom = dataRooms[x].id;
        let urlPanoram = recvestUrl +
dataRooms[x].attributes.Panorama.data.attributes.url;

        if(valueId == idRoom)
        {
          $("a-sky").attr("src",urlPanoram);
        }
      }

      console.log(valueId, );
      console.log(dataRooms);
    }
  </script>
</html>

```

Описание Index.js:

1.Объявление переменных:

- baseUrl: представляет базовый URL приложения.
- recvestUrl: представляет URL, используемый для выполнения

AJAX-запросов для получения данных.

2. Функция готовности документа:

- Код обернут в `$(function(){}),` что является сокращением для обработчика события готовности документа в jQuery. Он гарантирует, что код внутри будет выполнен, когда DOM полностью загрузится.
- AJAX-запрос - ModelFloors:
- AJAX-запрос выполняется к указанному URL (`recvestUrl + '/api/floors?populate=*'`) для получения данных об этажах.
- Свойство `url` запроса AJAX указывает конечную точку для получения данных об этажах.
- В случае успеха полученные данные сохраняются в переменной `dataFloors`.
- Функция `renderMap()` вызывается для отображения карты на основе полученных данных.

3. Функция `renderMap()`:

- Эта функция отвечает за рендеринг карты и связанных с ней комнат.
- Она инициализирует переменные для хранения идентификатора текущего этажа, изображения карты, номера этажа, описания и данных о комнатах.
- Она выполняет итерации по массиву `dataFloors` с помощью цикла `for...in` для обработки каждого этажа.
- Внутри цикла он извлекает соответствующую информацию для текущего этажа из массива `dataFloors`.
- Он определяет, является ли текущий этаж активной вкладкой и активной панелью вкладок, на основе индекса (`x`).
- Он создает URL-адрес для изображения карты, добавляя средний URL-адрес из атрибута карты, полученного из текущего этажа.
- Он добавляет новый элемент списка (``) к элементу `#pills-tab`, представляющему вкладку этажа. На кнопке отображается номер этажа.

- Он добавляет новую панель вкладок (<div>) к элементу #pills-tabContent, представляющему содержимое вкладки floor. Изображение карты отображается на панели вкладок.
- Он выполняет итерацию по данным комнат текущего этажа, используя вложенный for...in петля.
- Внутри вложенного цикла он извлекает соответствующую информацию для каждой комнаты (например, идентификатор комнаты, название и координаты) из массива комнат, полученного с текущего этажа.
- Он добавляет элемент <a> к текущей панели вкладок, представляющий собой маркер комнаты. Маркер позиционируется с помощью свойств CSS вверху и слева на основе координат комнаты.
- Маркер комнаты содержит изображение маркера и сохраняет идентификатор комнаты в качестве атрибута val.
- Событию onclick для маркера комнаты присвоено значение renderMain(this), которое запускает функцию renderMain() при нажатии на маркер комнаты.
- После завершения цикла функция записывает массив dataFloors в консоль.

Код получает данные об этажах через AJAX-запрос, итерирует данные для создания вкладок этажей и изображений карты, а также добавляет маркеры помещений с соответствующими обработчиками событий. Функция renderMain() отвечает за обработку события нажатия на маркеры помещений. См. таблицу 5.1.2.

Таблица 5.1.2.

```
let baseUrl = 'https://slowbag.github.io/Diplom/';
let recvestUrl = 'https://959e-5-165-156-247.ngrok-free.app';

let dataFloors;

$(function()
```

```

{
  // ModelFloors
  $.ajax({
    url: recvestUrl + '/api/floors?populate=*',
    method: 'get',
    dataType: 'json',
    success: function(data)
    {
      dataFloors = data.data;
      renderMap();
    }
  });

  async function renderMap()
  {
    let [id, map, numberFloor, Opisanie, rooms] = ['', '', '', ''];
    for(let x in dataFloors)
    {
      id = dataFloors[x].id;
      map = dataFloors[x].attributes.Map.data.attributes;
      numberFloor = dataFloors[x].attributes.NumberFloor;
      Opisanie = dataFloors[x].attributes.Opisanie;
      rooms = dataFloors[x].attributes.rooms.data

      let activeTab = (x == 0) ? 'nav-link active':'nav-link';
      let activeTabPane = (x == 0) ? 'tab-pane fade show active':'tab-pane fade';
      let imageUrl = recvestUrl + map.formats.medium.url;

      $('#pills-tab').append(`
        <li class="nav-item" role="presentation">
          <button class="${activeTab}" id="pills-${id}-tab" data-bs-toggle="pill" data-bs-target="#pills-${id}" type="button" role="tab" aria-controls="pills-${id}" aria-selected="true">${numberFloor}</button>
        </li>
      `)

      $('#pills-tabContent').append(`
        <div class="${activeTabPane}" id="pills-${id}" role="tabpanel" aria-labelledby="pills-${id}-tab" tabindex="0">
          
        </div>
      `)
      for(let y in rooms)
      {
        let idRooms = rooms[y].id;
        let nameRoom = rooms[y].attributes.NomerRoom;
        let pointX = rooms[y].attributes.PointX;
        let pointY = rooms[y].attributes.PointY;

```

```

        $("#pills-"+id).append(`
            <a style="position:absolute;z-index: 2; top:${pointY}px;
left:${pointX}px;cursor: pointer;" val="${idRooms}" onclick="renderMain(this)"
alt="${nameRoom}">
                
            </a>
        `);
    }
}
console.log(dataFloors)
}
});

```

2.4. Реализация серверной части

Ниже представлена реализация серверной части.

Описание admin.js:

Конфигурационный файл для приложения Node.js, использующий фреймворк Strapi. Он экспортирует объект, содержащий различные параметры конфигурации, основанные на переменных окружения.

auth:

- Представляет собой конфигурацию для настроек аутентификации.
- Свойство secret, которое устанавливается в значение переменной окружения ADMIN_JWT_SECRET. Используется для генерации и проверки JWT (JSON Web Token) для аутентификации администратора.

apiToken:

- Представляет собой конфигурацию для настройки токенов API.
- Свойство salt, которое устанавливается в значение переменной окружения API_TOKEN_SALT. Это значение используется для генерации безопасных API-токенов.

transfer:

- Представляет собой конфигурацию для настроек передачи.
- Свойство token, которое содержит дополнительные параметры конфигурации для токенов передачи.

Внутри свойства `token` есть свойство `salt`, которое устанавливается в значение переменной окружения `TRANSFER_TOKEN_SALT`. Это значение `salt`, используется для генерации безопасных маркеров передачи. См. таблицу 5.2.1.

Таблица 5.2.1.

```
module.exports = ({ env }) => ({
  auth: {
    secret: env('ADMIN_JWT_SECRET'),
  },
  apiToken: {
    salt: env('API_TOKEN_SALT'),
  },
  transfer: {
    token: {
      salt: env('TRANSFER_TOKEN_SALT'),
    },
  },
});
```

Описание `api.js`:

Код экспортирует объект конфигурации, который определяет различные настройки для REST API. Эти настройки направлены на управление поведением пагинации и расширение функциональности API.

`defaultLimit`: это свойство определяет количество элементов, возвращаемых по умолчанию на страницу при пагинации.

`maxLimit`: это свойство определяет максимальное количество элементов, которые могут быть запрошены на одной странице. Оно помогает предотвратить чрезмерное потребление ресурсов и потенциальные проблемы с производительностью

`withCount`: это булево свойство определяет, должен ли ответ API включать общее количество доступных элементов. При значении `true` ответ будет содержать счетчик, предоставляя дополнительную информацию об общем количестве элементов. Это может быть полезно для клиентских приложений, которым необходимо отображать информацию о пагинации. См. таблицу 5.2.2.


```
module.exports = {
  rest: {
    defaultLimit: 25,
    maxLimit: 100,
    withCount: true,
  },
};
```

Описание database.js:

Это файл конфигурации для подключения к базе данных в приложении Node.js, использующий фреймворк Strapi.

- Импортируется модуль `path` для обработки операций с путями к файлам.
- Конфигурация экспортируется как функция, принимающая в качестве параметра объект `env`. Этот объект `env` используется для доступа к переменным окружения.
- Извлекается значение переменной окружения `DATABASE_CLIENT`, которая определяет тип используемого клиента базы данных. Если значение не указано, то по умолчанию используется `sqlite`.
- Объект `connections` определяет различные конфигурации соединения с базой данных в зависимости от клиента базы данных.
- Для каждого поддерживаемого клиента базы данных (MySQL, MySQL2, PostgreSQL и SQLite) определяется объект соединения. Объект соединения содержит различные свойства, такие как `connectionString`, `host`, `port`, `database`, `user`, `password` и `ssl`. Эти свойства заполняются соответствующими значениями из переменных окружения, что позволяет настраивать их.
- Свойство `pool` в каждом объекте соединения определяет минимальное и максимальное количество соединений в пуле соединений.
- Оператор `return` возвращает объект, содержащий окончательную конфигурацию соединения с базой данных. Он включает свойство `client`,

которое устанавливается в значение переменной окружения `DATABASE_CLIENT`, и остальную конфигурацию соединения, полученную из объекта `connections` на основе выбранного клиента.

Свойство `acquireConnectionTimeout` устанавливает таймаут для получения соединения с базой данных, который извлекается из переменной окружения `DATABASE_CONNECTION_TIMEOUT`. Значение по умолчанию составляет 60000 миллисекунд (1 минута). См. таблицу 5.2.3.

Таблица 5.2.3.

```
const path = require('path');

module.exports = ({ env }) => {
  const client = env('DATABASE_CLIENT', 'sqlite');

  const connections = {
    mysql: {
      connection: {
        connectionString: env('DATABASE_URL'),
        host: env('DATABASE_HOST', 'localhost'),
        port: env.int('DATABASE_PORT', 3306),
        database: env('DATABASE_NAME', 'strapi'),
        user: env('DATABASE_USERNAME', 'strapi'),
        password: env('DATABASE_PASSWORD', 'strapi'),
        ssl: env.bool('DATABASE_SSL', false) && {
          key: env('DATABASE_SSL_KEY', undefined),
          cert: env('DATABASE_SSL_CERT', undefined),
          ca: env('DATABASE_SSL_CA', undefined),
          capath: env('DATABASE_SSL_CAPATH', undefined),
          cipher: env('DATABASE_SSL_CIPHER', undefined),
          rejectUnauthorized: env.bool(
            'DATABASE_SSL_REJECT_UNAUTHORIZED',
            true
          ),
        },
      },
    },
    pool: { min: env.int('DATABASE_POOL_MIN', 2), max:
env.int('DATABASE_POOL_MAX', 10) },
  },
  mysql2: {
    connection: {
      host: env('DATABASE_HOST', 'localhost'),
      port: env.int('DATABASE_PORT', 3306),
      database: env('DATABASE_NAME', 'strapi'),
      user: env('DATABASE_USERNAME', 'strapi'),
      password: env('DATABASE_PASSWORD', 'strapi'),
```

```

      ssl: env.bool('DATABASE_SSL', false) && {
        key: env('DATABASE_SSL_KEY', undefined),
        cert: env('DATABASE_SSL_CERT', undefined),
        ca: env('DATABASE_SSL_CA', undefined),
        capath: env('DATABASE_SSL_CAPATH', undefined),
        cipher: env('DATABASE_SSL_CIPHER', undefined),
        rejectUnauthorized: env.bool(
          'DATABASE_SSL_REJECT_UNAUTHORIZED',
          true
        ),
      },
    },
    pool: { min: env.int('DATABASE_POOL_MIN', 2), max:
env.int('DATABASE_POOL_MAX', 10) },
  },
  postgres: {
    connection: {
      connectionString: env('DATABASE_URL'),
      host: env('DATABASE_HOST', 'localhost'),
      port: env.int('DATABASE_PORT', 5432),
      database: env('DATABASE_NAME', 'strapi'),
      user: env('DATABASE_USERNAME', 'strapi'),
      password: env('DATABASE_PASSWORD', 'strapi'),
      ssl: env.bool('DATABASE_SSL', false) && {
        key: env('DATABASE_SSL_KEY', undefined),
        cert: env('DATABASE_SSL_CERT', undefined),
        ca: env('DATABASE_SSL_CA', undefined),
        capath: env('DATABASE_SSL_CAPATH', undefined),
        cipher: env('DATABASE_SSL_CIPHER', undefined),
        rejectUnauthorized: env.bool(
          'DATABASE_SSL_REJECT_UNAUTHORIZED',
          true
        ),
      },
    },
    schema: env('DATABASE_SCHEMA', 'public'),
  },
  pool: { min: env.int('DATABASE_POOL_MIN', 2), max:
env.int('DATABASE_POOL_MAX', 10) },
},
  sqlite: {
    connection: {
      filename: path.join(
        __dirname,
        '..',
        env('DATABASE_FILENAME', 'data.db')
      ),
    },
    useNullAsDefault: true,
  },
};

```

```
return {
  connection: {
    client,
    ...connections[client],
    acquireConnectionTimeout: env.int('DATABASE_CONNECTION_TIMEOUT',
60000),
  },
};
};
```

Описание middlewares.js:

Экспортирует массив, представляющий конфигурацию стека промежуточного ПО для приложения Strapi. Каждый элемент массива представляет уровень промежуточного ПО, который должен быть выполнен в указанном порядке.

- Оператор `'module.exports'` используется для экспорта конфигурации стека промежуточного ПО в качестве модуля.
- Массив начинается с серии строковых значений, представляющих встроенные промежуточные модули Strapi. К ним относятся:
 - `'strapi::errors'`: Промежуточное ПО для обработки ошибок и их форматирования.
 - `'strapi::security'`: Промежуточное ПО для обеспечения мер безопасности.
 - `'strapi::cors'`: CORS (Cross-Origin Resource Sharing) промежуточное ПО для контроля доступа из разных источников.
 - `'strapi::poweredBy'`: Промежуточное ПО для установки заголовка 'X-Powered-By'.
 - `'strapi::logger'`: Подпрограмма для протоколирования HTTP-запросов и ответов.
 - `'strapi::query'`: Подпрограмма для разбора и обработки параметров запроса.
 - `'strapi::body'`: Middleware для разбора и обработки тела запроса.
 - `'strapi::session'`: Middleware для обработки пользовательских сессий.

- 'strapi::favicon': Middleware для обслуживания фавикона.
- 'strapi::public': Middleware для обслуживания статических файлов из каталога 'public'.
- Промежуточное ПО 'strapi::cors' настроено с помощью объекта, содержащего свойства 'name' и 'config'. Свойство 'name' установлено в 'strapi::cors', а свойство 'config' содержит параметры конфигурации для CORS middleware.
- В объекте 'config' промежуточного ПО 'strapi::cors' устанавливаются следующие параметры:
 - - 'enabled': Булево значение, указывающее, включено ли промежуточное ПО CORS или нет.
 - - 'headers': Строка, представляющая разрешенные заголовки. В данном случае она имеет значение '*', разрешая все заголовки.
 - - 'origin': Массив разрешенных источников. В данном случае он разрешает запросы с 'http://localhost:1337', 'http://example2' и 'https://slowbag.github.io'. См. таблицу 5.2.4.

Таблица 5.2.4.

```
module.exports = [  
  'strapi::errors',  
  'strapi::security',  
  {  
    name: 'strapi::cors',  
    config: {  
      enabled: true,  
      headers: '*',  
      origin: ['http://localhost:1337', 'http://example2',  
        'https://slowbag.github.io']  
    }  
  },  
  'strapi::poweredBy',  
  'strapi::logger',  
  'strapi::query',  
  'strapi::body',  
  'strapi::session',  
  'strapi::favicon',  
  'strapi::public',  
];
```

Описание server.js:

Экспортирует объект конфигурации для приложения Strapi. Использует функцию `env` для получения переменных окружения и установки значений по умолчанию для определенных свойств.

- Оператор `module.exports` используется для экспорта объекта конфигурации в качестве модуля.
- Объект конфигурации содержит следующие свойства:
 - `host`: представляет собой адрес хоста для приложения Strapi. Он извлекает значение из переменной окружения `HOST` и принимает значение по умолчанию `0.0.0.0`, если переменная не установлена.
 - `port`: представляет собой номер порта, на котором будет запущено приложение Strapi. Он получает значение из переменной окружения `PORT` и по умолчанию принимает значение `1337`, если переменная не установлена.
 - `app.keys`: представляет собой ключи, используемые для защиты данных сессии в приложении Strapi. Он извлекает массив ключей из переменной окружения `APP_KEYS`.
 - `webhooks.populateRelations`: Это булев флаг, который определяет, должны ли веб-крючки заполнять отношения или нет. Он получает значение из переменной окружения `WEBHOOKS_POPULATE_RELATIONS` и принимает значение по умолчанию `false`, если переменная не установлена. См. таблицу 5.2.5.

Таблица 5.2.5.

```
module.exports = ({ env }) => ({
  host: env('HOST', '0.0.0.0'),
  port: env.int('PORT', 1337),
  app: {
    keys: env.array('APP_KEYS'),
  },
  webhooks: {
    populateRelations: env.bool('WEBHOOKS_POPULATE_RELATIONS', false),
  },
});
```

3. Тестирование виртуальной 3D-экскурсии по образовательной организации

Цель тестирования: Проверка работоспособности сайта и правильность ее выполнения.

Тестируется «Виртуальный тур»

Используемая техника:

ГП: GeForce GTX 1060 3GB

ЦП: AMD FX(tm)-6300 Six-Core Processor

Память: 12 GB RAM (11.92 GB RAM доступно)

Текущее разрешение: 1980 x 1080, 60Hz

Операционная система: Microsoft Windows 10 Pro

Подключенное оборудование:

- Проводная USB мышь;
- Проводная USB клавиатура

Сетевое соединение (витая пара, Wi-Fi);

Смартфон под управление Android 11.0

Таблица 6.1.

Протокол №1. Тестирование основных функций сайта

№	Описание теста	Ожидаемый результат	Полученный результат	Отметка о прохождении
1	Проверка загрузки сайта	Загрузка главной страницы	Главное страница загружена	ДА
2	Проверка камеры на перемещение	Камера перемещается	Камера перемещается	ДА
3	Проверка кнопок интерфейса на функционирование	Перемещение по колледжу, приближение и удаление камеры смена панорам	Смена ракурса камеры и панорамы	ДА
4	Проверка горячих точек	Смена панорамы на выбранную	Панорама сменилась	ДА

Продолжение таблицы 6.1

5	Проверка горячих точек на схеме колледжа на функционирование	Смена панорамы на выбранную	Панорама сменилась	ДА
6	Проверка гироскопа	Камера перемещается	Камера перемещается	ДА
7	Проверка всплывающего меню и функционирование кнопок	Меню появляется кнопки функционируют	Меню появилось Кнопки функционируют	ДА

Таблица 6.2

Протокол №2. Тестирование дополнительных функций программы

№	Описание теста	Ожидаемый результат	Полученный результат	Отметка о прохождении
1	Проверка на управление камерой через клавиатуру	Перемещение камеры	Камера перемещается	ДА
2	Проверка сайта на смартфоне	Сайт запустился интерфейс показывается корректно	Сайт запустился интерфейс показывается корректно	ДА

4. Расчет себестоимости и экономической эффективности виртуальной 3D-экскурсии по образовательной организации

4.1. Расчет трудоемкости разработки

Трудоемкость разных стадий и этапов разработки приведена в таблице 7.1.1.

Таблица 7.1.1.

№ п/п	Стадии и этапы разработки	Сроки разработки (в часах)
1	Разработка технического задания	10
2	Подготовительный этап	4
3	сбор информации	24
4	выбор объектного построения программы;	3
5	Обоснование и выбор средств разработки	2
6	Эскизный проект	9
7	Технический проект	15
8	Рабочий проект	80
9	Отладка, тестирование	30
ИТОГО		177

Определение затрат на разработку:

Цены за услуги создания панорамы по данным сайта <http://3dpano.pindora.com/prices.html> в среднем 2000. руб. за панораму, так как снималось на смартфон без специального оборудования, и квалификация дипломника минимальная используем для расчёта сумму 800. руб за панораму.

Общие затраты на оплату труда: Количество панорам*оплату за одну панораму = $800 \cdot 20 = 16000$ руб.

4.2. Расчет затрат на электроэнергию

Мощность ПК – 65Вт. Стоимость 1кВт – 3.11 рублей.

Затраты на электроэнергию: потребляемая мощность ПК*время работы*стоимость 1 кВт = $0,065 \text{ кВт} \cdot 177 \cdot 3.11 = 35.7$ рублей

4.3 Составление сметы всех затрат

Схема всех затрат в таблице 7.3.1

Таблица 7.3.1.

№ п/п	Наименование статей затрат	Сумма, руб.
1	Оплата труда	16000
2	Затраты на электроэнергию	35.7
ИТОГО Итого		16035.7

Таким образом, стоимость разработки проекта составляет 16035.7 рубля.

Заключение

Для студентов и преподавателей есть большой потенциал в использовании 3D виртуальных туров, которые помогают им эффективно взаимодействовать с информационным и образовательным контентом. Уникальным и захватывающим способом вовлечения студентов и школьников в образовательный контент являются не только интерактивные выступления, но и виртуальные учебные туры. Интерактивная визуализация образовательных концепций поможет студентам лучше запомнить, освоить, осмыслить и понять необходимую информацию, сформировать на своём рабочем месте навык к обучению и самосовершенствоваться в учебном процессе. Виртуальные туры имеют еще одну большую гибкость, отсутствующую у традиционных методов обучения. Студенты и преподаватели могут получить доступ к рекламному контенту и услугам интернета в любое время и из любого места, где они захотят это сделать (в том числе удаленно). Также, использование мультимедийного контента в виртуальных турах, такого как видео и аудиоматериалы позволяет студентам учиться самостоятельно и в полной мере адаптироваться к современным технологиям обучения, а также адаптироваться к современности согласно их индивидуальному стилю обучения. В целом, 3D виртуальные туры имеют большой потенциал в плане увеличения социальной вовлеченности учащихся, оптимизации процесса обучения и понимания содержания, развития умений и навыков, а также образовательного контента. Именно поэтому их считают наиболее ценным инструментом для преподавателей специальных предметов.

Список использованных источников

- 1) Николай П., Владимир Д. HTML, JavaScript, PHP и MySQL. Джентльменский набор Web-мастера. / Николай П., Владимир Д. – Изд 4-е. – СПб.: БХВ-Петербург, 2020 - 288 с.
- 2) Дронов В. HTML 5, CSS 3 и Web 2.0. Разработка современных Web-сайтов / Дронов В. – СПб.: БХВ-Петербург, 2021 - 355 с.
- 3) Лоусон Б., Шарп Р. Изучаем HTML 5 / Лоусон Б., Шарп Р. – СПб.: Питер, 2021 - 145 с.
- 4) Чак М., Бил К. HTML 5, HTML и XHTML. Подробное руководство / Чак М., Бил К. – СПб.: Символ-Плюс, 2021 - 405 с.
- 5) Бен Х. HTML и CSS. Путь к совершенству / Бен Х. – СПб.: Питер, 2021 - 224 с.
- 6) Дэвид С. М. HTML и CSS. Путь к совершенству / Дэвид С. М. – СПб.: Питер, 2020 - 504 с.
- 7) Дэвид С. М. JavaScript и jQuery. Исчерпывающее руководство / Дэвид С. М. – СПб.: Эксмо, 2020 - 250 с.
- 8) Джон Д. HTML и CSS. Разработка и дизайн веб-сайтов / Джон Д. – СПб.: Эксмо, 2020 - 300 с.
- 9) Дэвид Ф. JavaScript. Карманный справочник / Дэвид Ф. – СПб.: Вильямс, 2019 - 140 с.
- 10) Эрик А. М. CSS. Карманный справочник / Эрик А. М. – СПб.: Вильямс, 2018 - 188 с.
- 11) Дженнифер Н. Р. HTML5. Карманный справочник / Дженнифер Н. Р. – СПб.: Вильямс, 2020 - 321 с.
- 12) Дженнифер Н. Р. HTML5. Карманный справочник / Дженнифер Н. Р. – СПб.: Вильямс, 2021 - 188 с.
- 13) Уроки по krapo. [Электронный ресурс] // 1panorama.ru — URL: 1panorama.ru/uroki-po-krapo-3s (дата обращения: 19.03.2023).

14) Курцев Д. Как сделать виртуальную 3D экскурсию по учебному заведению. [Электронный ресурс] // kurets.ru — URL: kurets.ru/graphics/99-kak-sdelat-virtualnuyu-ekskursiyu (дата обращения: 10.05.2023).