

УДК 004.9:614.842.6

Жихарев М.Н.,

студент,

4 курс, факультет «Информационные технологии и программная

инженерия»

Санкт-Петербургский государственный университет телекоммуникаций

им. проф. М. А. Бонч-Бруевича

Россия, г. Санкт-Петербург

Научный руководитель: Мусаева Т.В.,

кандидат технических наук, доцент

доцент кафедры «Информатики и компьютерного дизайна»

Санкт-Петербургский государственный университет телекоммуникаций

им. проф. М. А. Бонч-Бруевича

Россия, г. Санкт-Петербург

ИНТЕРАКТИВНЫЙ КАТАЛОГ АВТОНОМНЫХ СИСТЕМ ПОЖАРОТУШЕНИЯ

Аннотация: Статья посвящена разработке интерактивного каталога технической продукции на примере автономных систем пожаротушения. Рассмотрена проблема недостаточной наглядности традиционных плоских каталогов для сложных инженерных изделий. Предложен подход, объединяющий серверную часть на ASP.NET Core, клиентский интерфейс на React и трёхмерную визуализацию на базе Three.js/React Three Fiber. Описана архитектура приложения FireCatalog, особенности интеграции gLTF-моделей, а также механизмы аутентификации, разграничения ролей и потоковой загрузки больших файлов. Приводятся результаты тестирования, подтверждающие работоспособность и практическую эффективность решения.

Ключевые слова: интерактивный каталог, трёхмерная визуализация, glTF, Blender, ASP.NET Core, React, Three.js, инженерная продукция, системы пожаротушения.

Annotation: *The article is devoted to the creation of an interactive catalog of technical products using the example of autonomous fire extinguishing systems. The problem of insufficient visibility of traditional flat catalogs for complex engineering products is considered. An approach combining a server side based on ASP.NET Core, a React client interface and 3D visualization powered by Three.js/React Three Fiber is proposed. The architecture of the FireCatalog application, features of integration of glTF models created in Blender, as well as mechanisms for authentication, role differentiation and streaming upload of large files are described. Test results confirming the operability and practical efficiency of the solution are given.*

Key words: *interactive catalog, 3D visualization, glTF, Blender, ASP.NET Core, React, Three.js, engineering products, fire extinguishing systems.*

Современные производственные компании всё чаще сталкиваются с ограничениями традиционных способов представления технической информации. Бумажные чертежи, статичные фотографии и разрозненные PDF-файлы не дают целостного объёмного образа изделия, что приводит к ошибкам при проектировании, монтаже и эксплуатации. Особенно остро эта проблема проявляется для изделий со сложной внутренней структурой, к которым относятся автономные системы пожаротушения. Пользователю необходимо не только знать характеристики, но и видеть конструкцию, расположение узлов, габаритные пропорции и взаимосвязи элементов.

Современные веб-технологии позволяют перенести трёхмерную визуализацию в браузер, превратив каталог продукции из пассивного справочника в интерактивную среду. Однако существующие САД-

ориентированные платформы (SolidWorks PDM, Autodesk Vault) избыточно сложны для непроектировщиков, а также дорогостоящих лицензий, установки специализированного ПО. Простые электронные каталоги с карточками товаров не поддерживают 3D-контент. Таким образом, ниша доступного веб-каталога с трёхмерным представлением инженерной продукции остаётся незаполненной.

Целью работы является разработка интерактивного каталога автономных систем пожаротушения, который объединяет карточки изделий, техническую документацию и трёхмерные модели в единой браузерной среде и повышает наглядность представления инженерной продукции по сравнению с традиционными статичными каталогами. Практической основой выбрано клиент-серверное приложение FireCatalog, созданное автором с нуля. Серверная часть реализована на ASP.NET Core, для клиентской части выбрана библиотека React с применением React Three Fiber и Drei для 3D-визуализации. Модели изделий созданы в Blender и экспортировались в бинарный формат GLB, который упаковывает геометрию, текстуры и материалы в один файл и оптимизирован для передачи по сети [1, с. 3].

Архитектура системы устроена так, что трёхмерная модель технически не является отдельным, особым объектом. Она хранится в той же базе данных и загружается через тот же API, что и обычные PDF-файлы или изображения. На уровне кода модель - это просто ещё один файл, прикрепленный к карточке изделия, только с другим типом (GLB/GLTF). За счёт этого модель не находится где-то сбоку, а естественно встраивается в карточку продукта наравне с описанием и документацией

Файл модели хранится в общей таблице ProductFiles наряду с PDF-документами и изображениями, имея лишь дополнительный признак типа (GLB/GLTF). Это позволило избежать выделения 3D-контента в отдельную сущность и сохранить единообразие API: клиент запрашивает список файлов продукта, а затем сам фильтрует модели для передачи в компонент

просмотрщика. Такая логика обеспечила связность каталога – описание, чертежи и трёхмерный объект ссылаются на один и тот же идентификатор продукта и доступны в одном интерфейсе.

Значительное внимание при реализации было уделено проблемам, возникающим при работе с тяжёлыми GLB-файлами. Стандартные ограничения Kestrel на размер тела запроса (по умолчанию около 30 МБ) не позволяли загружать модели весом 150 МБ и более. В конфигурации сервера параметр `MaxRequestBodySize` был установлен в `null`, а лимиты `FormOptions` увеличены до `long.MaxValue` [2, с. 1]. Для клиентской отправки больших файлов вместо `fetch` задействован `XMLHttpRequest`, позволяющий отслеживать прогресс загрузки. Ещё одна проблема была связана с несовпадением форматов данных между клиентом и сервером: клиент отправлял тип файла как строку (например, «Glb»), в то время как сервер ожидал числовой код перечисления `ProductFileType`. Автоматическое преобразование строки в `enum` вызывало ошибки десериализации, из-за которых запрос прерывался. Чтобы исключить подобные сбои, контроллер принимает тип файла как обычную строку, а затем вручную выполняет `Enum.TryParse`. Это простое решение полностью устранило нестыковку форматов и сделало обработку надёжной.

Компонент трёхмерного просмотрщика (`ProductViewer`) выполняет асинхронную загрузку модели через хук `useGLTF`, после чего вычисляет `bounding box` сцены, центрирует и масштабирует объект так, чтобы он занял предсказуемую часть экрана [3, с. 1-4]. Такой подход решил проблему значительного координатного смещения моделей, полученных из CAD-систем: из-за того что геометрический центр объекта мог находиться на большом удалении от начала координат, модель при начальной загрузке оказывалась вне поля зрения камеры. Управление камерой построено на `OrbitControls` с настроенными ограничениями по зуму и вращению. Имеющиеся в модели анимации не запускаются автоматически, а

активируются кликом пользователя и проигрываются однократно (режим LoopOnce), что исключает раздражение при частой работе с каталогом.

Система поддерживает пять ролей пользователей, выделенных на этапе анализа целевой аудитории: администратор, инженер, конечный потребитель, гость выставки и аудитор. Аутентификация построена на JWT-токенах, права доступа проверяются как на уровне API (policy-based авторизация), так и на уровне интерфейса (условный рендеринг). Такой двухуровневый контроль гарантирует, что инженер не сможет удалить модель даже прямым REST-запросом, а гость выставки не увидит ни панели загрузки файлов, ни внутренней документации. На рисунке 1 представлен вариант каталога для посетителей выставки, без какой-либо специфической документации.

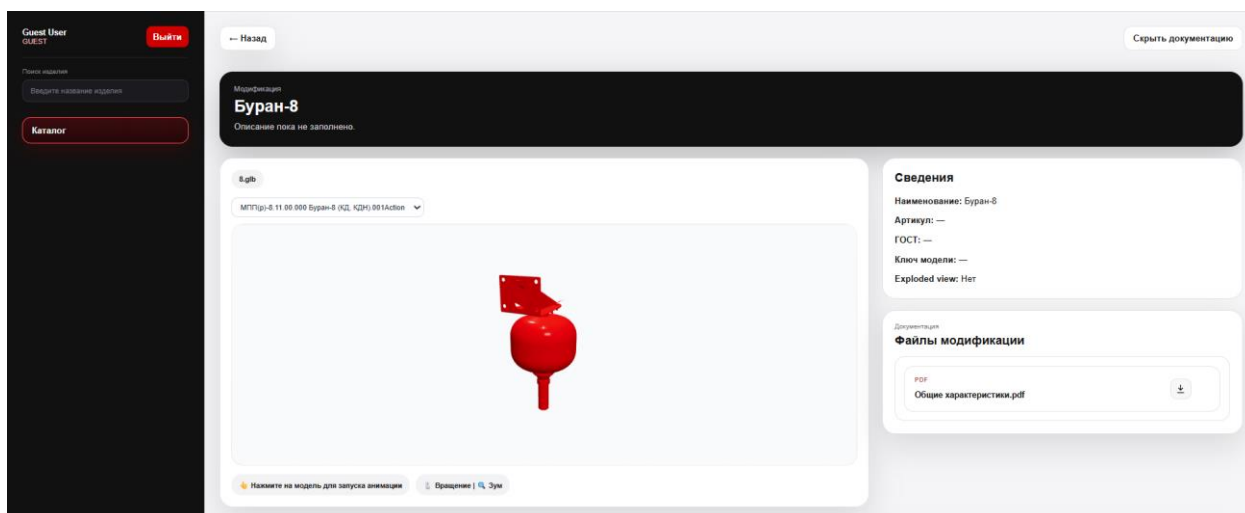


Рисунок 1. Выставочный вариант каталога для посетителей

Тестирование подтвердило работоспособность и стабильность решения. Модель весом 12 МБ загружается в среднем за 2-3 секунды, модель 161 МБ за 10-12 секунд с отображением индикатора прогресса. При быстрых переходах между карточками корректно обрабатывают CancellationToken на сервере и проверка монтирования компонента на клиенте, предотвращающая утечки памяти и визуальные артефакты.

Практическая ценность разработанного каталога заключается в сокращении времени на поиск и сопоставление информации. Сборщик видит

узел до того, как возьмёт деталь в руки, монтажник понимает пространственное положение элементов без восстановления по трём проекциям, маркетолог может демонстрировать изделие в разрезе без транспортировки физического образца. При этом система не требует установки дополнительного ПО - достаточно браузера. Всё это позволяет рассматривать FireCatalog как готовый к внедрению инструмент цифровой презентации инженерной продукции.

Дальнейшее развитие проекта может идти по пути внедрения сжатия Draco для ускорения загрузки моделей, реализации полноценного гостевого режима для выставок и адаптации интерфейса под мобильные устройства, используемые непосредственно на производственных участках.

Для создания карточных инженерных каталогов (каталожных листов, технических картотек) в России и странах СНГ используются стандарты, устанавливающие общие требования к составу данных и их оформлению. В частности, ГОСТ 33353.3-2019 регламентирует содержание и правила разработки каталожных листов продукции для государственных нужд и задаёт унифицированный набор сведений, обязательный для представления промышленного изделия [4, с. 5-10]. Однако подобные стандарты описывают плоскую структуру документа, ориентированную на текстовые и табличные данные, и не учитывают возможностей интерактивного представления. Разработанный каталог FireCatalog наследует логику обязательных реквизитов изделия (наименование, артикул, обозначение ГОСТ, технические характеристики), но при этом дополняет каталожную карточку трёхмерной моделью, доступной для непосредственного взаимодействия в браузере. Такой подход позволяет совместить формальные требования к структуре данных с возможностями современной веб-визуализации.

Использованные источники:

1. Khronos Group. glTF — Runtime 3D Asset Delivery [Электронный ресурс]. URL: <https://kcoley.github.io/glTF/> (дата обращения: 07.04.2026).
2. Microsoft. File uploads in ASP.NET Core [Электронный ресурс]. URL: <https://learn.microsoft.com/en-us/aspnet/core/mvc/models/file-uploads> (дата обращения: 10.04.2026).
3. Discover three.js. Load 3D Models in glTF Format [Электронный ресурс]. URL: <https://discoverthreejs.com/book/first-steps/load-models/> (дата обращения: 06.05.2026).
4. ГОСТ 33353.3-2019. Единая межгосударственная система. Каталогизация продукции для федеральных государственных нужд. Каталожные листы продукции. Содержание и правила разработки. — М.: Стандартинформ, 2019. — 20 с.