

DAY 3 : INTEGRATION & DATA MAIGRATION

env local file

```
$ envlocal X
1 # env local file
2 NEXT_PUBLIC_SANITY_PROJECT_ID="jmb9bw01"
3 NEXT_PUBLIC_SANITY_DATASET="production"
4 SANITY_API_TOKEN="skyd91z8TeN6o0qg17NwI8xCg7gbemL7i18zs1X6Iysapdv90b4mw8L1Z9yMapN9Y63P1D5891t4wjJ0iLy"
```

in package,jason import-data

```
() package,jason M X
() packagejason > ...
1 {
2   "name": "romplate_0",
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    "import-data": "node scripts/importSanityData.mjs"
11  },
12  "dependencies": {
13    "@sanity/client": "^6.27.2",
14    "@sanity/image-url": "^1.1.0",
15    "@sanity/vision": "^3.72.1",
```

sanity importData

```
JS importSanityData.mjs U X
scripts > JS importSanityData.mjs > ...
1
2 import { createClient } from '@sanity/client';
3 import fetch from 'node-fetch';
4
5 // Initialize Sanity client
6 const client = createClient({
7   projectId: "jmb9bw01",
8   dataset: "production",
9   useCdn: false, // Set to true if you want faster reads
10  apiVersion: "2025-01-13",
11  token: "skyd91z8TeN6o0qg17NwI8xCg7gbemL7i18zs1X6Iysapdv90b4mw8L1Z9yMapN9Y63P1D5891t4wjJ0iLy85Kmh3zR",
12});
13
14 // Function to upload an image to Sanity
15 async function uploadImageToSanity(imageUrl) {
16   try {
17     console.log('Uploading image: ${imageUrl}');
18
19     const response = await fetch(imageUrl);
20     if (!response.ok) {
21       throw new Error('Failed to fetch image: ${imageUrl}');
22     }
23
24     const buffer = await response.arrayBuffer();
25     const bufferImage = Buffer.from(buffer);
26
27     const asset = await client.assets.upload('image', bufferImage, {
28       filename: imageUrl.split('/').pop(),
29     });
30
31     console.log('Image uploaded successfully: ${asset._id}');
32     return asset._id;
33   } catch (error) {
34     console.error('Failed to upload image:', imageUrl, error);
35     return null;
36   }
37 }
```

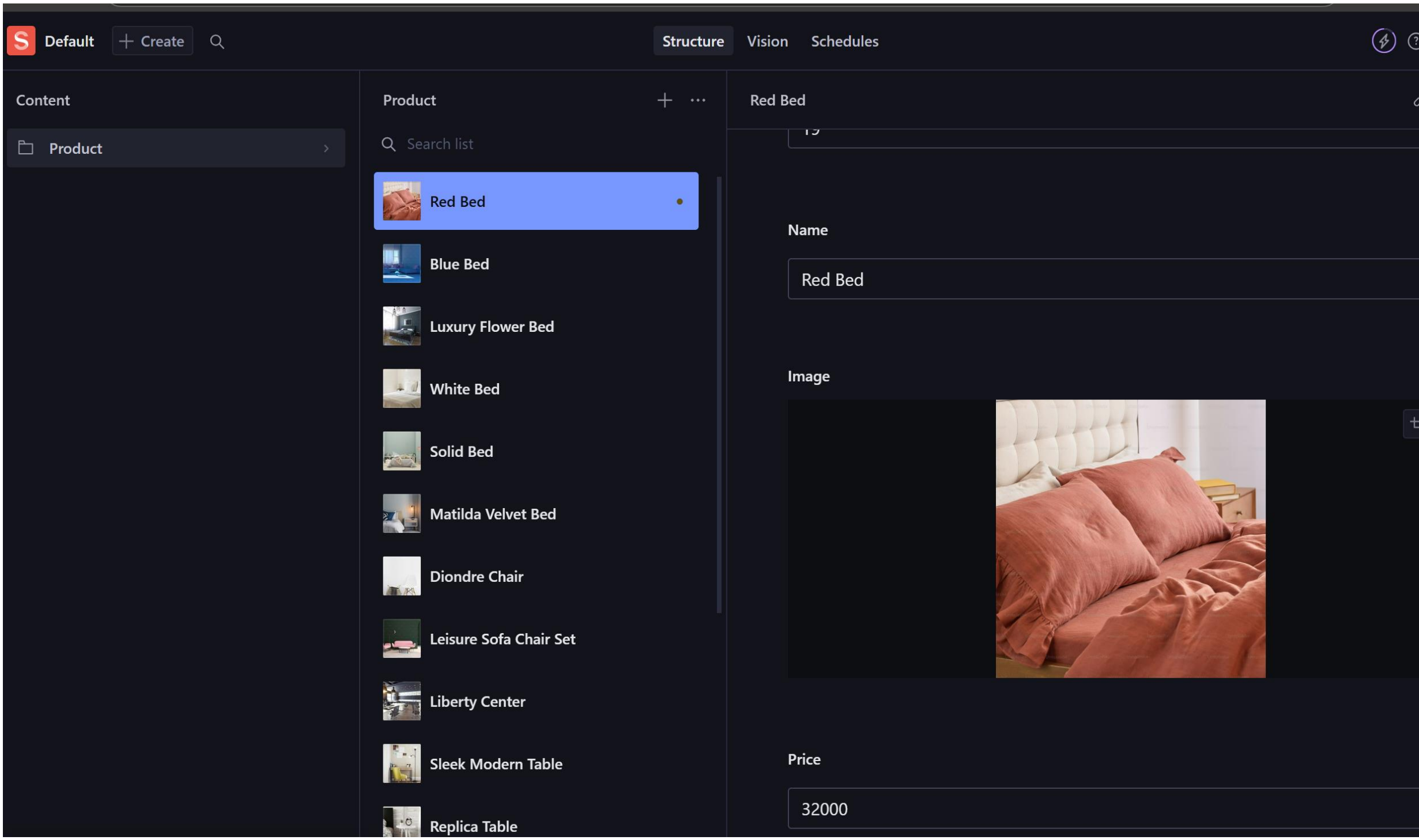
product schematype

```
src > sanity > schemaTypes > TS products > 00 default > TS fields
src > sanity > schemaTypes > TS products > 00 default > TS fields
1 export default {
2   name: 'product',
3   type: 'document',
4   title: 'product',
5   fields: [
6     {
7       name: 'id',
8       title: 'ID',
9       type: 'string',
10    },
11    {
12      name: 'name',
13      type: 'string',
14      title: 'Name',
15    },
16    {
17      name: 'image',
18      type: 'image',
19      title: 'Image',
20      options: {
21        hotspot: true, // Enable image cropping
22      },
23    },
24    {
25      name: 'price',
26      type: 'number',
27      title: 'Price',
28    },
29    {
30      name: 'description',
31      type: 'text',
32      title: 'Description',
33    },
34    {
35      name: 'discountPercentage',
36      type: 'number',
37      title: 'Discount Percentage',
38    },
39    {
40      name: 'isFeaturedProduct',
41      type: 'boolean',
42      title: 'Featured Product',
43    },
44    {
45      name: 'stockLevel',
46      type: 'number',
47      title: 'Stock level',
48    },
49  ],
50  {
51    name: 'category',
52    type: 'string',
53    title: 'Category',
54  },
55  ],
56  1,
57  };
58 }
```

import products in index.ts file

```
TS index.ts U X
src > sanity > schemaTypes > TS indexes > ...
1 import { type SchemaTypeDefinition } from 'sanity'
2 import product from './product'
3
4 export const schema: { types: SchemaTypeDefinition[] } = {
5   types: [product],
6 }
7
```

display prodict in sanity studio



ALL FILES REVIEW

