Implementing Products Search with Next.js

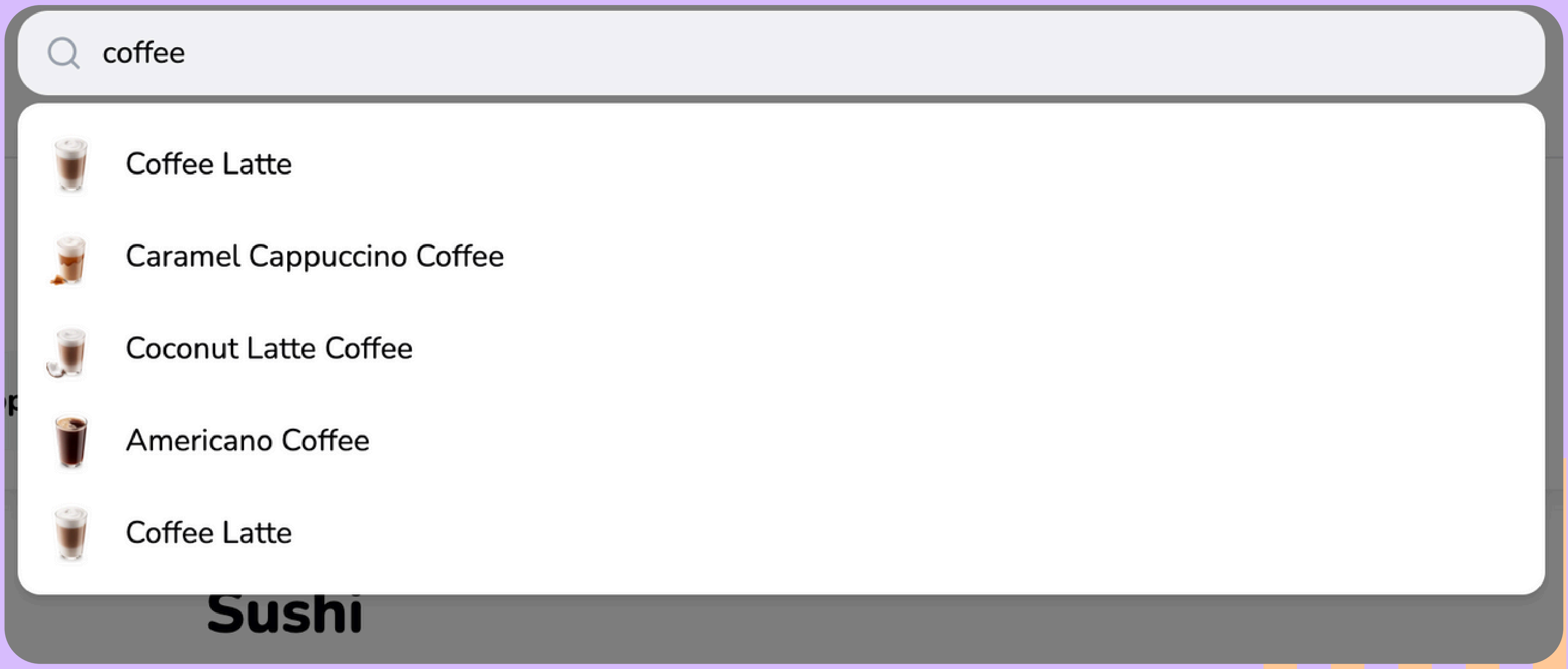
Search-Enabled Marketplace

Vladyslav Demirov
@vladyslav-demirov

The Use Case

💡 Imagine a sushi delivery app where users can search for products like “Dragon Roll” or “Salmon Sushi.”

📌 Key Features:
- Fetch product data dynamically from a database.
- Implement debounced search queries for real-time results.
- Display results in a dropdown for easy navigation.


coffee

Coffee Latte
Caramel Cappuccino Coffee
Coconut Latte Coffee
Americano Coffee
Coffee Latte

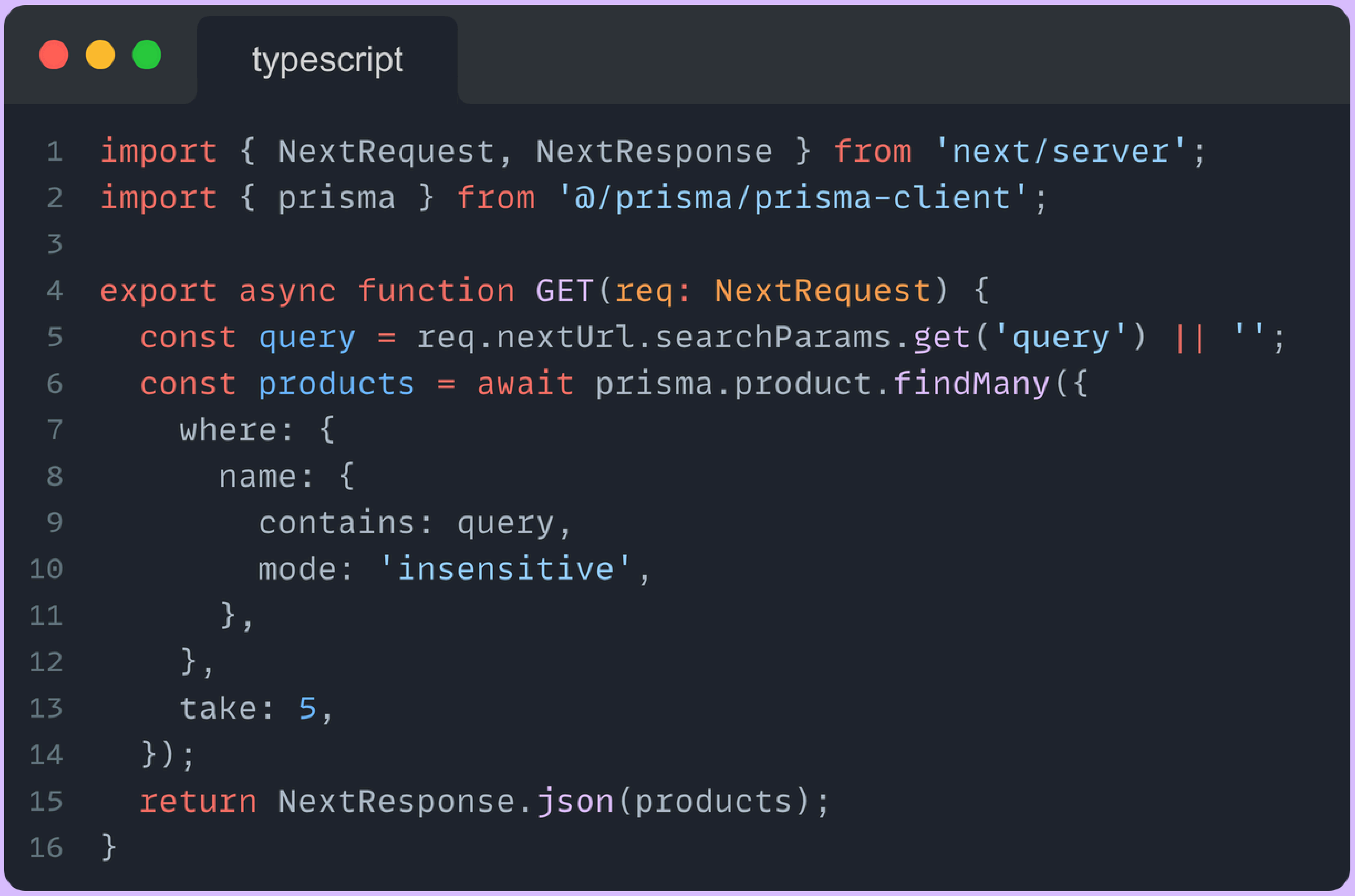
Sushi

Vladyslav Demirov

Backend API with Prisma

Create a **search endpoint** using Prisma ORM for optimised database queries.

api/products/search/route.ts



```typescript
import { NextRequest, NextResponse } from 'next/server';
import { prisma } from '@/prisma/prisma-client';

export async function GET(req: NextRequest) {
  const query = req.nextUrl.searchParams.get('query') || '';
  const products = await prisma.product.findMany({
    where: {
      name: {
        contains: query,
        mode: 'insensitive',
      },
    },
    take: 5,
  });
  return NextResponse.json(products);
}
```

Vladyslav Demirov

Setting Up an API Client

Create a **search endpoint** using Prisma ORM for optimised database queries.

services/products.ts

typescript

```typescript
import { axiosInstance } from '@/services/instance'
import { Product } from '@prisma/client'
import { ApiRoutes } from '@/services/constants'

export const search = async (query: string): Promise<Product[]> => {
  const { data } = await axiosInstance.get<Product[]>(
    ApiRoutes.SEARCH_PRODUCTS,
    { params: { query } },
  )
  return data
}
```

services/api-client.ts

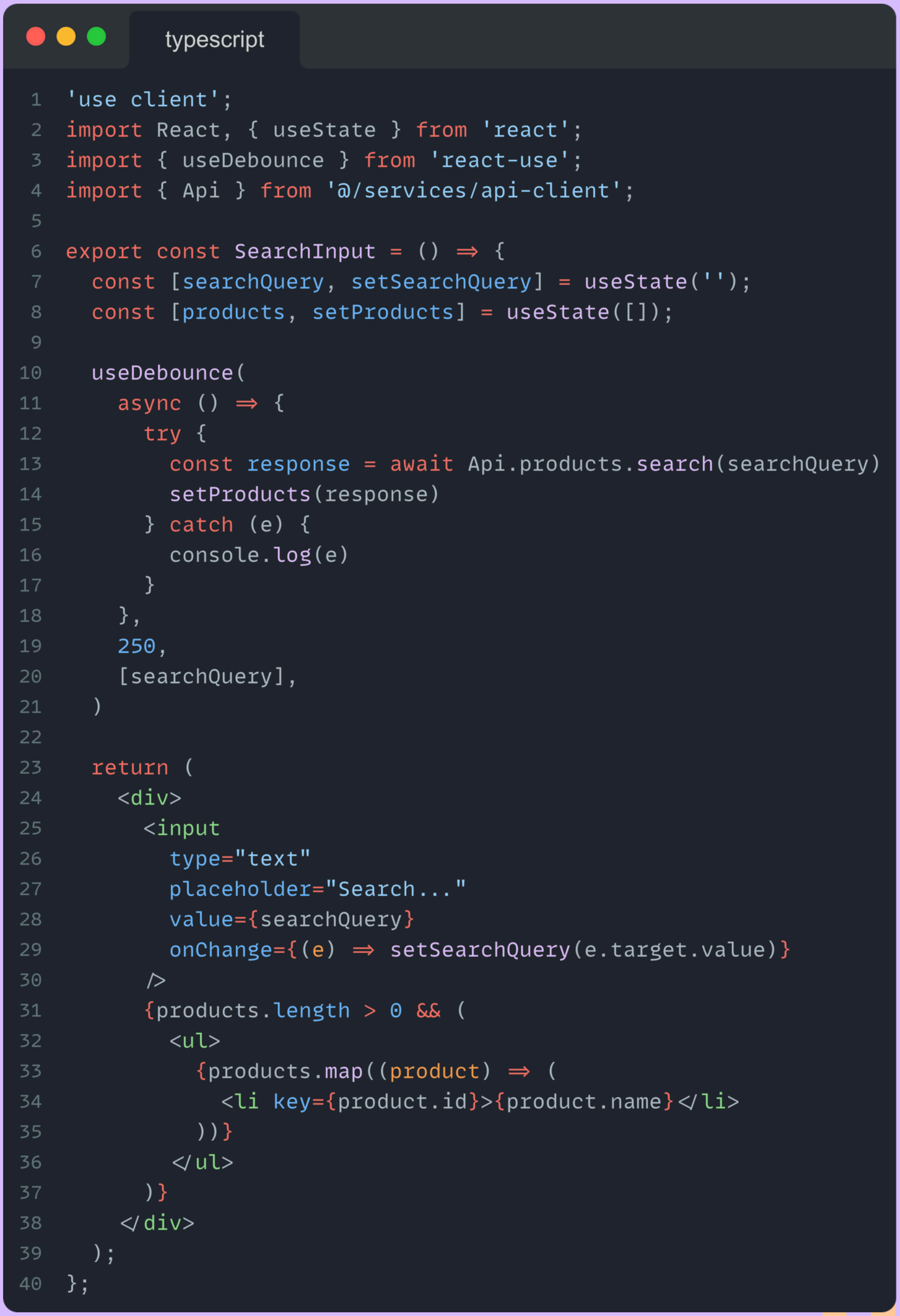
typescript

```typescript
import * as products from './products'

export const Api = {
  products,
}
```

Vladyslav Demirov

Debounced Search Component

Search bar with debouncing to minimise API calls.

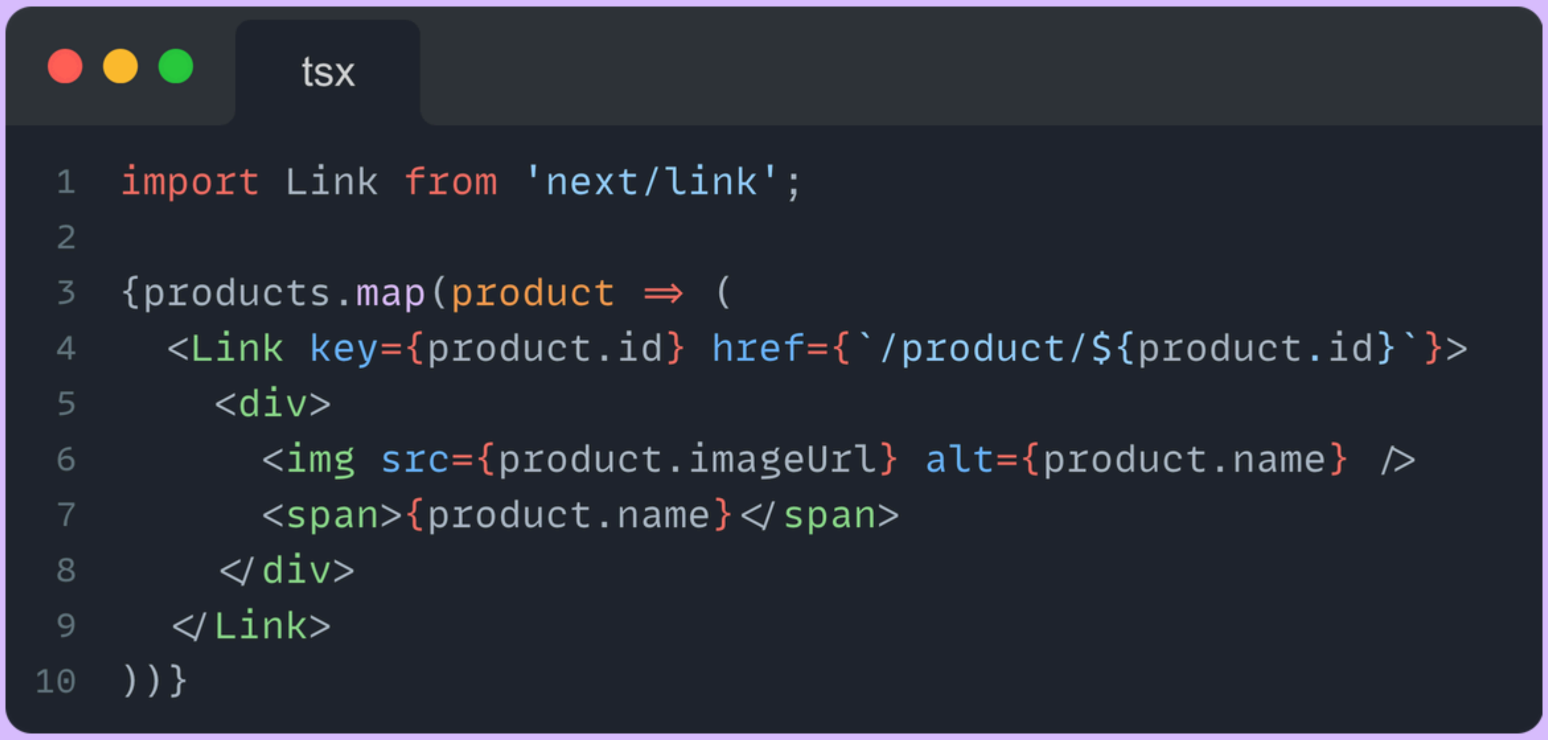


```typescript
'use client';
import React, { useState } from 'react';
import { useDebounce } from 'react-use';
import { Api } from '@/services/api-client';

export const SearchInput = () => {
  const [searchQuery, setSearchQuery] = useState('');
  const [products, setProducts] = useState([]);

  useDebounce(
    async () => {
      try {
        const response = await Api.products.search(searchQuery)
        setProducts(response)
      } catch (e) {
        console.log(e)
      }
    },
    250,
    [searchQuery],
  )

  return (
    <div>
      <input
        type="text"
        placeholder="Search..."
        value={searchQuery}
        onChange={(e) => setSearchQuery(e.target.value)}
      />
      {products.length > 0 && (
        <ul>
          {products.map((product) => (
            <li key={product.id}>{product.name}</li>
          ))}
        </ul>
      )}
    </div>
  );
};
```

Vladyslav Demirov

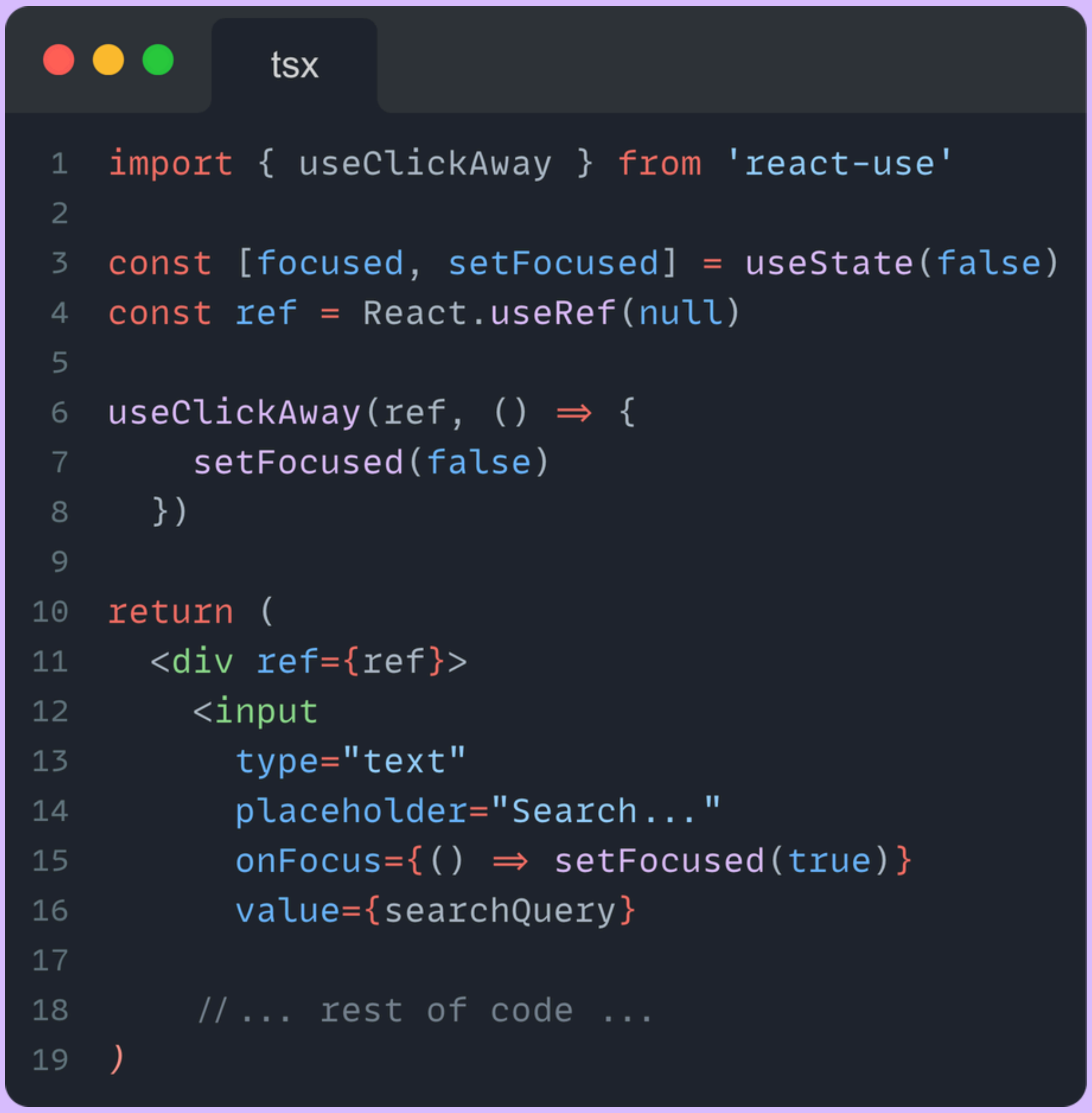
Enhancing the UI

Use **Next.js Link** for seamless navigation and display product images.



```tsx
import Link from 'next/link';

{products.map(product => (
  <Link key={product.id} href={`/product/${product.id}`}>
    <div>
      <img src={product.imageUrl} alt={product.name} />
      <span>{product.name}</span>
    </div>
  </Link>
))}
```

useClickAway() from **react-use** for focus control.



```tsx
import { useClickAway } from 'react-use'

const [focused, setFocused] = useState(false)
const ref = React.useRef(null)

useClickAway(ref, () => {
    setFocused(false)
  })

return (
  <div ref={ref}>
    <input
      type="text"
      placeholder="Search..."
      onFocus={() => setFocused(true)}
      value={searchQuery}

    // ... rest of code ...
)
```

Vladyslav Demirov

Conclusion

Benefits of This Approach

✅ Optimized Queries: Prisma ensures minimal database overhead.
✅ Debounced Search: Reduces API calls for better performance.
✅ Real-Time Results: Enhances user experience.
✅ Scalable: Supports growing product catalogs effortlessly.

Final Structure:

1️⃣ Backend: Prisma-based search API in /pages/api/products.
2️⃣ API Client: Axios abstraction for cleaner calls.
3️⃣ Frontend: React-based, debounced search component.
4️⃣ UI: Responsive dropdown with product details and navigation.

Vladyslav Demirov

HAPPY CODING

FOLLOW ME

Vladyslav Demirov
@vladyslav-demirov