САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Дисциплина: Бэк-энд разработка

Отчет

Лабораторная работа 2

Выполнили:

Никитин Павел
Жаров Александр

Группа

К33402

Проверил:
Добряков Д. И.

Санкт-Петербург

2024 г.

Задача

В рамках данной лабораторной работы Вам предложено выбрать один из нескольких вариантов. Выбранный вариант останется единым на весь курс и будет использоваться в последующих лабораторных работах.

По выбранному варианту необходимо будет реализовать RESTful API средствами express + typescript (используя ранее написанный boilerplate).

Ход работы

Для этой работы мы выбрали вариант с магазином одежды. Перед началом разработки мы определили базовые модели и api методы, которые мы хотим видеть в нашем приложении:

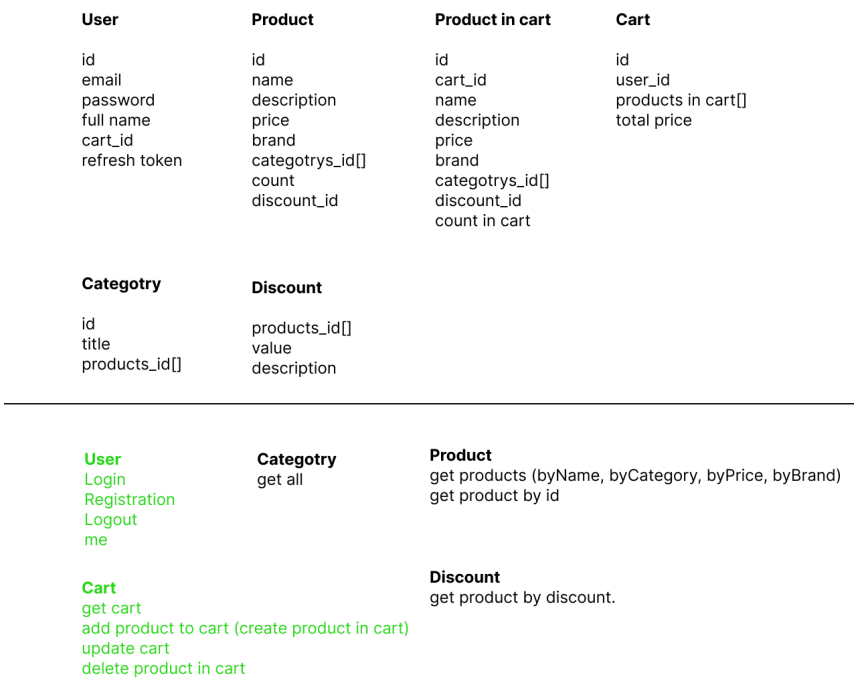

Рисунок 1 - концепция приложения

В качестве boilerplate приложения мы взяли мою первую лабораторную, по следующим причинам:

1. Интеграция с prisma - мы получаем готовую админку приложения, prisma studio, а также удобную авто генерируемую типизацию.

2. Nest задает удобную модульную структуру, позволяющий в будущем разбить проект на микросервисы.
3. Удобная интеграция с swagger - нету единого решения как использовать swagger и express, это хорошая альтернатива, для того чтобы избежать использования сомнительных библиотек.

При создании схемы для базы данных, мы хотели имплементировать все виды связей:

```
You, 3 дня назад | 1 author (You)
model User {
  id           Int     @id @default(autoincrement())
  email        String  @unique
  password     String
  fullName     String?
  cart         Cart?
  cartId       Int?
  refreshToken String?
}

You, 3 дня назад | 1 author (You)
model Cart {
  id             Int             @id @default(autoincrement())
  User           User            @relation(fields: [userId], references: [id])
  totalPrice     Int
  productsInCart ProductInCart[]
  userId         Int             @unique
}

You, 3 дня назад | 1 author (You)
model Product {
  id          Int        @id @default(autoincrement())
  title       String
  description String
  price       Int
  categories  Category[]
  Discount    Discount?  @relation(fields: [discountId], references: [id])
  discountId  Int?
}

You, 3 дня назад | 1 author (You)
model ProductInCart {
  id        Int   @id @default(autoincrement())
  productId Int
  Cart      Cart? @relation(fields: [cartId], references: [id])
  cartId    Int?
  count     Int
}

You, 3 дня назад | 1 author (You)
model Category {
  id       Int       @id @default(autoincrement())
  title    String
  Products Product[]
}      You, 3 дня назад • feature: add bd models

You, 3 дня назад | 1 author (You)
model Discount {
  id          Int       @id @default(autoincrement())
  value       Int
  description String
  Products    Product[]
}
```

Рисунок 2 - Prisma schema

One-to-one - пользователь и корзина, т.е. одна конкретная корзина, может иметь только одного пользователя.

One-to-many - корзина и продукты в корзине, одна корзина соответствует многим товарам, но у каждого товара одна корзина

Many-to-many - товар и категории, у товара может быть много категорий и у категории может быть много товаров

Код в модулях написан по структуре boilerplate:
1. controller - для обработки API запросов
2. service - для реализации основной логики
3. module - для объединения всех составных частей модуля.
4. dto - data to transfer models

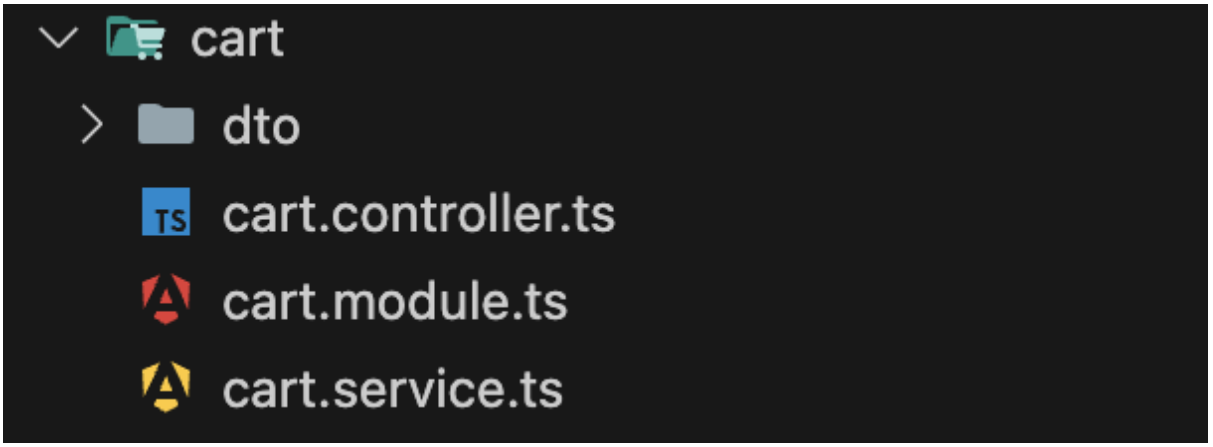

Рисунок 3 - пример модуля корзины

Для совместной разработки мы использовали расширение VSCode Liveshare

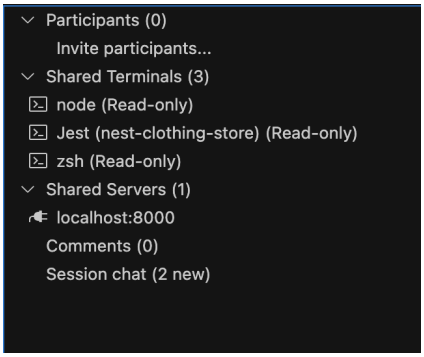

Рисунок 4 - настройки LiveShare

В качестве результата мы получили следующее:

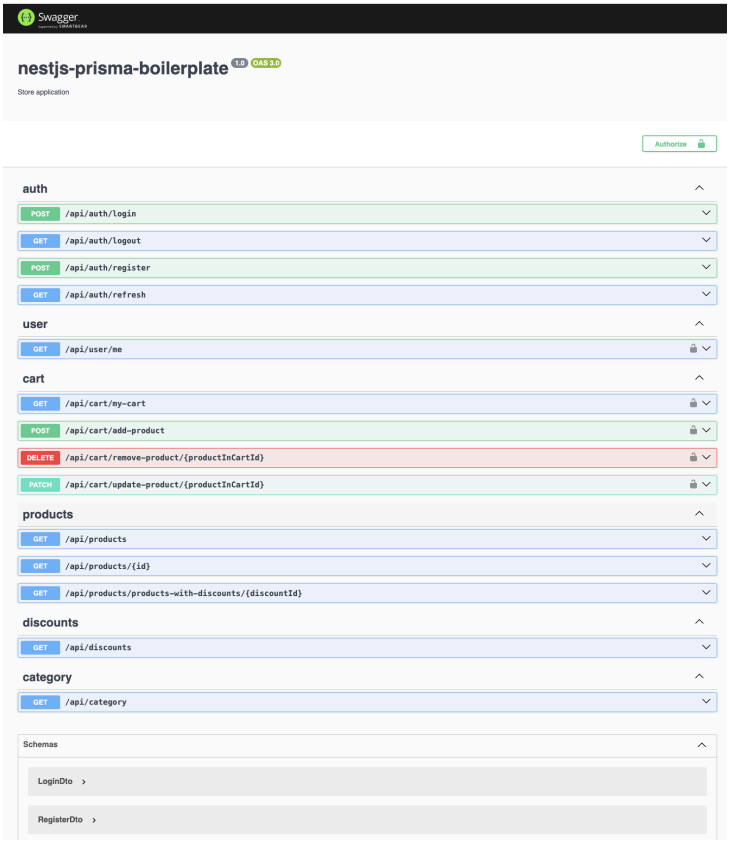

Рисунок 5 - swagger

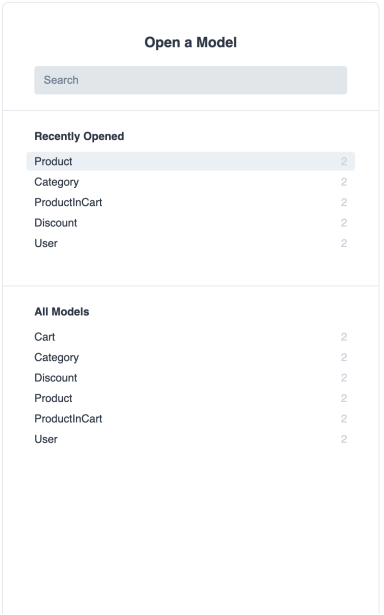

Рисунок 6 - prisma studio

Ссылка на результат -
https://github.com/pavel-nikitin-2022/nest-clothing-store

Вывод

В ходе лабораторной работы я освоил такой инструмент совместной разработки как liveshare, закрепил базовые знания по nest и prisma, научился грамотно декомпозировать задачи и разделять сущности.