**Министерство цифрового развития, связи и массовых коммуникаций Российской Федерации**

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«Московский технический университет связи и информатики»**

Кафедра математической кибернетики и информационных технологий

**Отчёт по практической работе**

на тему «Разработка Web-сайта по API фильмов»

по дисциплине «Основы UI и UX дизайна»

Выполнил студент группы БВТ2001

Щавлев Константин Владимирович.

Содержание

[1. Введение 3](#_Toc154155472)

[1.1. Цель работы 3](#_Toc154155473)

[1.2. Задания 3](#_Toc154155474)

[2. Теоретическая часть 3](#_Toc154155475)

[3. Выполнение 5](#_Toc154155476)

[1. Конструирование API. 5](#_Toc154155477)

[2. Структура Frontend части приложения 6](#_Toc154155478)

[3. Интеграция с backend частью 7](#_Toc154155479)

[4. Верстка страниц и компонентов 7](#_Toc154155480)

[5. Скриншоты Frontend части. 9](#_Toc154155481)

[6. Вывод 11](#_Toc154155482)

1. Введение
   1. Цель работы

Необходимо разработать SPA исходя из описания задачи.

* 1. Задания
* Имеется API со списком компьютерных комплектующих, с которым нужно наладить взаимодействие;
* Необходимо визуализировать данные полученные от API в виде таблицы, содержащей только важную информацию о компьютерного комплектующего;
* Для написания сайта необходимо использовать фреймворки React и NextJS;
* По возможности избегать перезагрузки страницы;

1. Теоретическая часть

Next.js — открытый JavaScript фреймворк, созданный поверх React.js для создания веб-приложений, созданный компанией Vercel (ранее ZEIT). Фреймворк был предназначен для решения проблемы React.js, связанной с отрисовкой приложения на стороне сервера - SSR. Работает на сервере и в браузере.

React Query — отличная библиотека хуков для управления запросами данных, которая полностью избавляет от необходимости помещать удаленные данные в глобальное состояние. Вам просто нужно указать библиотеке, где вы хотите получить данные, и она будет обрабатывать кеширование, фоновые обновления и аннулирование данных без какого-либо дополнительного кода или конфигурации.

React Query также устраняет необходимость использовать хуки useState и useEffect и заменяет их на несколько строк логики React Query. В конечном итоге это определенно поможет вам поддерживать приложение.

На JavaScript выполняется большая часть интерактивных элементов на сайтах и в мобильных приложениях. JavaScript отлично работает с HTML/CSS и интегрирован основные браузеры на рынке. Чистый JavaScript используется в вебе, а для общего применения JavaScript разработчики используют различные среды выполнения, например, Node.js.

Node.js — это среда выполнения кода JavaScript. Она позволяет использовать JavaScript как язык программирования общего назначения: создавать на нем серверную часть и писать полноценные десктопные приложения.

Основа Node.js — движок V8. Этот движок был разработан Google и используется в браузере Google Chrome. Он компилирует код JavaScript в машинный код, который понимает процессор. Однако, чтобы сделать из JavaScript язык общего назначения, одного движка недостаточно. Так, например, для создания серверной части нужно, чтобы язык умел работать с файлами, сетью и т.п. Для решения этой проблемы разработчики добавили к V8 дополнительные возможности, с помощью своего кода и сторонних библиотек. В итоге у них получился инструмент, который превращает JavaScript в язык общего назначения.

Node.js стала популярна среди разработчиков благодаря возможности создавать серверную и клиентскую часть на одном языке, скорости работы и NPM.

1. Выполнение
2. Конструирование API.

В качестве API была создана собственная RESTAPI – NewCatalog. В ней был реализован парсер страниц крупнейших ритейлеров компьютерной техники (Citilink, DNS, Regard, Online-Trade), хранение и извлечение данных из базы данных SQLite/PosgreSQL. Так же данная API была упакована в docker-контейнер. Структура API представлена на рисунке 1.

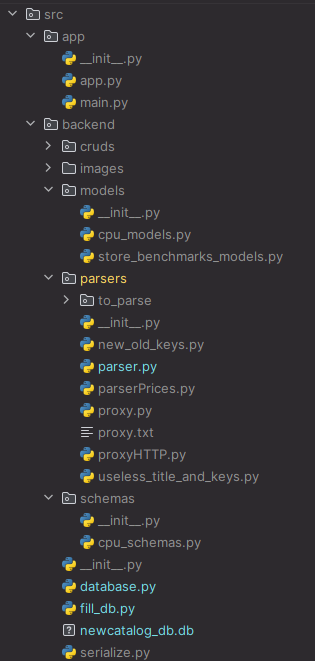


Рисунок 1. Структура проекта RestAPI NewCatalog.

1. Структура Frontend части приложения

Frontend часть приложения имеет следующую структуру:

1. Компоненты (components) – различные целостные структуры UI, которые могут быть переиспользованы неограниченное количество раз в различных частях приложения.
2. Страницы (pages) – страницы, которые и используют различные компоненты. Так же, данный объект содержит подобъект интеграции с Backend-частью - api
3. Публичные объекты (public) – здесь хранятся различные SVG иконки для категорий.

Структура проекта представлена на рисунке 2.

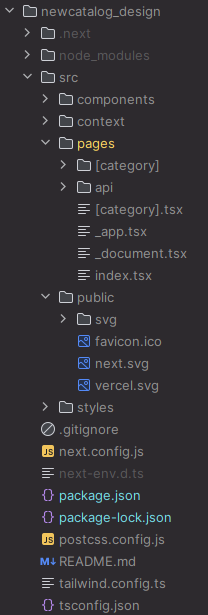


Рисунок 2. Структура Frontend части приложения.

1. Интеграция с backend частью

За данный аспект приложения отвечает файл src/pages/api/api.ts. Т.к. ответ от RestAPI передается на страницы запросов в виде словарей, то были реализованы два интерфейса: *ApiResponse* и *InnerDict*. Также были реализованы две функции: *fetchData* и *fetchModelData*. Код файла изображен на рисунке 3.

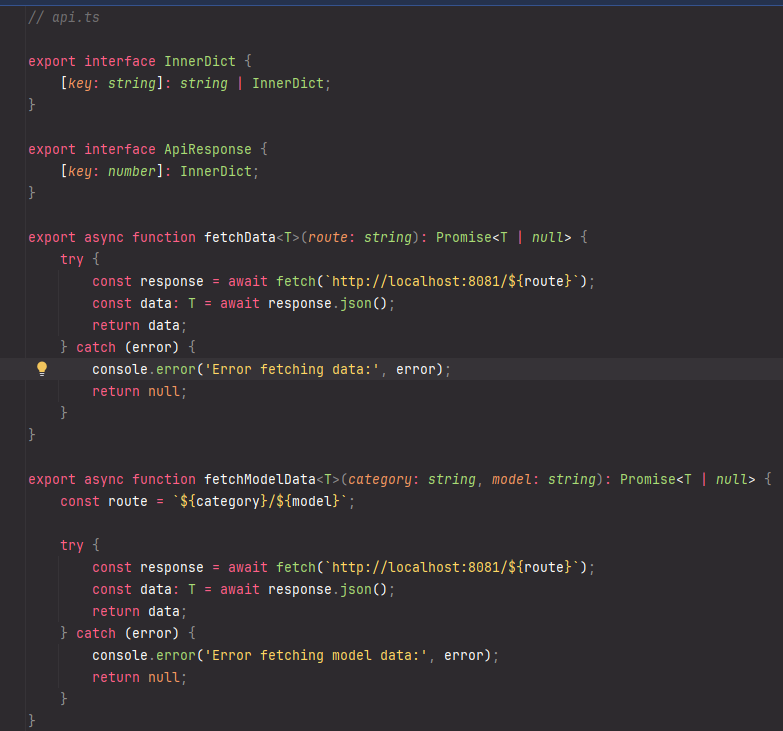


Рисунок 3. Интеграция с Backend частью приложения.

1. Верстка страниц и компонентов

Было реализовано 5 компонентов, а именно:

1. Botter
2. Header
3. Product
4. ProductDetails

Их используют следующие страницы:

1. Index.tsx
2. [category].tsx
3. [category]/[model].tsx

Скриншот кода Product и [category]/model.tsx представлены на рисунках 4-5.



Рисунок 4. Верстка компонента Product.

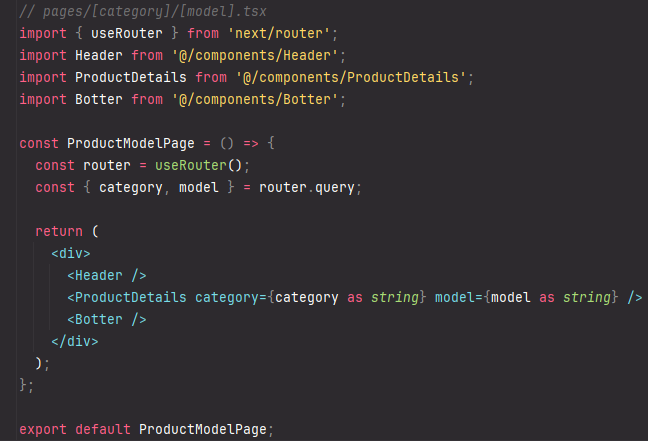


Рисунок 5. Верстка страницы [category]/[model].tsx

1. Скриншоты Frontend части.

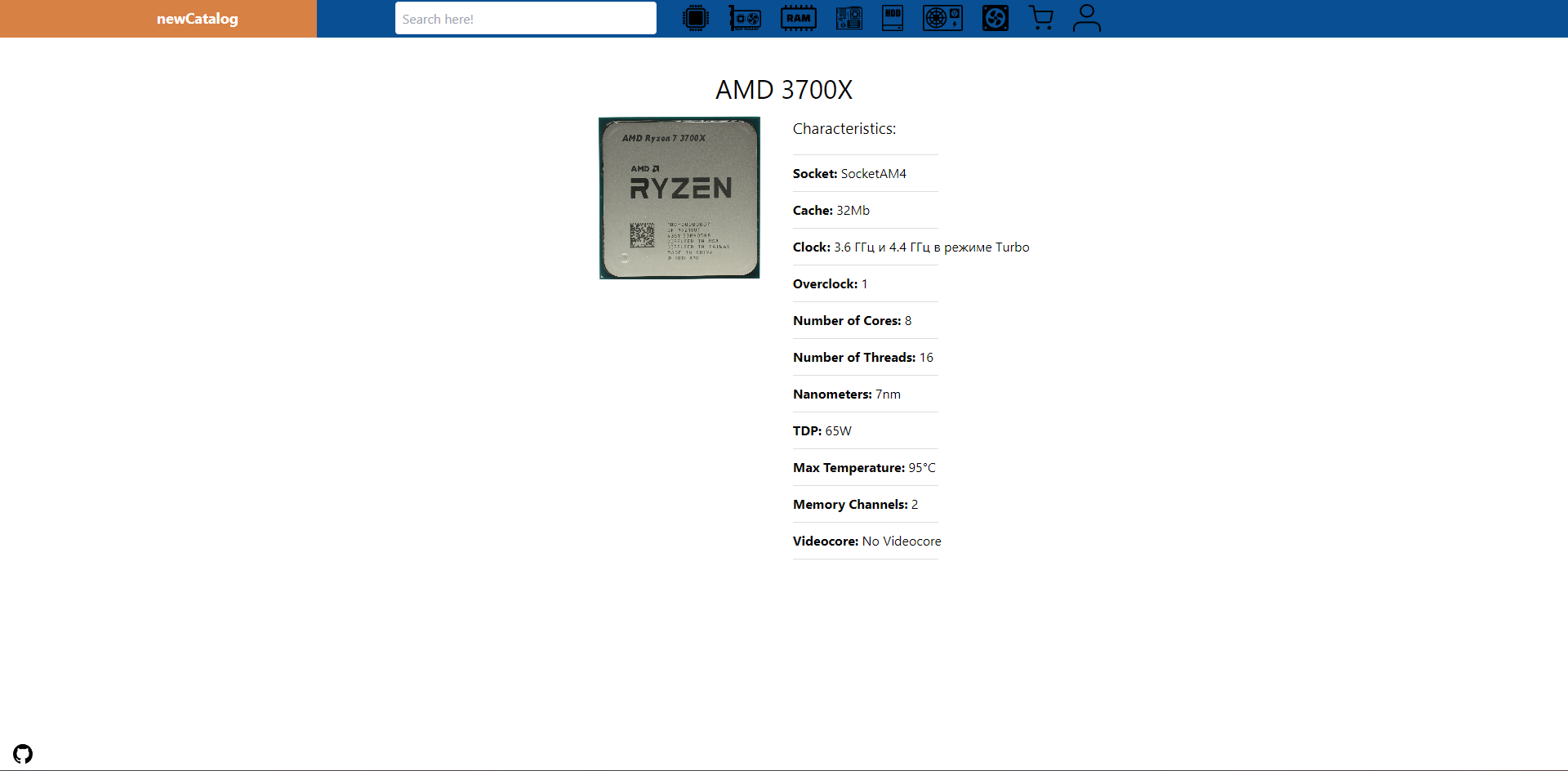


Рисунок 6. Скриншот детальной информации о компьютерном комплектующем.

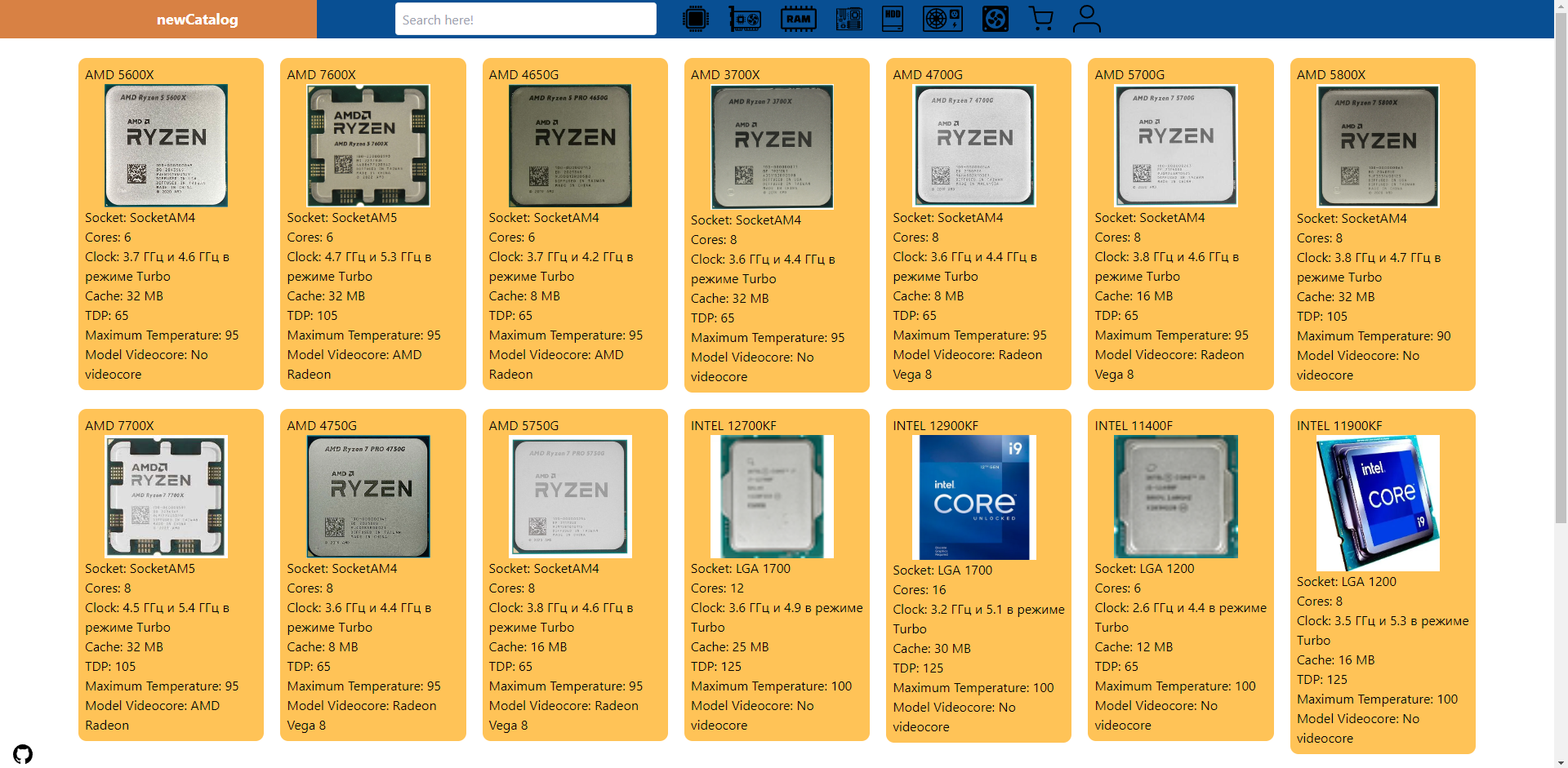


Рисунок 7. Скриншот «витрины» компьютерных комплектующих.

1. Вывод

В ходе проделанной работы мною был реализован веб-сайт с использованием фреймворков React и NextJS. Веб-сайт соответствует многим современным тенденциям в разработке и использует наиболее актуальный стек технологий и принципов программирования. Также реализованный сайт имеет вполне современный дизайн с адаптивной вёрсткой, что позволяет пользоваться им не только в браузере ноутбука или компьютера, но и в браузере мобильного устройства.