



# E-COMMERCE MARKETPLCE

*“NAVIGATING THE FUTURE OF  
ONLINE SHOPPING”*





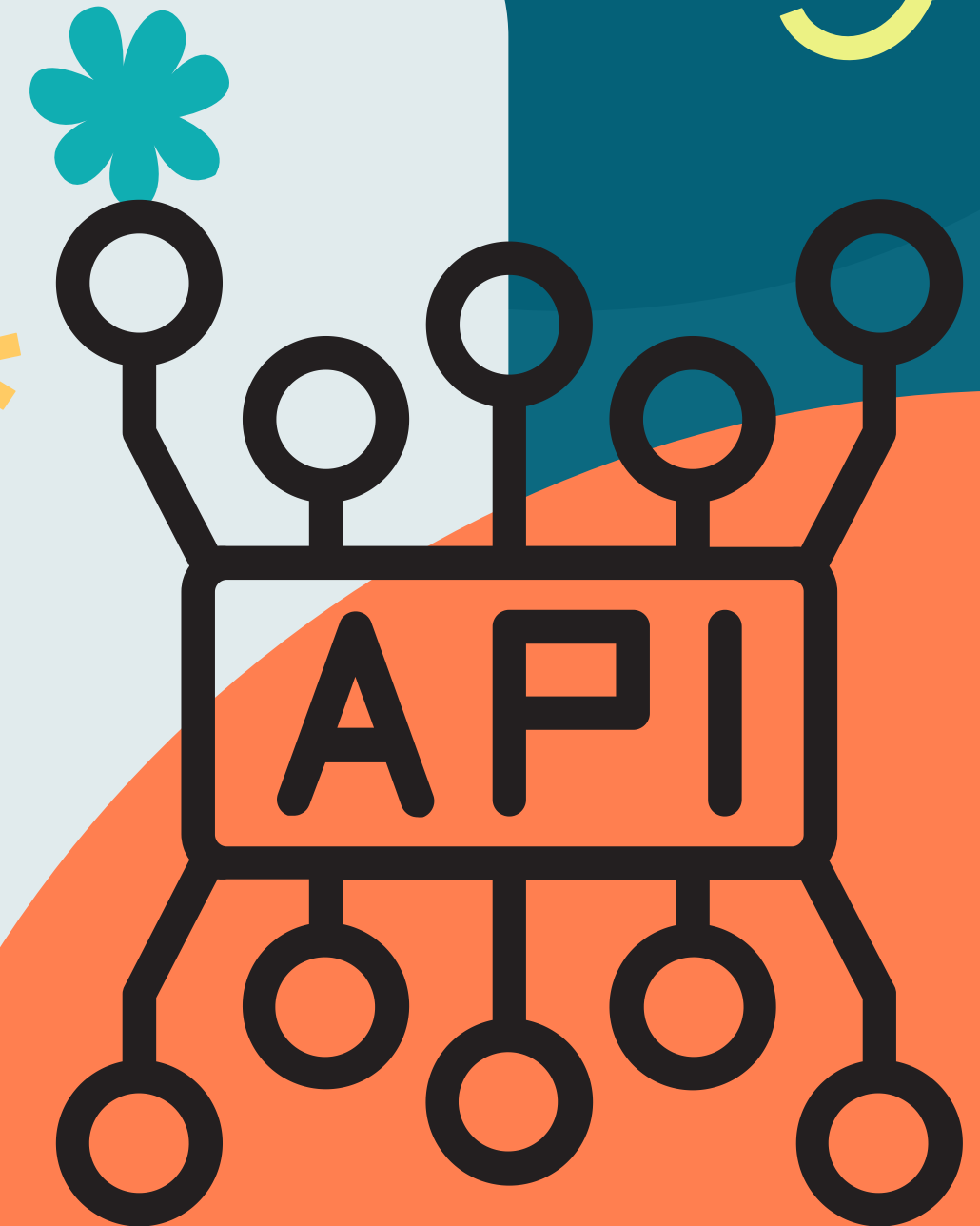
# TABLE OF CONTENT

- |          |                                    |          |                            |
|----------|------------------------------------|----------|----------------------------|
| <b>1</b> | API INTEGRATION AND DATA MIGRATION | <b>5</b> | DATA MIGRATION OPTIONS     |
| <b>2</b> | PROVIDED APIS                      | <b>6</b> | API INTEGRATION IN NEXT.JS |
| <b>3</b> | UNDERSTAND THE PROVIDED API        | <b>7</b> | SELF-VALIDATION CHECKLIST  |
| <b>4</b> | VALIDATE AND ADJUST YOUR SCHEMA    |          |                            |

# DAY 3 - API INTEGRATION REPORT SHOP.CO

## REVIEWED API DOCUMENTATION:

I carefully reviewed the API documentation and understood how the /products endpoint works. I analyzed the structure of the API response data, including the fieldnames and their data types



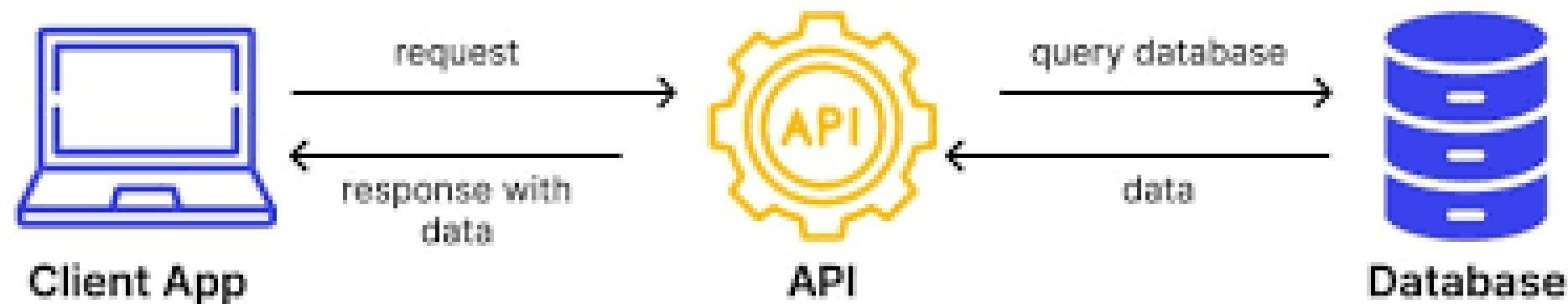


# SET UP API CALLS:

I used Postman to test the API endpoint and ensure the data was being returned correctly. In my Next.js project, I created utility functions to interact with the API. I utilized fetch to make POST requests to the API endpoints and stored the responses in variables. To verify the data structure, I logged the API responses in the console

# \* TYPES OF API CALLS ●

API calls are the medium by which they interact. An API call, or API request, is a message sent to a server asking an API to provide a service or information



## API METHODS

GET:

POST:

PUSH:

DELETE



# COMPARED API DATA WITH SANITY SCHEMA



I reviewed the API data structure and compared it with the existing schema in Sanity CMS. I identified discrepancies in field names and data types.

I UPDATED THE SANITY SCHEMA TO ALIGN WITH THE API DATA STRUCTURE. FOR INSTANC

API Field: product\_title → Sanity Field: name

API Field: price → Sanity Field: price (with the correct datatype)

I also added new fields in Sanity CMS to accommodate additional data from the API, as the API alone was insufficient to complete the product details on my website.



# **TO MIGRATE DATA FROM THE API TO SANITY CMS, I FOLLOWED THESE STEPS**

**I DECIDED TO USE THE PROVIDED API TO FETCH DATA AND WROTE A SCRIPT TO IMPORT IT INTO SANITY CMS.**

**I CREATED A SCRIPT FOLDER AND THEN CREATED A MIGRATION (.MJS) FILE TO FETCH DATA FROM THE API AND TRANSFORM IT INTO THE FORMAT REQUIRED BY SANITY CMS.**

**I USED THE SANITY CLIENT LIBRARY TO UPLOAD THE PRODUCT DATA TO THE CMS. I RAN THE MIGRATION SCRIPT TO IMPORT THE PRODUCT DATA INTO SANITY CMS.**

**I VERIFIED THE IMPORTED DATA BY CHECKING THE SANITY DASHBOARD AND ENSURING THAT ALL FIELDS WERE CORRECTLY POPULATED.**



# PROJECT OVERVIEW: API INTEGRATION AND DATA MIGRATION TO SANITY CMS

In this project, I successfully integrated the provided API into my Next.js frontend and migrated product data into Sanity CMS. I adjusted the schema to align with the API data structure and ensured the data was accurately displayed in the frontend. This exercise provided me with valuable practical experience in API integration, data migration, and schema validation, which are essential skills for building scalable marketplaces.

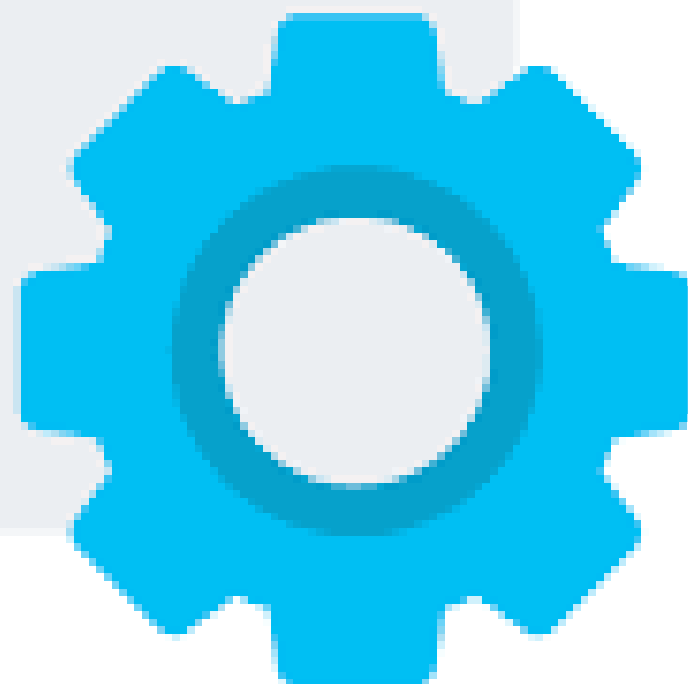
API INTEDRIATION

Data Migration to SanityCMS

Frontend  
Display(Accurate  
Data)



# API



40%

```
Selection View Go Run ... ← →
migrate.mjs U
1 NEXT_PUBLIC_SANITY_PROJECT_ID="aputurw"
2 NEXT_PUBLIC_SANITY_DATASET="production"
3 NEXT_PUBLIC_SANITY_AUTH_TOKEN="skkcehIVuq3oYyz2r63Det3t2SeuizlKkymT8UcQKspkHd3LTMSCLHCEQJpWqWqyZu"
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS
}
Uploading image: https://cdn.sanity.io/images/7xb4qach/production/86f5d2d6686d3ed48142a28f9da39fa633fc56b-193x262.jpg
Image uploaded successfully: image-86f5d2d6686d3ed48142a28f9da39fa633fc56b-193x262.jpg
Product Classic Black Pullover Hoodie uploaded successfully: {
  _createdAt: '2025-01-18T08:12:31Z',
  _id: 'SpI3808PMKhRhuCfuQxL83',
  _rev: 'SpI3808PMKhRhuCfuQxL83',
  _type: 'product',
  _updatedAt: '2025-01-18T08:12:31Z',
  category: 'hoodie',
  description: 'This versatile and stylish black hoodie is a must-have in any wardrobe. Featuring a comfortable, soft
  lined fit, it is perfect for casual outings or layering for colder weather. The hoodie includes an adjustable drawstr
  us front pocket, making it both practical and fashionable. The black color adds a timeless touch, making it easy to
  shorts, or joggers for an effortlessly cool look.',
  discountPercent: 0,
  image: {
    _type: 'image',
    asset: {
      _ref: 'image-86f5d2d6686d3ed48142a28f9da39fa633fc56b-193x262.jpg'
    }
  },
  isNew: false,
  name: 'Classic Black Pullover Hoodie',
  price: 120
}
Uploading image: https://cdn.sanity.io/images/7xb4qach/production/e44986c6cf46ac416a2f4dc8a3acbbas55457e-278x296.png
```

# Client.ts



athon\.next\server

```
1 import { createClient } from 'next-sanity'
2
3 import { apiVersion, dataset, projectId } from '../env'
4
5 export const client = createClient({
6   projectId: process.env.NEXT_PUBLIC_SANITY_PROJECT_ID,
7   dataset: process.env.NEXT_PUBLIC_SANITY_DATASET,
8   apiVersion: '2025-01-17',
9   useCdn: true, // Set to false if statically generating pages, using ISR or tag-based revalidation
10  token: process.env.NEXT_PUBLIC_SANITY_AUTH_TOKEN,
11 })
12
```

# products.ts

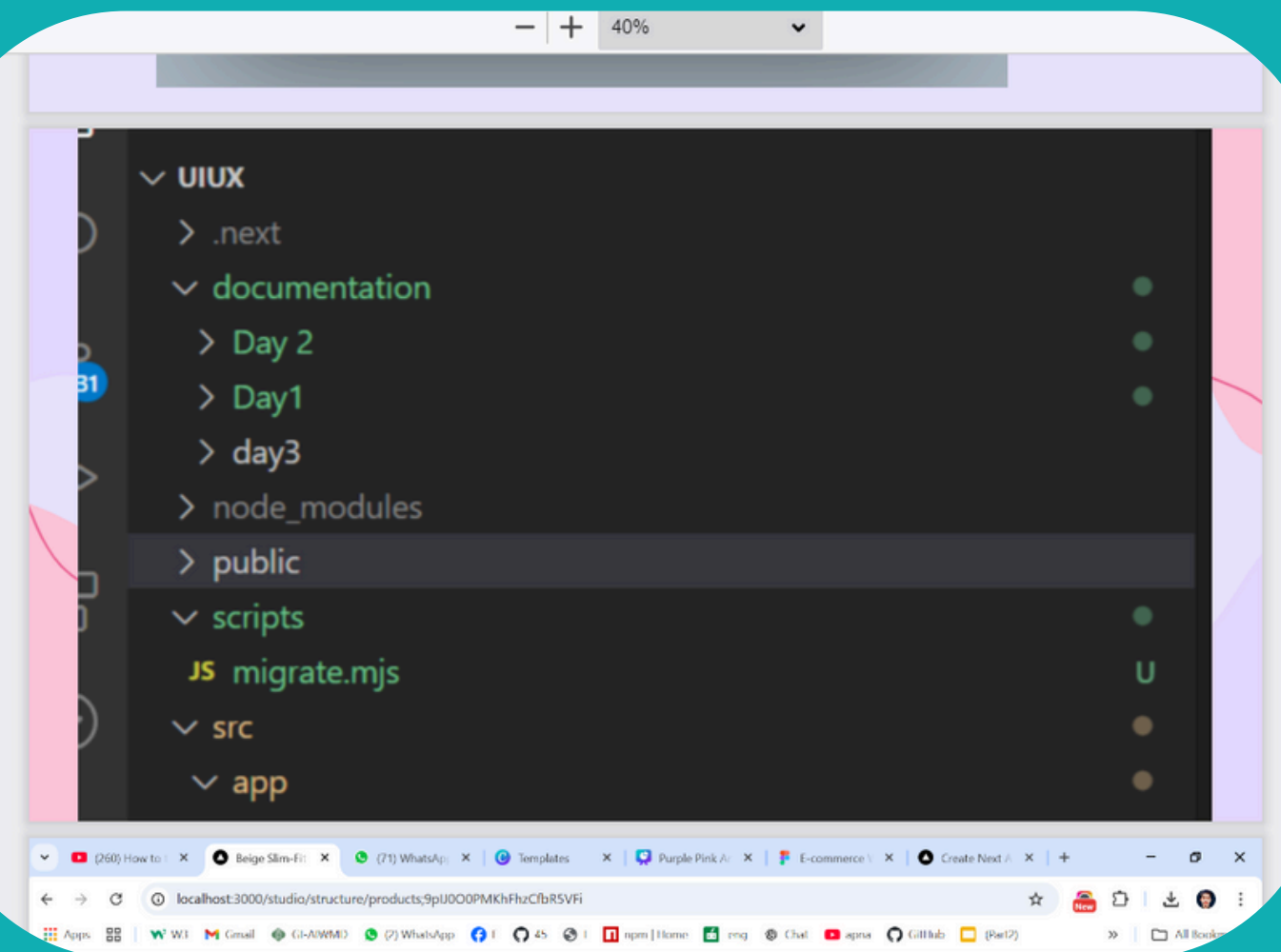


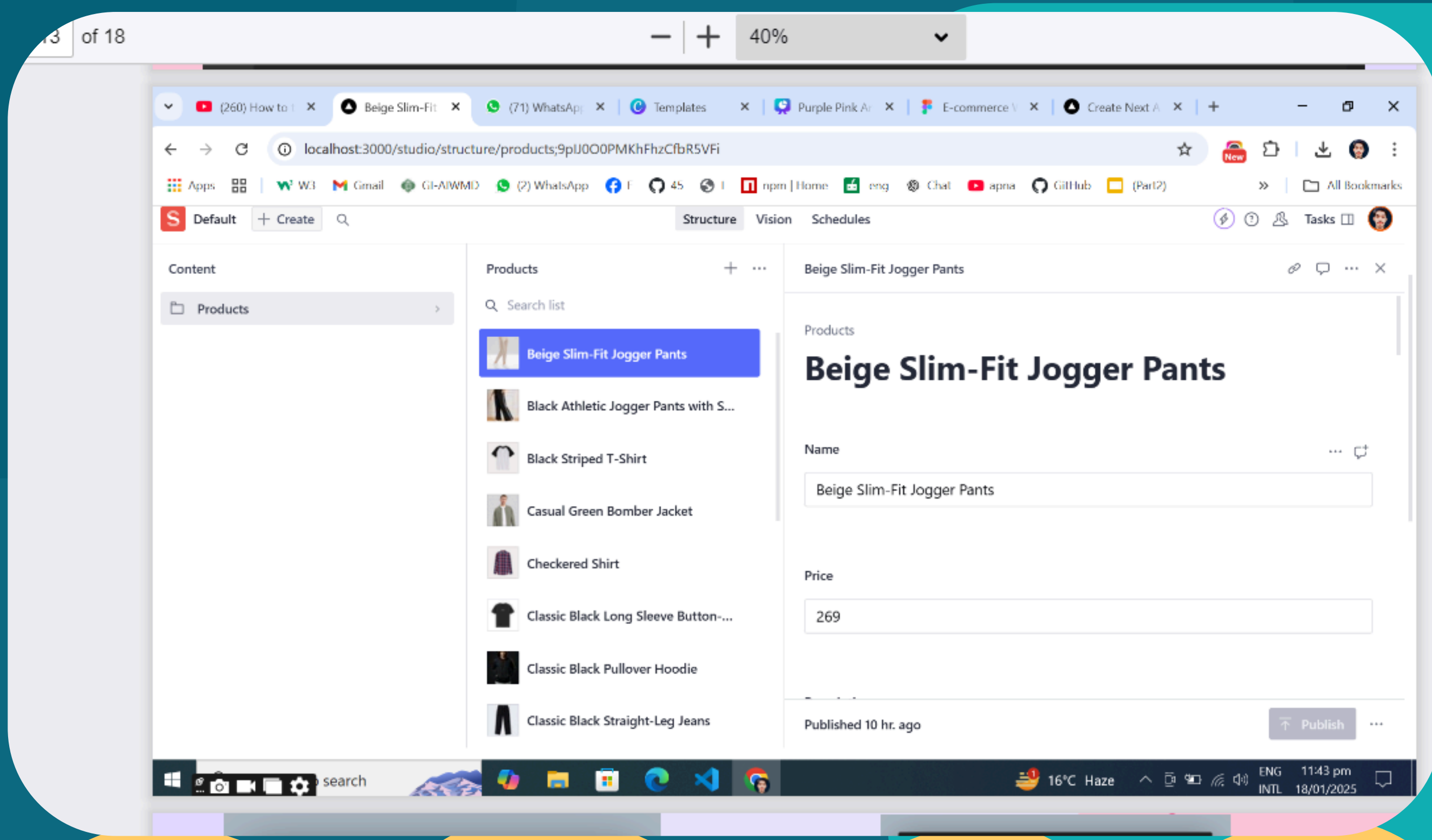
```
0 of 18 - + 40% v
1 import { defineType } from "sanity"
2
3 export default defineType({
4   name: 'products',
5   title: 'Products',
6   type: 'document',
7   fields: [
8     {
9       name: 'name',
10      title: 'Name',
11      type: 'string',
12    },
13    {
14      name: 'price',
15      title: 'Price',
16      type: 'number',
17    },
18    {
19      name: 'description',
20      title: 'Description',
21      type: 'text',
22    },
23    {
24      name: 'image',
25      title: 'Image',
26      type: 'image',
27    },
28    {
29      name: "category",
30      title: "Category",
31      type: 'string',
32      options: {
33        list: [
34          {title: 'T-Shirt', value: 'tshirt'},
35          {title: 'Short', value: 'short'},
36          {title: 'Jeans', value: 'jeans'},
37          {title: 'Hoodie', value: 'hoodie'},
38          {title: 'Shirt', value: 'shirt'},
39        ]
40      }
41    },
42    {
43      name: "discountPercent",
44      title: "Discount Percent",
45      type: 'number',
46    },
47    {
48      name: "new",
49      type: 'boolean',
50      title: "New",
51    },
52    {
53      name: "colors",
54      title: "Colors",
55      type: 'array',
56      of: [
57        {type: 'string'}
58      ]
59    },
60    {
61      name: "sizes",
62      title: "Sizes",
63      type: 'array',
64      of: [
65        {type: 'string'}
66      ]
67    }
68  ],
69 })
```

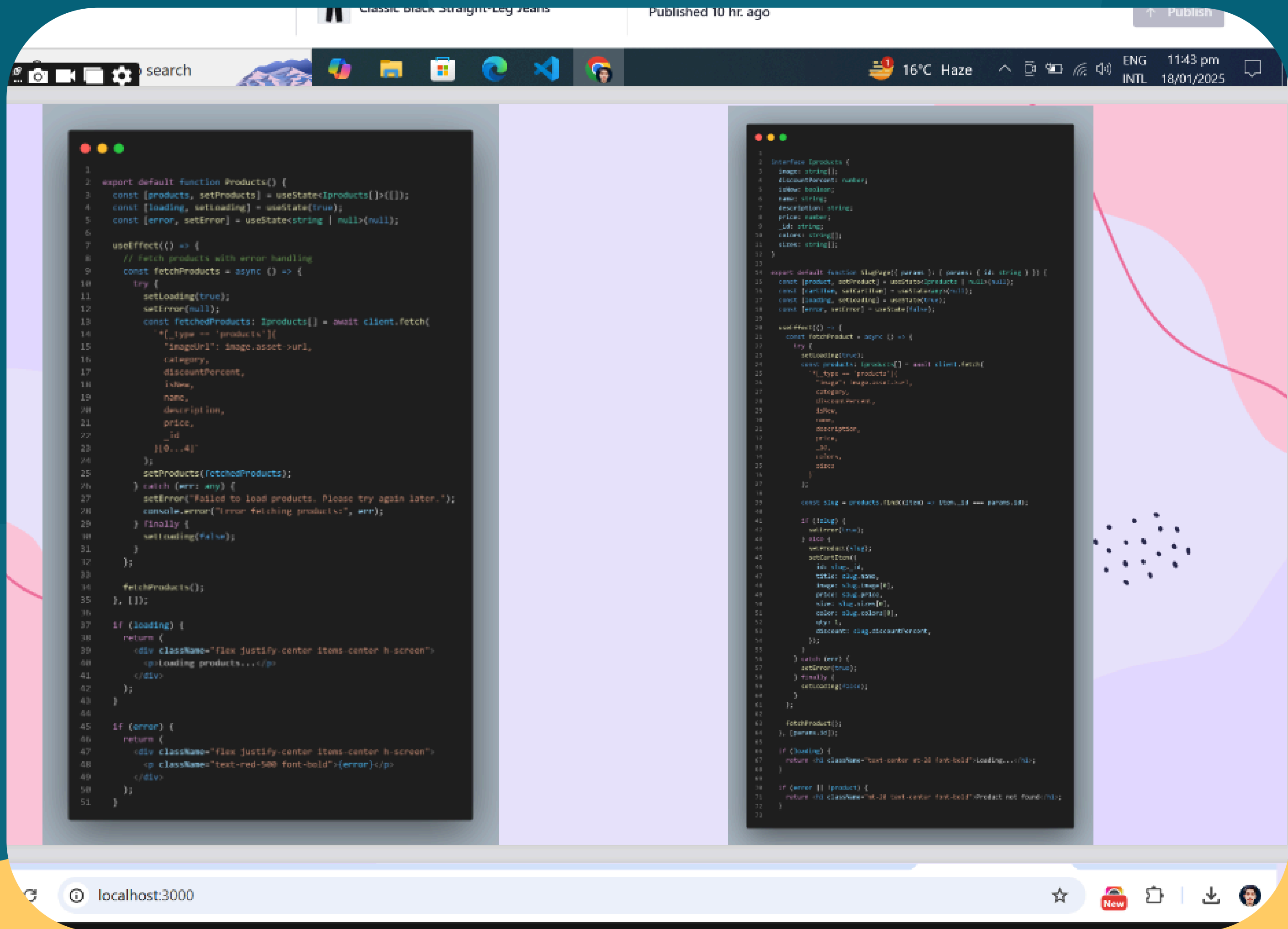
# package.json

```
1 {
2   "name": "uiux",
3   "version": "0.1.0",
4   "private": true,
5   "scripts": {
6     "dev": "next dev",
7     "build": "next build",
8     "start": "next start",
9     "lint": "next lint",
10    "migrate": "node scripts/migrate.mjs"
11  },
```

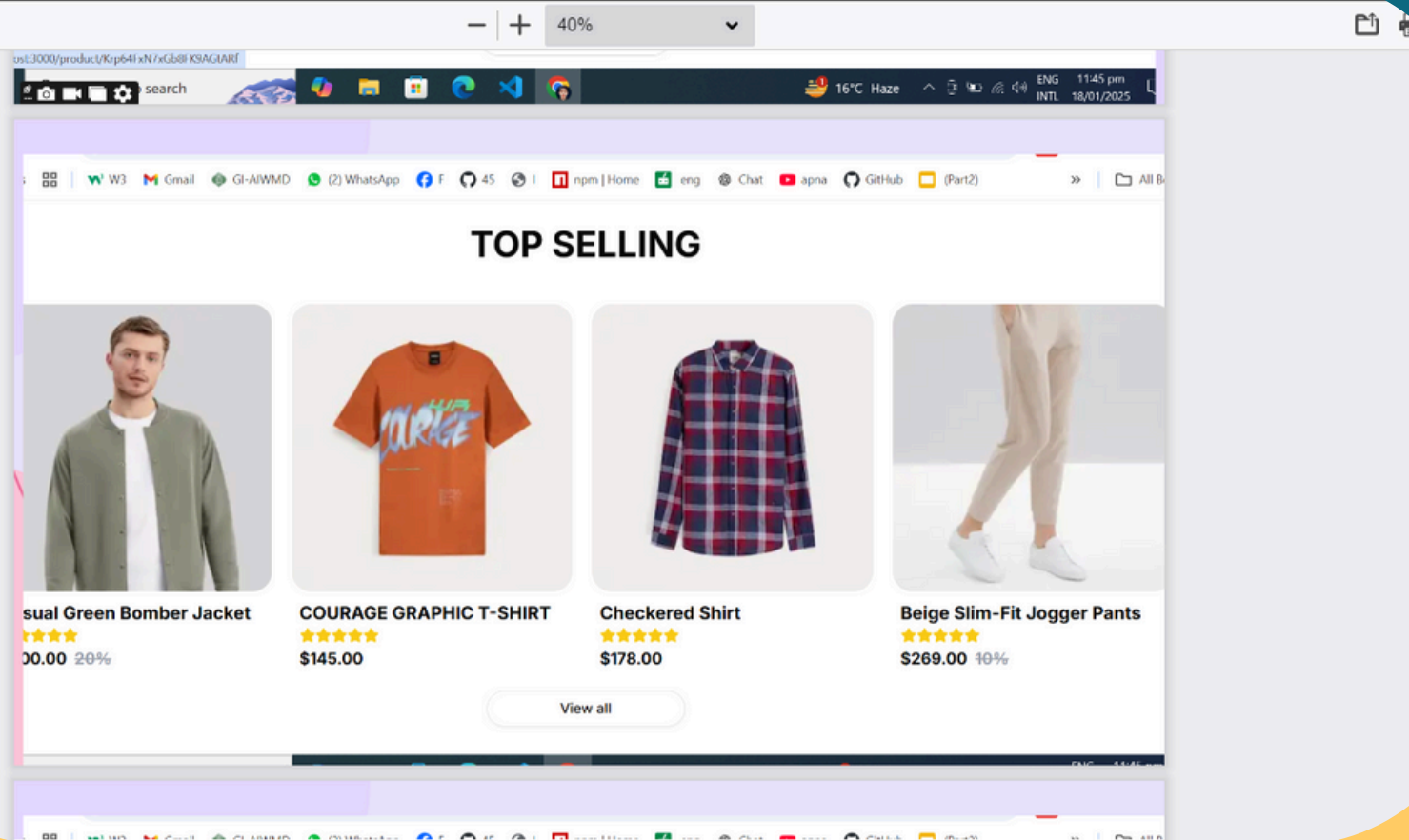
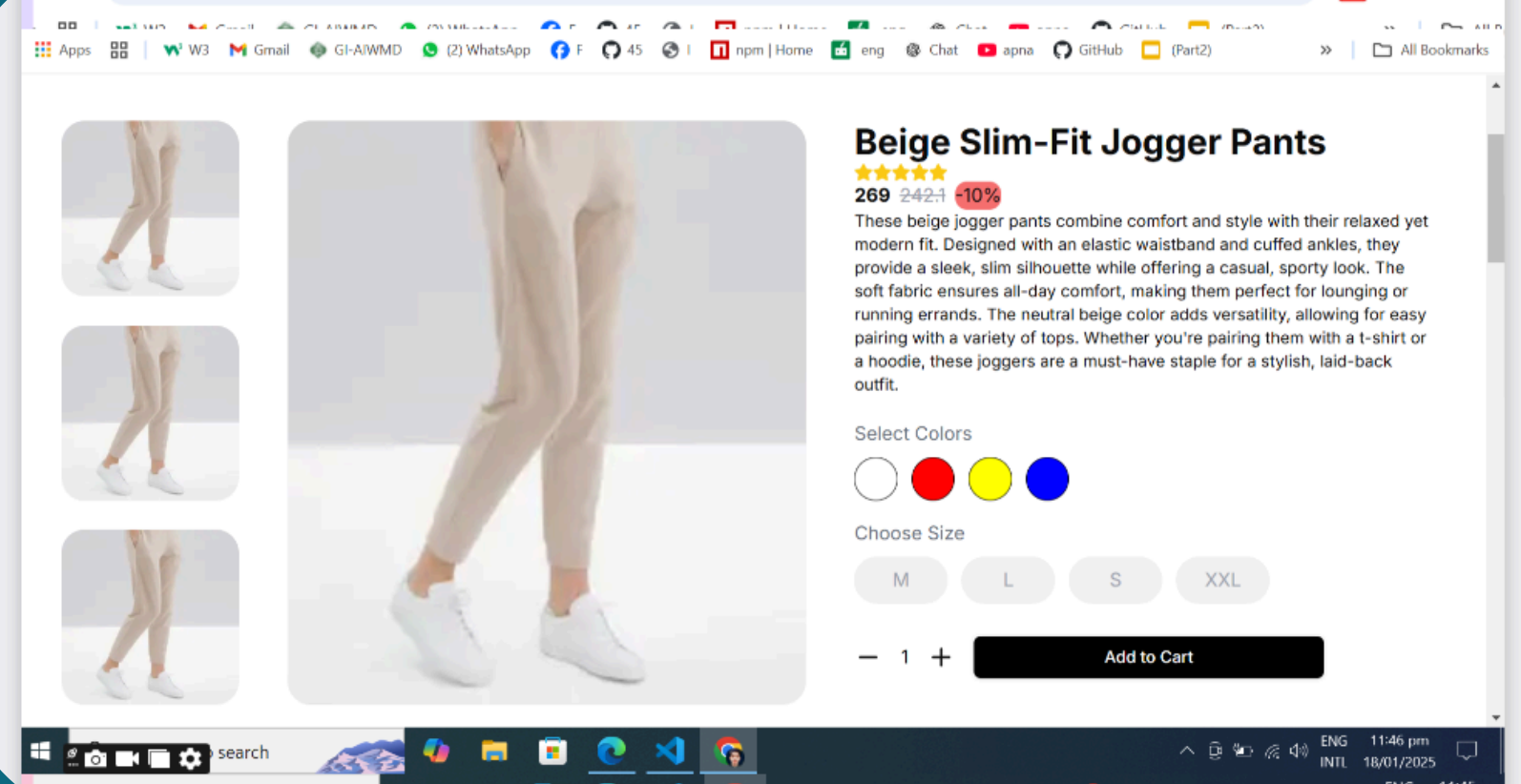
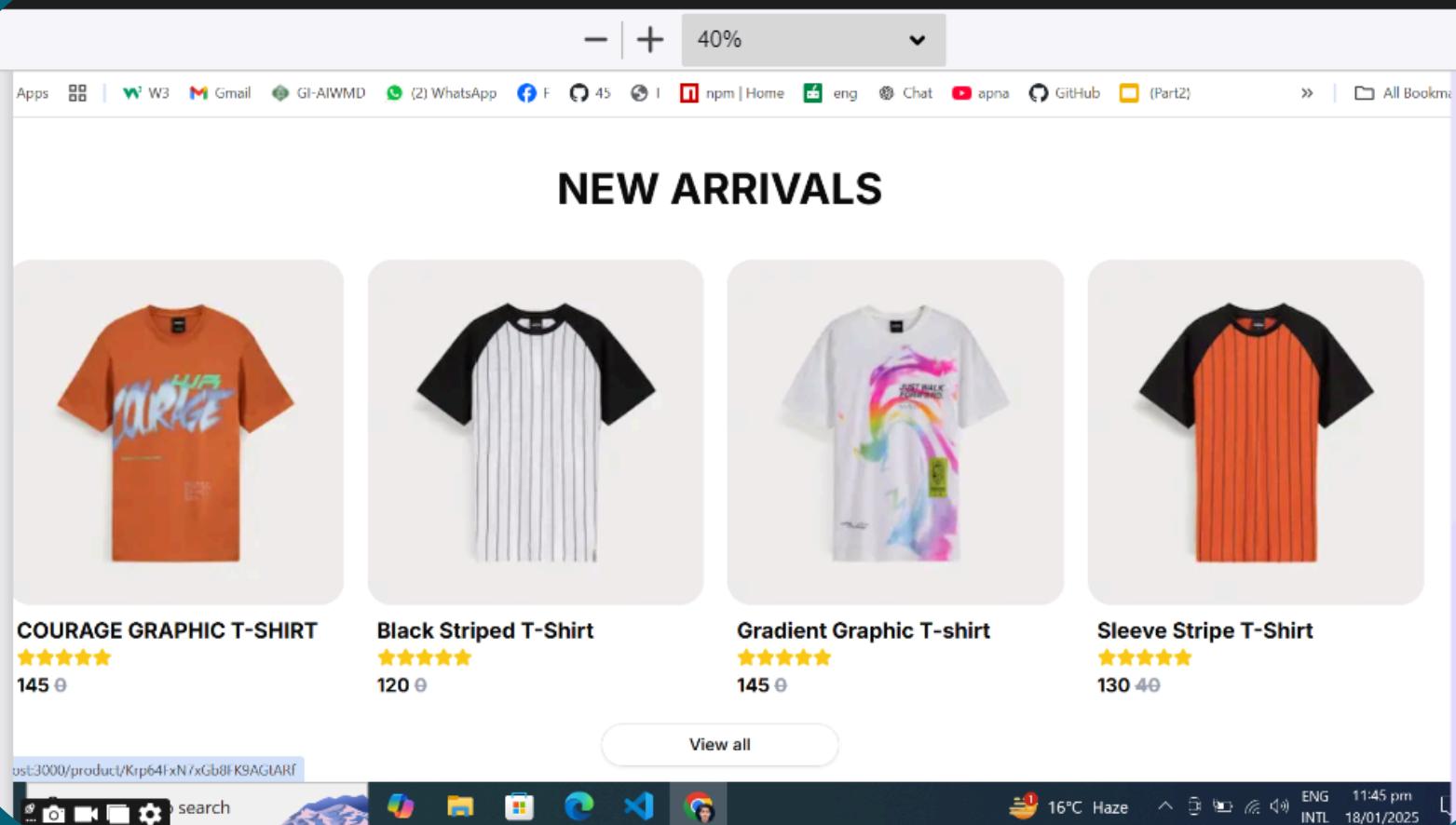
# Data migrate file











**THANK  
YOU**

