

Data structure : The API provides Product Data including fields like “productName”, “category”, “price” , “description” , “image”

Library Used: Axios was used to fetch data from the API.

Fetching Data from API

```
const response = await axios.get('https://6784fdee1ec630ca33a6b00b.mockapi.io/products');  
const products = response.data;
```

Final schema 😊

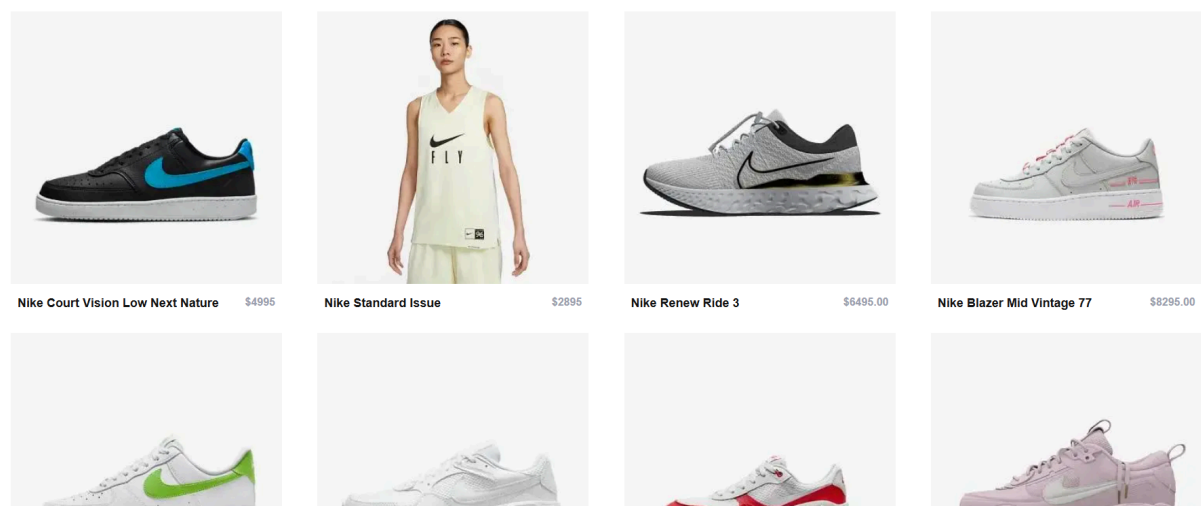
You, 3 weeks ago | 1 author (You)

```
interface ProductSchema {
  name: string;
  title: string;
  type: string;
  fields: ProductField[];
}

const productSchema: ProductSchema = {
  name: 'product',
  title: 'Product',
  type: 'document',
  fields: [
    {
      name: 'id',
      title: 'ID',
      type: 'number',
      validation: Rule => Rule.required().integer().min(1)
    },
    {
      name: 'productName',
      title: 'Product Name',
      type: 'string',
      validation: Rule => Rule.required()
    },
    {
      name: 'category',
      title: 'Category',
      type: 'string',
      validation: Rule => Rule.required()
    },
    {
      name: 'colors',
      title: 'Colors',
      type: 'array',
      of: [{ type: 'string' }],
      validation: Rule => Rule.required()
    },
    {
      name: 'price',
      title: 'Price',
      type: 'string',
      validation: Rule => Rule.required().positive()
    },
    {
      name: 'inventory',
      title: 'Inventory',
      type: 'number',
      validation: Rule => Rule.required().integer().min(0)
    },
    {
      name: 'status',
      title: 'Status',
      type: 'string',
      options: {
        list: [
          { title: 'Just In', value: 'Just In' },
          { title: 'Available', value: 'Available' },
          { title: 'Out of Stock', value: 'Out of Stock' }
        ],
        layout: 'dropdown'
      },
      validation: Rule => Rule.required()
    },
    {
      name: 'image',
      title: 'Image',
      type: 'image'
    }
  ]
}
```

See all 8 launches @ 0.0.0

Frontend Display



CMS (sanity) fields

